# 産業テキスト情報とグラフニューラルネットを用いた潜在的取引の予測

# Transaction Prediction using Textual Industry Information and Graph Neural Network

皆川直人 [1,2*] 和泉潔 [1] 坂地泰紀 [1] 佐野仁美 [1]
Naoto Minakawa[1,2] Kiyoshi Izumi[1] Hiroki Sakaji[1] Hitomi Sano[1]

[1] 東京大学
[1] The University of Tokyo
[2] 株式会社みずほ銀行
[2] Mizuho Bank, Ltd.

**Abstract:** 金融機関の取引データを活用すると，企業の活動をリアルタイムで把握することが可能である．これらのデータを有効活用すれば与信管理をはじめ，CRM 等にも活用使途が見出せる．一方，主要な取引先以外の企業については取引量が少ない等，データ活用上は注意を要する．すなわち，潜在的な取引を見逃している可能性がある．本問題に対し，取引主体をノード，取引有無をエッジとした企業間ネットワークを構成すれば，取引有無の予測は本ネットワーク上のリンク予測問題として定式化できる．昨今のグラフ上の深層学習の進展に伴い，より精度の高いリンク予測が可能となった．本研究では，最近のグラフ深層学習手法を使った潜在的な取引予測について紹介する．

## 1 はじめに

金融取引データを活用することにより，取引先企業がどういった企業と取引を行なっているかをリアルタイムに把握することが可能である．規模の大きな金融機関では膨大な量の取引データを有しており，経済との連関も期待できる．実際に，Garanti BBVA 銀行の取引データがトルコ経済の GDP 予測に用いられたケースでは，同銀行の個人取引データが同国 GDP の 6%，法人取引データが同国 GDP の 36%を占めることが報告されている [1].

一方，当該取引データから必ずしも全ての企業の取引を把握できるわけではなく，活動実態を把握できる企業は主要な取引先に限られる．そこで取引発生の有無を予測出来れば，本来的に発生しえる取引を把握でき，与信管理・CRM 等への応用も考えられる．例えば，取引有無がある程度正確に予測できる際，モデルによる予測結果をベンチマークとしたときに，本来見込める取引先企業からのフローが少なくなっている実態等を把握するという活用方法が考えられよう．

こうした潜在的な取引予測を考える際，オーソドックスな機械学習手法を用いる手段も考えられるが，企業間ネットワークを構成することで，単純な取引相手のみとの繋がりだけではなく，取引相手がその先でどのような企業と取引を行なっているかという情報も考慮することが可能である．従来から行われてきた複雑ネットワークを用いた分析も有効であるが [4]，昨今のグラフ上の深層学習の進展に伴い，ネットワークのトポロジーのみならず，ノードの持つ特徴等を考慮してリンク予測が行えるようになった．本研究では，金融取引データからネットワークを構築する際に特有の問題を議論しつつ，最近のグラフ深層学習手法を使った潜在的な取引予測について紹介する．

## 2 先行研究

昨今，グラフ上の深層学習・表現学習手法は急速な発展を遂げており，物理学・生物学・知識グラフ・交通・ソーシャルネットワーク・自然言語処理・画像処理等，様々な領域への応用が研究されている．例えば，タンパク質の相互作用予測，医薬品の副作用予測，知識グラフ補完，交通予測，推薦エンジン，質疑応答，画像分類等の応用例がある． [22].

金融領域においては，主に株価予測，ローンのデフォルト予測，E コマースの推薦エンジン，詐欺取引の検知，イベント予測，取引予測等の応用例がある．[11, 22, 17].

*E-mail: naoto-minakawa@g.ecc.u-tokyo.ac.jp
本論文の内容や見解は、執筆者個人に属し、所属企業の公式見解を示すものではない.

本研究と最も近い先行研究としては，様々なグラフ表現学習手法を金融取引データへと適用した，以下のようなものがある．

藤塚ら [24] は，node2vec [5] や Jaccard 係数 [12] 等の複雑ネットワーク・グラフ理論で用いられてきた方法を用いて，取引データから企業間の取引発生を行なっている．

異なるアプローチとして，クレジットカード取引の系列を考慮し，特定の時間ウィンドウの中で顧客が２つのマーチャントに対して取引を行なった際にこれらのマーチャントに対してリンクを張るという考え方で，顧客とマーチャントよりなる二部グラフを構成した例もある．[7]　本研究では，metapath2vec [3] を参考にした手法を用いて，ネットワーク上のメタパスを制限し，異質性（Heterogeneity）を考慮したマーチャントの潜在表現を得ている．

これら２つの研究で用いられている手法では，いずれもネットワークのトポロジーのみを考慮している．

Shumovskaia et al. [15] はグラフニューラルネットに基づいたアプローチを活用しており，ネットワークのトポロジー以外にも，ノードやエッジについての時系列情報も加味して取引予測を行なっている．同研究では，graph convolutional network (GCN) [9], graph attention network (GAT) [16], and SEAL [21] をベースにした手法を用いているが，銀行取引データの時間的に変化する特性を捉えることにフォーカスしている．

取引予測とは異なるもの，森ら [23] は graph convolutional network (GCN) [9], graph attention network (GAT) [16] をベースにした手法で予測対象企業の格付が特定のカテゴリー以下である確率を予測している．

今回の研究では先述の先行研究とは異なるノード特徴の活用，より金融取引ネットワークに適した取引額の考慮，異なるネットワーク構造の構成を試みる点が異なっている．

# 3　提案手法

本節では，我々が提案する金融取引ネットワークに対する表現学習手法の概要について説明する．端的には，各口座についての産業テキスト情報の潜在表現をノード特徴とし，近傍ノード情報だけでなくエッジの情報を活用した注意機構を持つ graph attention network (GAT) [16] と，ノード潜在表現について集約関数の単射性を高めるため，graph isomorphism network (GIN) [20] を活用，各ノードの潜在表現を得た後，Graph Autoencoder (GAE) [8] により取引発生を予測すべくデコードを行うものである．

## 3.1　ノード特徴

まず，各口座についての産業テキスト情報を抽出する．表１のように，各々の口座について，その企業の属する産業やセクターについての説明が記載されており，例えば Company A が加工肉を取り扱う食品産業に属することが分かる．抽出された産業テキストデータについて、空白や改行等のイレギュラーな文字を除去，語形変化・活用変化している単語の原型を復元 (stemming, lemmatization)，ストップワード (“a,”“the,”“of,” ...) の除去を行い，Doc2Vec[10] を用いて，潜在表現を得る．

## 3.2　Doc2Vec

Doc2Vec [10] は文章の潜在表現を得るための手法であり，Word2Vec [13] と類似している．Word2Vec [13] では２つのモデルが存在し，それぞれ continuous bag of words (CBOW) と skip-gram である．CBOW では特定の単語の周辺単語を考慮し，特定の単語が何かを予測するのに対し，skip-gram では特定の単語からその周辺単語を予測する．これらのタスクを学習させた後に得られる特徴を単語のベクトルとして用いる仕組みである．学習には stochastic gradient descent algorithms (SGD) がしばしば用いられる．

Doc2Vec [10] についても２つのモデルが存在し，それぞれ distributed memory model of paragraph vectors (PV-DM) と，paragraph vector with distributed bag of words (PV-DBOW) である．PV-DM が CBOW に対応し，PV-DBOW が skip-gram に対応している．

PV-DM は特定の単語の周辺単語に加えてパラグラフベクトルを考慮し，特定の単語を予測する．PV-DBOW は，対象となるパラグラフからランダムに抽出する単語を予測させる．入力においては周辺単語を無視するため，計算コストは相対的に低いが精度面で PV-DM に劣る．これらのタスク学習後に得られるベクトルをパラグラフの潜在表現として用いる．

表 1: 産業テキスト情報のイメージ

| Account | Description |
|---|---|
| Company A | food, beverage, and tobacco manufacturer; Deals with frozen meat products, processed meat products, processed milk products, and etc. |

### 3.3 Graph Attention Network

グラフを$G(V,E)$，ノードを$u \in V$，エッジを$(u,v) \in E$，ノード$u$の近傍ノードを$\mathcal{N}(u)$，隣接行列を$\mathbf{A}$，ノード$u$の$k$層目の潜在表現を$\mathbf{h}_u^{(k)}$で表す．$\mathbf{h}_u^{(0)}$は，ノード$u$についてのノード特徴である．つまり，ノード特徴行列$\mathbf{H} \in R^{|V| \times d}$のノード$u$に対応する列ベクトルとなる．$\mathbf{W}^{(k)} \in R^{|Hidden| \times |Input|}$は，$k$層目の共有パラメータであり，$|Hidden|$は隠れ層（出力）の次元を，$|Input|$は入力の次元を表す．$\alpha_{u,v}$はノード$u$のノード$v$に対するアテンションスコアである．$\sigma$は活性化関数であり，**ReLU**を採用している．

これらのノーテーションを用いると，GAT [16] は以下で表される．$k$層目における，ノード$u$の近傍ノード$\mathcal{N}(u)$についてのメッセージ集約関数$\mathbf{m}_{\mathcal{N}(u)}^{(k)}$は，

$$\mathbf{m}_{\mathcal{N}(u)}^{(k)} = \sum_{v \in \mathcal{N}(u)} \alpha_{u,v} \mathbf{h}_v^{(k)}$$

また，アテンションスコアは以下で計算される．

$$\alpha_{u,v} = \frac{exp\left(\mathbf{LReLU}(\mathbf{a}^\top [\mathbf{W}^{(k)}\mathbf{h}_u^{(k)} || \mathbf{W}^{(k)}\mathbf{h}_v^{(k)}])\right)}{\sum_{v' \in \mathcal{N}(u)} exp\left(\mathbf{LReLU}(\mathbf{a}^\top [\mathbf{W}^{(k)}\mathbf{h}_u^{(k)} || \mathbf{W}^{(k)}\mathbf{h}_{v'}^{(k)}])\right)}$$

$||$はベクトルの結合を表し，$\mathbf{a}^\top \in R^{2|Hidden|}$は重みベクトルである．ノード$u$についての潜在表現は以下のように更新される．

$$\mathbf{h}_u^{(k+1)} = \sigma\left(\mathbf{W}^{(k)}\mathbf{m}_{\mathcal{N}(u)}^{(k)}\right)$$

### 3.4 取引額を考慮した注意機構

GATの注意機構について，エッジ特徴も考慮する場合の最もスタンダードな方法は，エッジ特徴$\mathbf{e}$をノード特徴$\mathbf{h}_u^{(k)}$と$\mathbf{h}_v^{(k)}$に追加的に結合してアテンションスコアを計算する方法である [2]．このタイプの注意機構はPyTorch [1]ベースのグラフニューラルネットワークライブラリであるPyTorch Geometric [2]においても採用されている．ここで，$\mathbf{W}_e^{(k)} \in R^{|E| \times d_e}$は$k$層目におけるエッジ特徴に対する共有パラメータ，$d_e$はエッジ特徴の次元数，$\mathbf{a'}^\top \in R^{2|Hidden|+d_e}$は重みベクトル，**LReLU**は**LeakyReLU**を表す．今回の場合$\mathbf{e}_{u,v}$はノード$u$からノード$v$の取引額である．

$$\alpha_{u,v} = \frac{exp\left(\mathbf{LReLU}(\mathbf{a'}^\top [\mathbf{W}^{(k)}\mathbf{h}_u^{(k)} || \mathbf{W}^{(k)}\mathbf{h}_v^{(k)} || \mathbf{W}_e^{(k)}\mathbf{e}_{u,v}])\right)}{\sum_{v' \in \mathcal{N}(u)} exp\left(\mathbf{LReLU}(\mathbf{a'}^\top [\mathbf{W}^{(k)}\mathbf{h}_u^{(k)} || \mathbf{W}^{(k)}\mathbf{h}_{v'}^{(k)} || \mathbf{W}_e^{(k)}\mathbf{e}_{u,v'}])\right)}$$

ノード$u$に対する，メッセージ集約関数$\mathbf{m}_{\mathcal{N}(u)}^{(k)}$と更新関数は元のGATと同様である．つまり，

$$\mathbf{m}_{\mathcal{N}(u)}^{(k)} = \sum_{v \in \mathcal{N}(u)} \alpha_{u,v} \mathbf{h}_v^{(k)}$$
$$\mathbf{h}_u^{(k+1)} = \sigma\left(\mathbf{W}^{(k)}\mathbf{m}_{\mathcal{N}(u)}^{(k)}\right)$$

しかしながら，上記の注意機構では取引額の効果は限定的である．Doc2Vecを用いて得られるノード特徴は$d$次元であり，共有パラメータ$\mathbf{W}^{(k)}$における次元数$|Hidden|$についても，数十から数百となるのに対し，エッジ特徴としては取引額だけになるためスカラーである．実際に，今回の実験設定においても$d = 300$，$|Hidden| = 128$を採用している．

言い換えると、$\mathbf{W}_e^{(k)}\mathbf{e}_{u,v}$は，結合された近傍ノード特徴とエッジ特徴のベクトル$\mathbf{W}^{(k)}\mathbf{h}_u^{(k)} || \mathbf{W}^{(k)}\mathbf{h}_v^{(k)} || \mathbf{W}_e^{(k)}\mathbf{e}_{u,v}$の次元数$2|Hidden|+1$のうち，1次元のみしか貢献していない．

この問題は，我々の設定のみに固有の事象というよりは，金融取引からネットワークを構成する際に幅広く現れる問題だと考えられる．ノードについては，今回のように口座や企業のテキスト情報の潜在表現を用いたり，従業員数，資産・売上・収益といった財務諸表や収益計算書の様々な情報を考慮して容易に高次元の特徴を構成出来る一方，エッジについては，高次元の特徴を構成しにくい．取引額以外に，取引頻度や分散・標準偏差等の記述統計量を含めて特徴量の次元を増やすことも考えられるが，金融取引についてはその取引額が最も重要であり，ノード特徴と比較して特徴の候補は少ないと考えられる．本問題を解決するため，上記のエッジ特徴を考慮した注意機構の改善として，ノード特徴とエッジ特徴をバランス良く考慮するための注意機構を以下に提案する．

$$\alpha_{u,v} = \frac{exp\left(\mathbf{LReLU}\left(\mathbf{a}^\top [\mathbf{W}^{(k)}\mathbf{h}_u^{(k)} || \mathbf{W}^{(k)}\mathbf{h}_v^{(k)}] + \frac{log(e_{u,v})}{\gamma}\right)\right)}{\sum_{v' \in \mathcal{N}(u)} exp\left(\mathbf{LReLU}\left(\mathbf{a}^\top [\mathbf{W}^{(k)}\mathbf{h}_u^{(k)} || \mathbf{W}^{(k)}\mathbf{h}_{v'}^{(k)}] + \frac{log(e_{u,v'})}{\gamma}\right)\right)}$$

ここで，$\mathbf{a}^\top [\mathbf{W}^{(k)}\mathbf{h}_u^{(k)} || \mathbf{W}^{(k)}\mathbf{h}_v^{(k)}]$は，$1 \times 2|Hidden|$サイズのベクトル$\mathbf{a}^\top$と，$2|Hidden| \times 1$サイズのベクトル$[\mathbf{W}^{(k)}\mathbf{h}_u^{(k)} || \mathbf{W}^{(k)}\mathbf{h}_v^{(k)}]$の内積であるため，1次元である．

**LReLU**()の中で，1次元のノード特徴に関する情報と1次元のエッジ特徴に関する情報を足していることになり，次元数の観点でノードとエッジが同程度貢献していると解釈できる．

エッジ特徴について対数を取る理由は，取引額が10 (USD)単位から100,000,000 (USD)まで幅がある中，ごく少数の取引のみが多額となっているため，それらの取引による歪みを軽減するためである．また，$log(e_{u,v})$自体がその取引の重要性を表していると考えられるため，追加的に重みベクトルを乗しない．

[1] https://pytorch.org/
[2] https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html#torch_geometric.nn.conv.GATConv

$\gamma$ はエッジ特徴の影響度を調整するための定数、ないしは学習可能なパラメータである．$\gamma$ を調整することで，エッジ特徴 $log(e_{u,v})$ の値の範囲を，ノード特徴由来の情報 $\mathbf{a}^\top[\mathbf{W}^{(k)}\mathbf{h}_u^{(k)}||\mathbf{W}^{(k)}\mathbf{h}_v^{(k)}]$ の値の範囲と同程度にスケーリングさせることが可能である．今回の実験では，これらの値の範囲を観察し，$\gamma = 10000$ を採用した．

簡単のため，また実務的にも自然な設定であるため，ここではエッジ特徴として取引額のみ，つまり 1 次元の特徴を考慮するケースを考慮したが，多次元のエッジ特徴を考慮することも可能である．例えば，$e_{u,v} = \sum_i \beta_i \mathbf{e}_{u,v}^{(i)}$ のように，$i$ 番目のエッジ特徴 $\mathbf{e}_{u,v}^{(i)}$ について，重み $\beta_i$ を乗したものの和を取ることで，引き続きノード由来の情報とエッジ由来の情報を 1 次元に保ち，アテンションスコアを計算可能である．

### 3.5 Graph Isomorphism Network

GAT のような枠組みの典型的なグラフニューラルネットワーク，すなわち，近傍ノードからのメッセージ集約と，集約された情報の更新で構成されるようなグラフニューラルネットワークについては，グラフの同型性を見分ける表現力が高々Weisfeiler–Lehman graph isomorphism test と同等であることが示されている [20].

Weisfeiler–Lehman graph isomorphism test は，異なるグラフを見分けるためのテストで，まず最初に各ノードにラベルをアサインして初期化し，近傍ノードのラベル集合をハッシュ化することで各ノードのラベル更新を繰り返し，ラベル更新がなされなくなった段階で繰り返しを止める [6, 18].

メッセージ集約型のグラフニューラルネットワークについては，ノード集約関数が単射となるときに，その表現力が最高となり，Weisfeiler–Lehman test と同等となる [20].

GIN [20] は，上記のようなノード集約関数の単射性を満たすネットワークであり，下記のように表される．$\epsilon^{(k)}$ は $k$ 層目の学習可能なパラメータ，ないしは定数である．

$$\begin{aligned}\mathbf{m}_{\mathcal{N}(u)}^{(k)} &= \sum_{v\in\mathcal{N}(u)} \mathbf{h}_v^{(k)} \\ \mathbf{h}_u^{(k+1)} &= \mathbf{MLP}^{(k+1)}\left((1+\epsilon^{(k+1)})\mathbf{h}_u^{(k)} + \mathbf{m}_{\mathcal{N}(u)}^{(k)}\right)\end{aligned}$$

原論文において，著者らはノードのワンホットエンコーディングをノード特徴として用いているが，それ自体で単射性を保証できるためであり，ノード集約の際に多層パーセプトロン (**MLP**) を用いる必要がない．我々の場合，ノード特徴は産業テキストの潜在表現であり，口座に対して一意でない，すなわち単射性を満たさないため，以下のようにノード集約の際に **MLP** を用いている．

$$\begin{aligned}\mathbf{m}_{\mathcal{N}(u)}^{(k)} &= \sum_{v\in\mathcal{N}(u)} \mathbf{MLP}^{(k)}(\mathbf{h}_v^{(k)}) \\ \mathbf{h}_u^{(k+1)} &= \mathbf{MLP}^{(k+1)}\left((1+\epsilon^{(k+1)})\mathbf{MLP}^{(k)}(\mathbf{h}_u^{(k)}) + \mathbf{m}_{\mathcal{N}(u)}^{(k)}\right)\end{aligned}$$

### 3.6 メッセージ集約関数の単射性と産業テキスト情報の利用意義

産業テキスト情報を用いることは，一見メッセージ集約の単射性の観点で非効率的に見える．実際，今回の場合はワンホットエンコーディングを使う方が単射性の観点では優れている．しかしながら，産業テキストに口座毎の記述を追加することで容易にノードに対して一意な潜在表現を得ることが出来，また潜在表現を用いることでノード同士の類似性も導入できる．例えば Company A が加工肉を扱う会社，Company B が乳製品を扱う会社，Company C が自動車小売会社とした際，ワンホットエンコーディングではこれらの会社を単に区別するしか出来ないが，Doc2Vec 等で潜在表現を得る場合は，これらを区別しつつ，Company A と B は同じ食品業で近い業種だが，Company C は自動車小売業と別業種であることを区別できる．実務上，産業毎の慣習や，近しい取引傾向が見られることも加味すると，産業テキスト情報を活用することでより自然な取引ネットワークが構成出来ると考えられる．

### 3.7 Non-probabilistic Graph Autoencoder

GAT 層と GIN 層を経て得られるノード潜在表現は，non-probabilistic graph autoencoder (GAE) [8] を使ってデコードされ，取引発生有無の予測に用いられる．ノードの潜在表現を $\mathbf{Z}$，特に，ノード $u$ に対しての潜在表現ベクトルを $\mathbf{Z}_u$ で表すことにする．この時，元のネットワークの隣接行列 $\mathbf{A}$ は以下のように復元される．

$$\mathbf{A} \approx \mathbf{sigmoid}(\mathbf{Z}\mathbf{Z}^T)$$

特定のエッジについて，その有無，つまり取引発生有無は以下のように復元される．

$$\mathbf{A}_{u,v} \approx \mathbf{sigmoid}(\mathbf{Z}_u\mathbf{Z}_v^T)$$

## 4 評価実験

### 4.1 データセット

データセットは，ある金融機関の１ヶ月分の取引データを個社情報を排したものを利用した．データは月次

の取引額集計値で，図表２のように，仕向先・被仕向先・送金金額が記録されたものとなっている．実際は個社情報を排した ID 毎の集計取引額であるが，説明のため架空データを記している．産業テキスト情報は，表１に記したようなもので当該金融機関で各仕向先・被仕向先の口座に紐づく．中央銀行の定義している産業分類を元にしているものである．

表 2: 月次集計取引データのイメージ

| Sender | Receiver | Amount |
|---|---|---|
| Company A | Company B | 1,000,000 |
| Company A | Company C | 500,000 |
| Company A | Company D | 100,000 |
| Company E | Company A | 1,500,000 |
| Company F | Company A | 400,000 |

これらの取引データに基づき，各口座に対応する産業テキストの潜在表現をノード特徴，取引発生の有無をエッジ，月次集計取引額をエッジの重みとした有向ネットワークを構成した．本ネットワークのノード数は 170,094，エッジ数は 1,244,639，最大次数は 15,431，平均次数は 7.2863，最小次数は 1，平均クラスタリング係数は 0.0573 である．前節で言及の通り，取引額の単位は 10 (USD) のオーダーから 100,000,000 (USD) のオーダーまで様々である．

表 3: 構成された取引ネットワークの情報

| 情報 | 値 |
|---|---|
| ノード数 | 170,094 |
| エッジ数 | 1,244,639 |
| 最大次数 | 15,431 |
| 平均次数 | 7.2863 |
| 最小次数 | 1 |
| 平均クラスタリング係数 | 0.0573 |

## 4.2 タスク

提案手法の精度評価としては，リンク予測，つまり，取引発生予測を採用した．金融取引ネットワークについては，取引発生予測自体が，CRM 等の観点からも有用であること，取引ネットワークの場合は企業間関係と捉えられるノード間の関係性を見るのに適したタスクであると考えるためである．

評価指標としては，リンク予測の場合の評価として標準的であり，先行研究 [24, 15] においても用いられている ROC-AUC スコアを用いた．なお，後述するマスク率に応じた検証・企業数毎の検証では正解率・適合率・再現率等の評価指標も用いている．

## 4.3 ベースライン手法

ベースライン手法として，標準的なグラフ表現学習･深層学習手法である，node2vec [5], graph convolutional network [9], GAT [16] and GIN [20] を用いた．Jaccard coefficient[12] のような複雑ネットワークの手法も広く用いられているが，先行研究において比較されているため，今回は対象外とした [24].

node2vec [5] は shallow embeddings と呼ばれるグラフ表現学習手法のうち，最も有名なものの一つである [6]．Shallow embedding の所以は，ノード特徴が潜在表現行列のノードに対応する行になっていることである．グラフニューラルネットと比較して，まず第一に，パラメータ共有がなく，相対的に計算非効率であること，第二に，ネットワークのトポロジーだけを考慮し，ノード特徴を活用できていないこと，最後に，学習時に未知のノードに対しては潜味表現を獲得できないこと (*transductive*) であることが知られている [6].

node2vec [5] は，shallow embeddings の最も代表的なアルゴリズムである DeepWalk [14] と同様，ランダムウォークを用いてノードの潜在表現を得る手法である．DeepWalk [14] は ネットワークの各ノードについてランダムウォークを生成し，ノード系列を得，それらを Word2Vec[13] のアルゴリズムである skip-gram で学習させることでノード潜在表現を得る．

node2vec [5] は，現在のノードに対して近く，ないしは，遠くを探索するかを制御するパラメータ $p$ と $q$ を有する 2 次のランダムウォークを用いる点が DeepWalk [14] と異なる．

GCN [9] はグラフニューラルネットの中でも最も代表的なものの一つである．GCN のモデル構造は比較的シンプルで，しばしばパフォーマンスを示すことから，比較の際のベンチマークとして用いられることが多い．また，グラフ信号処理を裏付けとした数学的基礎に基づいており，スペクトルベース (spectral-based) と分類される [19]．GCN [9] は以下のように定式化される．

$$\mathbf{H}^{(k+1)} = \sigma\left(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\mathbf{H}^{(k)}\mathbf{W}^{(k)}\right)$$

$\mathbf{H} \in R^{|V|\times d}$ はノード特徴行列，$k$ はレイヤー数で今回は $k = 2$ を使用，$\mathbf{H}^{(k)} \in R^{|Hidden|\times|Input|}$ は $k$ 層目のノード潜在表現，$\tilde{\mathbf{A}}$ はセルフループを加味した隣接行列，$\mathbf{W}^{(k)} \in R^{|Hidden|\times|Input|}$ は $k$ 層目の共有パラメータ，$\sigma$ は活性化関数であり，今回 **ReLU** を使用した．また，$\mathbf{H}^{(0)} = \mathbf{H}$ であることに留意されたい．$\tilde{\mathbf{D}}$ は次数行列であり，次で定義される．

$$\tilde{\mathbf{D}}_{i,i} = \sum_j \tilde{\mathbf{A}}_{i,j}$$

GAT[16] と GIN[20] については前節にて説明の通りである．

### 4.4 実験

今回の提案手法と，各ベースライン手法を用いた場合の取引発生予測精度の比較を行う実験を行なった．

プログラムの実装においては，産業テキスト情報の処理については nltk [3]，Doc 2Vec については gensim [4]，各ベースライン手法・提案手法については PyTorch geometric [5]，評価手法については scikit-learn [6]を用いた．

node2vec[5] については，ランダムウォークのバイアスパラメータは $p = q = 1$，ランダムウォークの長さは 15，コンテキストサイズは 10，潜在表現の次元数は 128，ウィンドウサイズは 10 を用いた．

グラフニューラルネットの各ベースライン手法・提案手法については，Doc2Vec で得られる潜在表現の次元 $d$ を 300，1 層目の出力次元を 128，2 層目の出力次元を 64 とした．

損失関数には，クロスエントロピーを用いて確率的勾配降下法にて各重みパラメータ・バイアスパラメータの最適化を行い，学習率は 0.001，オプティマイザーは Adam を用いた．エポック数は 100 とし，全エッジ数の 70%を訓練データ，10%を検証データ，20%をテストデータとした．検証スコアが最良の時のテストデータに対する ROC-AUC スコアを記録した．実験結果は表 4 の通りである．

また，先行研究 [24, 23] と同様，訓練データの割合を変化させた時の評価指標の変化や，取引社数によっての違いも検証した．訓練データの割合を変化させた時の評価指標の変化については，表 5 のように node2vec[5]，GCN [9]，提案手法の 3 つで検証した．また，訓練データの割合を変化させた時の評価指標の変化については表 6 のように提案手法のみで観察した．

表 4: 実験結果：ベースライン手法との比較

| 手法 | ROC-AUC スコア (Train:80%) |
|---|---|
| node2vec [5] | 0.7984 |
| GCN [9] | 0.8772 |
| GAT [16] | 0.8344 |
| GIN [20] | 0.8973 |
| 提案手法 | **0.9497** |

[3]https://www.nltk.org/
[4]https://pypi.org/project/gensim/
[5]https://pytorch-geometric.readthedocs.io/en/latest/
[6]https://scikit-learn.org/stable/

表 5: 実験結果：訓練データ率を変化させた際の比較

| 手法 | Train:80% | Train:50% | Train:20% |
|---|---|---|---|
| node2vec [5] | 0.7984 | 0.7467 | 0.6656 |
| GCN [9] | 0.8772 | 0.8498 | 0.8015 |
| 提案手法 | **0.9505** | **0.9479** | **0.9379** |

表 6: 実験結果：取引数毎の評価

| 企業数 * | ROC-AUC | 正答率 | 適合率 | 再現率 |
|---|---|---|---|---|
| 全体 | 0.9505 | 0.7965 | 0.9816 | 0.6044 |
| 101 社以上 * | 0.8885 | 0.8326 | 0.9839 | 0.8359 |
| 51-100 社 * | 0.8731 | 0.6593 | 0.9783 | 0.5960 |
| 10-50 社 * | 0.8933 | 0.6826 | 0.9763 | 0.4747 |
| 10 社以下 * | 0.8847 | 0.6593 | 0.9476 | 0.2093 |

* 取引社数の考え方について，例えば 100 社以上とはエッジ (i,j) について i,j のいずれかが 100 社以上の時に 100 社以上としている．

## 5 結果と考察

図表に示されるように，テストデータの ROC-AUC スコアが 0.9505 と最高値となっていることが分かる．ベースライン手法と比較して，産業テキスト情報をノード特徴として用いていること，取引額を考慮した注意機構，1 層目の GAT にて近傍ノードの産業情報と取引額を考慮して 2 層目の GIN で集約関数の単射性を上げる構造が寄与していると考えられる．

node2vec の結果と，GCN・GAT・GIN といったグラフニューラルネットの違いの一つとしては，ノード特徴の活用の有無がある．これらの ROC-AUC スコアがより高い理由としては，ネットワークのトポロジーに加えて，産業テキスト情報の類似性等を加味できていることが理由と考えられる．GCN・GAT・GIN の比較については，前節で述べたように，その表現力の高さから GIN がより高い ROC-AUC スコアを示していると考えられる．

訓練データ率の変化に伴う ROC-AUC スコアは，訓練データ率の低下に伴い全体的に下落しているが，提案手法や GCN[9] の方が node2vec[5] の場合と比べて緩やかである．

また取引社数毎の検証では，ROC-AUC スコアでは大きな違いは見られないもの，再現率に着目すると取引社数の減少に応じて著しく低下していることが読み取れる．これは，取引社数の少ない企業がリンクを持ちにくく，逆も然りであるネットワーク全体の傾向を反映しており，リンクがあるケースを正しく予測出来

ていないことに由来すると考える．同様の傾向は森ら [23] の研究でも観察された．

今後の課題として，まずはノード特徴に口座特有のテキスト情報を追加することで，ノード特徴自体を一意的に設定しつつ，産業の連関性等も加味することで今回の提案手法の表現力をより高めることである．今回はデータ取得上時間的制約があったため，産業情報のみで実験を実施したが，データを増やすことだけで改善が期待できる事項である．

加えて，今回の提案手法は本質的には金融取引ネットワークの潜在表現を獲得するための手法であるため，株価・経済予測や格付といった領域にも容易に応用可能である．冒頭でも述べたように，規模の大きな金融機関の取引データは大企業から中小企業・スタートアップまで様々な業種業態の企業の膨大な取引量を有しているため，経済予測への応用が期待できる．

## 6 まとめ

本研究では，グラフニューラルネットに基づいた金融取引データの表現学習手法・取引予測について提案を行なった．各口座の特性を捉えるため，Doc2Vec により産業テキストの潜在表現を得ることで，各口座の類似性を考慮することが出来る．また，テキストの潜在表現をノード特徴とし，エッジには月次合算取引額を採用した取引ネットワークを構築し，本ネットワークに対して取引額を考慮した注意機構を有する GAT と GIN を用いることで，より表現力の高いネットワーク構造を構成し，潜在的な取引予測について従来手法を上回る精度を実現した．本提案手法は，口座についてのテキスト情報を追加するだけでノード特徴の一意性を保ちつつ，類似性も考慮可能なスケーラブルな拡張性を持ち，株価・経済予測，格付といったといった領域にも容易に応用可能である．

## 謝辞

本研究は，JST，未来社会創造事業，JPMJMI20B1 の支援を受けたものである．

## 参考文献

[1] Ali B. Barlas, Seda Guler Mert, Berk Orkun Isa, Alvaro Ortiz, Tomasa Rodrigo, Baris Soybilgen, and Ege Yazgan. Big data information and nowcasting: Consumption and investment from bank transactions in turkey. 2021.

[2] Jun Chen and Haopeng Chen. Edge-featured graph attention network. 2021.

[3] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. Metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 135–144, New York, NY, USA, 2017. Association for Computing Machinery.

[4] Janina Engel, Michela Nardo, and Michela Rancan. *Network Analysis for Economics and Finance: An Application to Firm Ownership*, pages 331–355. Springer International Publishing, Cham, 2021.

[5] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 855–864. Association for Computing Machinery, 2016.

[6] William L. Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159, 2020.

[7] Anish Khazane, Jonathan Rider, Max Serpe, Antonia Gogoglou, Keegan Hines, C. Bayan Bruss, and Richard Serpe. Deeptrax: Embedding graphs of financial transactions. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 126–133, 2019.

[8] Thomas N. Kipf and Max Welling. Variational graph auto-encoders. In *NIPS 2016 Workshop on Bayesian Deep Learning*, BDL '16, 2016.

[9] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17, 2017.

[10] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–1188–II–1196. JMLR.org, 2014.

[11] Xiaoxiao Li, Joao Saude, Prashant Reddy, Manuela Veloso, JP Morgan, and AI Research. Classifying and understanding financial data using graph neural network. In *The AAAI-20 Workshop on Knowledge Discovery from Unstructured Data in Financial Services*, AAAI-KDF '20, 2020.

[12] Linyuan Lu and Tao Zhou. Link prediction in complex networks: A survey. *Physica A*, 390(6):1150–1170, 2011.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations*, ICLR '13, 2013.

[14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 701–710, New York, NY, USA, 2014. Association for Computing Machinery.

[15] Valentina Shumovskaia, Kirill Fedyanin, Ivan Sukharev, Dmitry Berestnev, and Maxim Panov. Linking bank clients using graph neural networks powered by rich transactional data: Extended abstract. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 787–788, 2020.

[16] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*, ICLR '18, 2018.

[17] Jianian Wang, Sheng Zhang, Yanghua Xiao, and Rui Song. A review on graph neural network methods in financial applications. 2021.

[18] Boris Weisfeiler and A. A. Lehman. A Reduction of a Graph to a Canonical Form and an Algebra Arising During This Reduction. *Nauchno-Technicheskaya Informatsia*, Ser. 2(N9):12–16, 1968.

[19] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

[20] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *Proceedings of the 7th International Conference on Learning Representations*, ICLR '19, pages 1–17, 2019.

[21] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, volume 31 of *NeurIPS '18*, pages 5165–5175. Curran Associates, Inc., 2018.

[22] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[23] 森 正和, 與五澤 守, and 工藤 剛. Gcn による取引関係グラフからの企業の特徴量抽出. In **第** *24* **回人工知能学会金融情報学研究会** *(SIG-FIN)*, SIG-FIN-024, 2019.

[24] 藤塚 理史 and 工藤 剛. グラフエンベディングを活用した潜在取引関係予測. In **第** *24* **回人工知能学会金融情報学研究会** *(SIG-FIN)*, SIG-FIN-024, 2019.