

Manufacturing Service Capability Prediction with Graph Neural Networks

Yunqing Li^a, Xiaorui Liu^b, Binil Starly^c

^a*Edward. P. Fitts Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, 27695, NC, USA*

^b*The Department of Computer Science, North Carolina State University, Raleigh, 27695, NC, USA*

^c*School of Manufacturing Systems and Networks, Arizona State University, Mesa, 85212, AZ, USA*

Abstract

In the current landscape, the predominant methods for identifying manufacturing capabilities from manufacturers rely heavily on keyword matching and semantic matching. However, these methods often fall short by either overlooking valuable hidden information or misinterpreting critical data. Consequently, such approaches result in an incomplete identification of manufacturers' capabilities. This underscores the pressing need for data-driven solutions to enhance the accuracy and completeness of manufacturing capability identification. To address the need, this study proposes a Graph Neural Network-based method for manufacturing service capability identification over a knowledge graph. To enhance the identification performance, this work introduces a novel approach that involves aggregating information from the graph nodes' neighborhoods as well as oversampling the graph data, which can be effectively applied across a wide range of practical scenarios. Evaluations conducted on a Manufacturing Service Knowledge Graph and subsequent ablation studies demonstrate the efficacy and robustness of the proposed approach. This study not only contributes a innovative method for inferring manufacturing service capabilities but also significantly augments the quality of Manufacturing Service Knowledge Graphs.

Keywords: Node Classification, Link Prediction, Graph Neural Network, Manufacturing Service Capability, Manufacturing Service Knowledge Graph

1. Introduction

1.1. Background and Motivation

Recent global crises, including the pandemic, the shift towards near-shoring in manufacturing, and escalating geopolitical tensions, have significantly impacted supply chains across all major industries [1]. Small manufacturing companies, which are the backbone of most manufacturing economies [2], have been disproportionately affected by supply chain disruptions, with delays in manufacturing and shipping, and shortages across the board [3]. The crises prevent these enterprises, which typically rely on interpersonal connections and regional web directories to look for new business prospects. Therefore, it is vital to implement advanced and effective tactics for the identification of small manufacturing firms and their capabilities to assist them in being discovered and vetted into the global supply chains[4].

1.2. Challenges in Identifying Manufacturing Service Capabilities

Manufacturing Service Capability (MSC) [5] is the ability of manufacturing enterprises to effectively integrate and configure various resources, reflecting their proficiency in completing specific tasks. This comprehensive concept spans the entire life-cycle of manufacturing, including design, simulation, and production capabilities. It's evidenced in various forms, from industry-recognized certifications like Capability Maturity Model Integration (CMMI), indicating a commitment to quality and consistency, to specific manufacturing processes such as drilling and milling. MSC also covers the adaptability of manufacturers to serve different industries, like medical industry and automotive industry, and their capacity to work with diverse materials, such as plastics and steel.

Traditional methodologies constrain the scope of MSC identification, restricting it to a limited scale. Within the business sector, platforms such as Thomasnet [6] and Google Maps necessitate that manufacturers independently catalog their competencies. This self-reporting approach slows down the expansion of manufacturing business networks. In the academic context, there is always an assumption that MSC data is pre-defined and uniformly structured for supply-demand matching [7, 5]. However, in reality, a significant portion of MSC data, particularly from smaller, local businesses, is derived from their distinct and varied website structures. To effectively address this, there's a critical need to develop a universally adaptable method that can autonomously and efficiently identify MSCs on a much broader scale.

Existing approaches to automatically identifying MSC [8] rely on keyword matching or Natural Language Processing (NLP). Keyword matching involves identifying keywords such as “CNC machining” or “injection molding” on a manufacturer’s website. NLP-based methods analyze relevant information from textual data sources, such as websites, catalogs, or documents. For instance, Named Entity Recognition (NER) can be adopted to identify specific MSCs [9]. The widespread application of various methods in identifying and categorizing businesses and services has greatly contributed to the optimization of supply chains and decision-making processes in the manufacturing domain [10, 11].

However, these methods often suffer from two critical limitations: wrong identification and misidentification. Wrong identification occurs when a business or service is incorrectly categorized. For example, a company selling Computer Numerical Control (CNC) machines may be incorrectly identified as a CNC machining provider, even if it cannot provide CNC machining services [12]. This misclassification can lead to incorrect assumptions about the company’s capabilities, resulting in inappropriate decisions and actions by other parties, such as suppliers or potential customers. Misidentification happens when a business or service’s capabilities are not fully recognized or understood. For instance, manufacturers skilled in titanium processing might be integrated into the supply chain for aircraft or medical device production, given that titanium is a frequently used material in the aerospace industry [13] or the medical industry [14]. However, if these capabilities are not accurately identified, the manufacturer may be overlooked for contracts or collaborations in these sectors. It is crucial to uncover, integrate, and utilize hidden information in manufacturing data sources to aid decision-making, and risk management, and gain insights into the flow of goods, materials, and resources through the supply chain.

1.3. Objectives

Recently, Knowledge Graphs (KGs) and Graph Neural Networks (GNNs) are of paramount importance in data representation and knowledge extraction [15, 16]. KGs effectively manage complex data with interconnected entities, offering scalability and ease of updates. GNNs complement this by adeptly learning from data’s complex relationships and patterns, particularly useful in graph-structured data. In the realm of MSC identification, combining KGs and GNNs leads to a more adaptable, precise, and scalable approach. This integration is key in accurately distinguishing the unique characteristics

of various manufacturers, thus reducing misidentifications. This synergy has propelled the development of GNN-based systems for digital supply chain representation in manufacturing, as further detailed in recent research [17].

This paper seeks to harness the power of KGs and GNNs to enhance the discoverability of small manufacturing businesses and their MSCs by prospective clients. The identification of MSCs enabled by KG and GNNs significantly aids startups, entrepreneurs, and researchers in gaining insights from manufacturing data to select potential business partners. In this work, we employ an automated approach to construct a Manufacturing Service Knowledge Graph (MSKG) as introduced in [18], serving as the foundational framework for our analysis. An MSKG is comprised of two distinct node categories: “Manufacturer” and “Service”. It encapsulates two types of relational links: one that establishes connections between “Manufacturer” and “Service” nodes, and another that delineates affiliations amongst “Service” nodes. The central challenge lies in effectively modeling the problem using a graph-based approach while concurrently enhancing performance through the strategic application of feature engineering methods. The task of modeling the problem is not only selecting the most appropriate architecture but also modifying the setting of models and data especially for addressing the business objective. It also includes designing effective feature representations that can significantly elevate the overall predictive power and generalizability of our method. The main contributions highlighted in this paper are:

1. We introduce a methodology to deduce MSCs by graph-based node classification, offering unique advantages in the realm of graph-based information inference.
2. We propose a feature engineering approach tailored for MSKGs that enhances the performance of graph-based analysis by aggregating information from nodes’ neighborhoods.
3. We propose to mitigate the issue of node class imbalance in real-world heterogeneous graphs by generating synthetic edges and nodes, which can be generalized to various practical scenarios.

An example to identify MSC using our approach is shown in Figure 1. Suppose we aim to determine if a manufacturer possesses the capability to cater to the automotive industry and handle copper processing. Initially, an MSKG is formulated by gleaning textual data from different manufacturing data sources. Then, the graph’s nodes and edges are synthesized to

balance the node classes of two distinct node classification objectives: “Does the manufacturer serve the automotive industry?” and “Does the manufacturer process copper?”. Following this, we aggregate information from neighboring nodes within the graph. In the final phase, we employ GNN algorithms to train two distinct node classifiers. These classifiers’ outcomes then help ascertain the manufacturer’s capability in the automotive sector and copper processing. The refinement in the second and third steps ensures enhanced precision in node classification, leading to more accurate insights into manufacturing competencies.

For the rest of the paper, we review the related work in Sec. 2. The studied problem is defined and the details of the proposed method are presented in Sec. 3. In Sec. 4, the experiments are conducted to demonstrate the effectiveness of our method. In Sec. 5, the limitations and future work of our method are concluded.

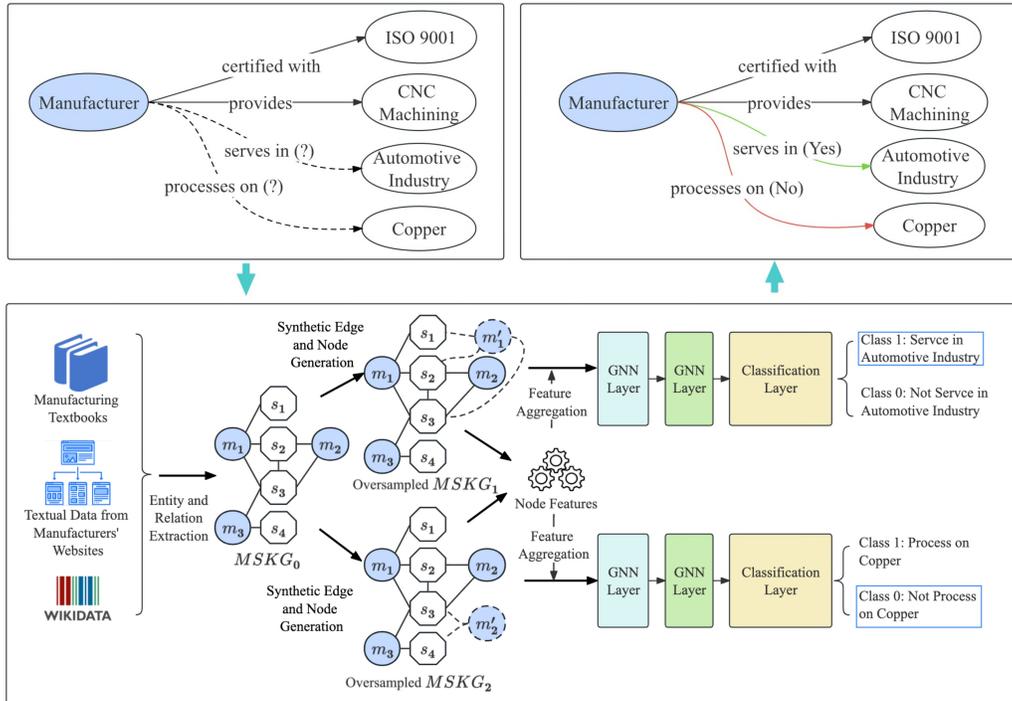


Figure 1: An Example of Identifying MSC

2. Related Work

2.1. Manufacturing Data Sourcing and Inference

Currently, there are various methods for manufacturing data sourcing. Key term matching and Term Frequency-Inverse Document Frequency (TF-IDF) [19] help with extracting essential words from manufacturing textual data [20]. K-means and Latent Dirichlet Allocation (LDA) algorithms are used for document clustering and topic modeling in manufacturing text data mining [21]. When handling a large amount of manufacturing textual data, several forms of NLP have been constructed through different kinds of embedding representation schemes. For example, Word2Vec [22], Doc2Vec [23], Bidirectional Encoder Representations from Transformers (BERT) [24], OpenAI embedding [25] are widely used in dealing with content embedding in the manufacturing domain. Named Entity Recognition is applied to identify goods, materials, and resources in different parts of the supply chain [26].

With the development of the digital supply chain, applications of data-driven inference in the manufacturing domain have been evolving in recent years. There are inference systems constructed which are related to evaluating management of supply chain performance [27], supply chain risks [28] or downstream demand inference [29]. Manufacturing information inference is inferring information about a manufacturing process or service based on available data and evidence, which can help suppliers to optimize their cost-effectiveness, and customer satisfaction as well as be noticed and considered by more potential clients. Villas-Boas [30] conducts inference on vertical relationships between manufacturers and retailers. A framework to enable the reusability of manufacturing knowledge through inference rules applied to manufacturing ontologies is introduced in [31]. Cao et al. [5] propose a model for estimating MSCs, specifically emphasizing machining and production services. Another framework for canonicalizing MSC models is proposed using the reference ontology [32]. But none of them is about gaining insightful inference on the latent relationships between manufacturers and various manufacturing services.

2.2. Knowledge Graph Construction from Unstructured Data

Constructing a KG from unstructured data is more challenging due to the inherent difficulty in accurately extracting entities and relationships from such data. In the healthcare domain, Health KG Builder is introduced by [33], which can be used to construct disease-specific and extensible health

KGs from unstructured sources. Zhu et al. [34] focuses on the application of KG in the traditional geological field and proposes a novel method to construct KG from complex geological unstructured data. Li and Starly [35] presents a bottom-up approach to parse through unstructured text available on the websites of small manufacturers across the United States to construct a MSKG [18].

2.3. Current Graph Neural Networks and Downstream Applications

GNNs are able to effectively extract complicated, non-linear relationships in datasets. One type of GNN is Message-passing neural networks (MPNNs) [36] which utilize a message-passing mechanism to aggregate information from neighboring nodes in the graph and accordingly update the delineation of each node. GraphSAGE [37] and graph convolutional networks (GCNs) [38], SGC-GNNs [39] and EdgeConv [40] are MPNNs. APPNP [41] incorporates personalized PageRank scores into the propagation process to improve prediction accuracy, making it a specialized variant within the MPNNs. Another type is Graph Generative Models (GGM) which are used for generating synthetic graph structures based on probabilistic models such as Markov random fields or Bayesian networks. Instances of GGMs include graph recurrent neural networks (GRNNs) [42] and graph generative adversarial networks (GraphGANs) [43]. Last is Graph Transformer Models [44] which are built on the transformer framework, such as graph transformers (GTrs) [45] and graph attention transformers (GATs) [46].

With the rapid development of GNNs, they have demonstrated state-of-the-art performance on diverse graph downstream applications. Three typical tasks are node classification [47], link prediction [48] and graph classification [49]. Node classification is when we have a KG with a certain ratio of nodes labeled, a classifier is trained on those labeled nodes so that it can classify the unlabeled nodes in the graph. Class imbalance has been an essential challenge in node classification.

2.4. Class Imbalance

Class imbalance has been a vital research topic in machine learning for years. Numerous tasks, like fraud detection [50] and sentiment analysis [51] suffer from class imbalance. The main approach is changing the data itself or the way the model is used to solve class imbalance, such as undersampling and oversampling. Undersampling is a technique that reduces the number of instances in the majority class so that it is more evenly represented with the

minority class. When removing instances from the majority class, we may lose essential information in those instances which could negatively affect the performance of the models. Oversampling is duplicating existing instances in minority classes which can mitigate class imbalance but result in overfitting in the data training process. Many variations of oversampling have been proposed to improve its effectiveness. Synthetic Minority Over-sampling Technique (SMOTE) [52] is one of the most popular over-sampling methods. It involves generating synthetic minority class samples by interpolation to multiply the representation of the minority class. To address node class imbalance issue in a graph, current methods such as GraphSMOTE [53] and GATSMOTE [54] are proposed, which leverage synthetic data generation to balance class distribution, enabling models to better handle underrepresented classes.

2.5. Feature Engineering

Feature engineering is a fundamental component of machine learning that profoundly influences model performance. Data transformation techniques encompass one-hot encoding, feature scaling, etc., which are crucial for managing data heterogeneity and scaling issues. Dimension reduction approaches, such as t-distributed stochastic neighbor embedding (t-SNE) [55], Principal component analysis (PCA) [56] and feature selection, aid in managing high-dimensional data and curating a subset of the most informative features. Additionally, domain-specific knowledge often guides the creation of task-specific features [57, 58]. The choice of feature engineering method depends on the data type and problem domain. Hence, it is essential to select and customize the strategy of feature engineering based on the characteristics of the KGs in the manufacturing domain to enhance the model’s predictive ability in identifying MSCs.

3. Methodology

3.1. Problem Statement

The objective of our paper is to identify if a manufacturer is capable of a specific potential manufacturing service, which can be converted to a graph-based information inference problem. The reasons are as follows: first, the representations of MSCs is easy to show in graphs. Manufacturers, as well as their services (like machining and automotive industry), can be represented as nodes, while the relationships between them, indicating manufacturers’

service capability, can be represented as edges. Second, graph has scalability and flexibility. As more data about new manufacturers and their services are gathered, the graph can be continuously expanded. For instance, if a new service becomes relevant to the automotive industry, it can be easily incorporated into the graph, allowing for dynamic updating of inferences. Last but not least, the inference can be conducted through connectivity in the graph. If a new manufacturer provides both machining and 3d printing, and these capabilities are commonly found among manufacturers serving the automotive industry (as seen in the graph), then it’s likely that this manufacturer also has potential in the automotive domain. This conclusion can be inferred based on the proximity and connectivity patterns in the graph.

In this study, our objective is to infer MSCs of manufacturers, which is achieved by utilizing the connections identified within the MSKG. Our approach offers a detailed assessment of the manufacturers’ expertise and proficiency across diverse manufacturing domains. Within the MSKG, the connections between manufacturers and services are classified into four key categories: “provide”, “certified with”, “serve in”, and “process on”. These categories are directly linked to the targeted manufacturing services, encompassing areas such as manufacturing processes, certifications, industries, and materials.

$G = \{M, S, A, F\}$ is the MSKG used for graph-based information inference. The attributed network is depicted by various elements, as follows:

- $M = \{m_1, \dots, m_n\}$ is a set of n nodes, where each node represents a unique manufacturing business. Within the graph, these nodes can be oversampled, as various manufacturer nodes can potentially link to a group of same manufacturing service nodes. M_t is a subset of M used in training.
- $S = \{s_1, \dots, s_i\}$ is a set of i manufacturing service nodes, where each node represents a unique entity so they can’t be oversampled within the graph. S_t is a subset of S used in training.
- $A \in \mathbb{R}^{p \times p}$ is the adjacency matrix of G , $p = n + i$. A_{km} is essentially a binary indicator that tells you whether the vertices corresponding to row k and column m are directly connected in G .
- $F \in \mathbb{R}^{p \times d}$ is the node attribute matrix, where $F[j, :] \in \mathbb{R}^{1 \times d}$ is the node attributes of node j . d is the dimension of the node attributes.

- $Y \in \mathbb{R}^p$ represents the class information for nodes in G . Y_t is a subset of Y for training. The task of node classification is to predict whether a given node represents a manufacturer as well as it is connected to a designated manufacturing service node. The manufacturer nodes that have direct relationships with these services are labeled as 1, and all other nodes are labeled as 0.
- $C = \{c_1, c_2\}$, $|c_1|$ is the size of majority node class and $|c_2|$ is the size of minority node class.
- Imbalance ratio, $\frac{|c_2|}{|c_1|}$ determines the level of node class imbalance.

In the methodology, the inference of MSC is modeled as a GNN-based node classification problem. The proposed method is composed of four subsections: 3.2 Problem Modeling. It introduces why and how to model our problem into a node classification task over MSKGs. 3.3 Synthetic Edge and Node Generation. It leverages random sampling and stratified sampling on edges associated with minority classes to balance the node classes to obtain an augmented graph. 3.4 Feature Aggregation. It aggregates and encodes neighbor nodes' information through Doc2Vec and t-SNE and combines them with original node features to form the node attribute matrix of the augmented graph. 3.5 GNN Classification. It utilizes a GraphSAGE classifier to predict binary class labels of the nodes in the oversampled augmented graph. The following parts provide further details on each step.

3.2. Problem Modeling

The process of constructing an MSKG serves as the foundation for our graph-based inferential procedures. The construction of the MSKG is carried out in three phases. First, a web-scraping process is initiated to gather the text content from the manufacturers' websites in the United States to create manufacturer nodes in M . Second, we identify manufacturing service nodes in S as well as the edges between them such as subclass relationships from Wikidata and standard manufacturing textbooks. Third, after text pre-processing, keyword matching is conducted between M and S to obtain the relationships. G_0 denotes the initial graph constructed. The basic schema structure of an MSKG includes two entity types: manufacturer name, M ; 2) manufacturing service, S . S encompasses the manufacturing process, relevant certifications, materials utilized, and the industries in which the

manufacturer operates. Equation (1) is used to initialize the node attribute matrix, designated by $F[j]$:

$$F[j] = \begin{cases} 0 & \text{if } j \in M \\ 1 & \text{if } j \in S \text{ and } j \text{ is an industry} \\ 2 & \text{if } j \in S \text{ and } j \text{ is a service} \\ 3 & \text{if } j \in S \text{ and } j \text{ is a material} \\ 4 & \text{if } j \in S \text{ and } j \text{ is a certification} \end{cases} \quad (1)$$

G is obtained from G_0 by excluding the target manufacturing service nodes and the edges directly connected to them. This step is essential for node classification. For instance, if we need to infer which manufacturer serves the medical industry, we initially mask the correct answer within the graph. By excluding the “Medical Industry” node and its corresponding manufacturer relationships, we can partition the modified graph: some nodes for training a graph-based classifier and others for subsequent prediction and evaluation.

To deduce the MSC from the MSKG, we explore both link prediction and node classification approaches to tackle the challenge. Node classification is selected as the primary method for our study, with link prediction serving as the comparative approach. The reason is that in MSKGs, the number of links is typically 15 times greater than the number of nodes. Given this, computing predictions for every potential link can be computationally intensive. It’s essential to save on computational costs, especially for the dynamic nature of MSKGs. To accomplish our primary objective, node classification aims to discern whether a node represents a manufacturer node and is directly linked to a manufacturing service node, while link prediction is designed to predicting the relationships between manufacturer nodes and a designated manufacturing service node. On the other hand, node classification aims to discern whether a node represents a manufacturer and is directly linked to a manufacturing service node.

3.3. Synthetic Edge and Node Generation

Possible ways to oversample the graph could be through node duplication, or adapt SMOTE to the graph data, like GraphSMOTE. These methods are sub-optimal for our graph due to the following reasons: 1) Since each node in S is unique, directly oversampling entities in S may distort the overall structure of the MSKG, leading to a misrepresentation of the relationships

between nodes. Simply oversampling entities in the graph may easily cause overfitting during the training process as well. 2) Since nodes of M and S can be classified into the same node classes, GraphSMOTE or performing SMOTE in the node embedding space may result in the creation of synthetic nodes in the minority class that are neither similar to entities in M and S . 3) Both SMOTE and GraphSMOTE, regardless of whether synthetic nodes are generated through interpolation in the raw feature space or embedding space, fail to consider the diversity within the same node class in a heterogeneous graph. This implies that within a single node class, there may be different node types, leading to a situation where certain types can be oversampled in while others are not.

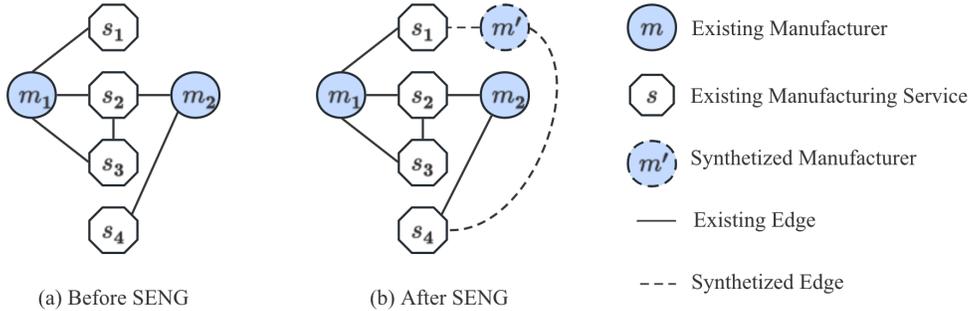


Figure 2: An Example of SENG

Hence, we propose a heuristic approach, Synthetic Edge and Node Generation (SENG), to conduct graph oversampling on an MSKG shown in Figure 2. SENG-oversampling utilizes bagging of entities and edges as a means of mitigating the overfitting that is introduced through oversampling in the training process. To generate a synthetic node along with all the edges associated, there are six steps: 1) To minimize redundancy in the generation of synthetic nodes, a random selection α is made from the set $[2, 3, 4]$. This choice is deliberately constrained to values greater than 1 and less than 5, thus ensuring a diversified range of selections; 2) Randomly sample α elements from M with replacement to obtain a subset of M , M_{sub} ; 3) Use M_{sub} to get corresponding A_{sub} ; 4) Use A_{sub} to obtain a subset of S , S_{sub} ; 5) Randomly sample $\frac{1}{\alpha}$ from S_{sub} ; 6) Create a synthetic node m' along with synthetic edges connecting m' with entities in S_{sub} . After this generation process, node classes are more balanced. However only applying SENG may cause severe

overfitting during the training process. To improve training performance, we introduce the feature aggregation process in the following step.

3.4. Feature Aggregation

Word embedding [59], a technique based on artificial neural networks, allows textual data to be represented in a way that can be understood by computers, which is achieved by representing words as vectors. Doc2Vec, an extension of the Word2vec model, is a word embedding method that generates a vector representation of a paragraph, allowing for the detection of semantic similarity. This vector representation captures the meaning and context of the words in the document, as well as their order and arrangement. T-SNE is a machine learning algorithm that can project high-dimensional data into a lower-dimensional space as well as preserve the structure of the data. Both Doc2Vec and t-SNE are used in the node feature generation process.

Feature Aggregation (FA) aims to enhance the performance of node classification by enriching node features. Through SENG, G is transformed into \tilde{G} , with corresponding changes occurring in \tilde{M} , \tilde{S} , \tilde{A} , \tilde{F} , \tilde{C} and \tilde{Y} . Textual information from the neighboring nodes, which are their names, are collected to populate the representation of manufacturer node features in \tilde{F}' . D , a dictionary, which is a built-in data structure that allows the storage and retrieval of key-value pairs. In this paper, D contains all the entities in \tilde{M} as keys and their first-order related names of neighbouring nodes in \tilde{S} as values. Each value, indexed by a key, is a paragraph within a corpus. Each paragraph corresponds to an entity in \tilde{M} . The vectors of paragraph are learned by Doc2Vec such that each paragraph is mapped to a high dimension space, feature matrix F_1 . Dimensionality reduction via t-SNE is performed to project high dimensional vectors into 2-dimensional space and generate the feature matrix F_2 . In Equation (2), each row in F_2 is integrated with $\tilde{F}[j]$ where $\tilde{F}'[j, :] \in \mathbb{R}^{1 \times 3}$ is the updated node features of node j .

$$\tilde{F}'[j, :] = \begin{cases} \tilde{F}[j] + F_2[j, :], & \text{if } j \in M \\ \tilde{F}[j] + [0, 0], & \text{otherwise} \end{cases} \quad (2)$$

3.5. GNN Classification

This step is to train a GNN-based node classification model on the augmented graph $\tilde{G} = \{\tilde{M}, \tilde{S}, \tilde{A}, \tilde{F}'\}$. The binary labels assigned to each node are determined by assessing whether they establish direct connections with

the target MSCs within G_0 . We partition our graph data for training, validation, and testing. Additionally, we employ stratified splitting for the augmented graph data and incorporate it into the training data. The class imbalance problem is addressed in 4.2 through SENG, resulting in equal class support while using a weighted cross-entropy loss in model training.

GNN-based node classification employs a specialized deep learning framework to categorize or label individual nodes within a given graph. A two-layer GraphSAGE is adopted to derive node embeddings. An output layer is then appended, processing the feature vectors from the final GraphSAGE layer to assign node classification labels. At the first layer, the aggregated embedding $h_1^{N(j)}$ at node j , based on the set of sampled neighbor nodes $N(j)$, is concatenated with the node’s attributes $\tilde{F}'[j, :]$ from \tilde{G} . The equation to generate aggregated information $h_1^{N(j)}$ at node j is represented as Equation (3). Passing and concatenating the aggregated information with node attributes $\tilde{F}'[j, :]$ from \tilde{G} , a node embedding of j at the first layer is expressed as Equation (4).

$$h_{N(j)}^1 = \text{MEAN} \left(\{ \tilde{F}'[u, :] \mid \forall u \in N(j), \forall j \in V \} \right) \quad (3)$$

$$h_j^1 = \text{ReLU} \left(W^1 \cdot \text{CONCAT}(\tilde{F}'[j, :], h_{N(j)}^1) \right) \quad (4)$$

$W_k (k = 1, 2, 3)$ refers to the weight parameters of each layer. The mean aggregator is applied in the aggregated information equation at each layer. Similarly, at the second layer, the aggregated neighbor nodes’ embedding $h_{N(j)}^2$ at node j is combined with node j ’s embedding from the previous layer, as depicted in Equations (5) and (6). ReLU is used as the activation function in generating node embeddings at both layers.

$$h_{N(j)}^2 = \text{MEAN} \left(\{ h_u^1, \forall u \in N(j), \forall j \in V \} \right) \quad (5)$$

$$h_j^2 = \text{ReLU} \left(W^2 \cdot \text{CONCAT}(h_j^1, h_{N(j)}^2) \right) \quad (6)$$

In addition, the second layer is appended by a sigmoid layer to predict node labels as expressed in Equation (7). P_j is the probability that node j is related to a certain manufacturing service. The classifier is finally optimized by cross-entropy loss as shown in Equation (8). V is the union of $\tilde{M} \cup \tilde{S}$.

$$P_j' = \text{Sigmoid} \left(\text{ReLU} \left(W^3 \cdot \text{CONCAT}(h_j^2, H^2 \cdot \tilde{A}[:, j]) \right) \right) \quad (7)$$

$$L_{\text{node}} = \sum_{j \in V} (1(Y_j == 1) \cdot \log(P_j)) \quad (8)$$

The predicted label of node j , Y'_j , is set as:

$$Y'_j = \begin{cases} 1, & \text{if } P_j > 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Hence, the objective of our framework is to minimize L_{node} . \emptyset is the parameter of the node classifier.

$$\min_{\emptyset} L_{\text{node}} \quad (10)$$

3.6. Training Algorithm

The procedure for executing our framework is outlined in Algorithm 1. From Line 1 to Line 8, the graph is augmented by SENG. From Line 9 to Line 15, node features are enriched and integrated by FA. From Line 16 to the end, a GraphSAGE classifier is trained on the augmented graph \tilde{G} . Oversampling Scale (OS) determines how much oversampling is applied to the minority class. For example, if OS is set to 1, it means the number of samples in the minority class is doubled by generating synthetic samples until the class distribution is more balanced.

The design of our algorithm has four advantages: 1) The primary advantage is its dynamic adaptability, facilitated by the automated updating of MSKG combined with the utilization of GraphSAGE for inductive learning, which allows for the continual integration of evolving manufacturing service capabilities. 2) It is simple to implement SENG over a minority class on a heterogeneous graph or a bipartite graph without distortion appearing during the oversampling process. 3) The utilization of FA significantly enhances the representation of nodes within the graph and subsequently improves the performance of node classification. 4) SENG and FA are applied independently, and either can be removed from the whole algorithm if necessary.

4. Experiments

In this section, we conduct experiments to assess the effectiveness of the proposed method for inferring relationships between manufacturers and manufacturing services. In the experimental evaluation, both real-world datasets

Algorithm 1: Node Classification on MSKG

Data: $G = \{M, S, A, F, C, Y\}$

Result: Predicted node labels Y'

```
1 if  $\frac{|c_2|}{|c_1|} \leq 0.7$  then
2   Number of Oversampling nodes  $No = (1 + OS) \cdot c_2$ ;
3   for  $i = 1$  to  $No$  do
4      $M_i =$  entity set of  $\alpha$  random selections from  $M$ ;
5      $S_i =$  node set of entities from  $S$  which directly relate to nodes
        in  $M_i$ ;
6      $S'_i =$  randomly sample  $\frac{1}{\alpha}$  of the elements from  $S_i$ ;
7      $S'_i = \text{set}(S'_i)$ ;
8     Connect synthetic node  $i$  to elements in  $S'_i$ , update  $G$  to
        augmented  $\tilde{G}$ ;
9 for node  $q$  in  $M'$  do
10    $S_q =$  node set of entities from  $S'$  which directly relate to node  $q$ ;
11    $D[q].\text{append}(S_q)$ ;
12 Use  $D$  to train Doc2Vec, obtain  $F_1$ ;
13  $F_2 \leftarrow \text{t-SNE}(F_1)$ ;
14 for node  $j \in \tilde{M} \cup \tilde{S}$  do
15   Generate  $\tilde{F}'[j, :] \in \mathbb{R}^{1 \times 3}$  based on Equation (2);
16 Randomly initialize  $W_k$ ;
17 while Not Converged do
18   Learn node embeddings according to Equation (3) - (6);
19   Update the model using  $L_{\text{node}}$ ;
20 Return trained node classifier;
```

and the datasets augmented from real-world datasets with imbalanced class distributions are utilized. Specifically, the following questions are addressed in this study:

1. How does the node classification result in the performance of MSC identification compared with link prediction?
2. Is our method pervasive to a different classifier structure?
3. How does the utilization of FA in our method result in performance compared with other feature engineering approaches?
4. How does the performance of our method vary under different OSs in the imbalanced node classification task?
5. Is our method pervasive to different imbalance ratios?

The experimental settings, including datasets, baselines, configurations and evaluation metrics are presented in 4.1. Question (1)-(5) are addressed in 4.2 - 4.6 respectively.

4.1. Settings

4.1.1. Datasets

We conduct experiments based on a MSKG to identify if manufacturers are capable of the following manufacturing services: “Machining”, “Copper”, “Heat Treatment” and “ISO 9001”. The MSKG [60], containing 7,052 nodes and 112,873 relationships, has been constructed by keyword matching between textual data from over 7,000 manufacturers’ websites in the United States as well as common manufacturing services, which are selected from Wikidata and the manufacturing textbooks.

The task of node classification within the MSKG is to predict whether a given node represents a manufacturer as well as it is connected to a certain manufacturing service or not. The manufacturer nodes that have direct relationships with these services are labeled as 1, and all other nodes are labeled as 0. Once this labeling has been performed, the direct relationships between the manufacturer nodes and the selected manufacturing services are removed from the graph. Node class distributions of the datasets regarding selected manufacturing services are shown in Table 1. Classes in these datasets follow a genuine imbalanced distribution. For each dataset, we split graph data for training, validation and testing following an 8:1:1 ratio. In 4.5 and 4.6, the datasets generated from original datasets are varied by changing the oversampling ratio and imbalance ratio to analyze the performance of the proposed method under different imbalanced scenarios.

Table 1: Node Class Distributions of the Datasets

Datasets	Majority Class	Imbalance Ratio (%)
Machining	1	58.66
Copper	0	19.00
Heat Treatment	0	27.30

The link prediction on the MSKG is to predict whether an edge between a manufacturer node and a designated service node exists or not. The edges between the manufacturer nodes and the selected manufacturing services are split following an 8:1:1 ratio for training, validation and testing. The rest of the edges in the graph are added to the training data.

4.1.2. Baselines

The performance of our proposed method is evaluated in comparison to alternative solutions for identifying MSCs on the MSKG. These solutions can be used to establish a benchmark for highlighting the improvement or added value of our work. For link prediction tasks, not only GraphSAGE, GCN, EdgeConv, SGC, and APPNP are used as GNN classifiers and compared, but also FA component is utilized on GraphSAGE and GCN to see if integrating FA can enhance the performance of link prediction. For node classification tasks, we assess the performance of a GraphSAGE Classifier in comparison to the following methods:

- *GraphSAGE*: A GraphSAGE node classifier trained and tested on an imbalanced dataset without any pre-processing or balancing techniques applied.
- *SENG – GraphSAGE*: Utilizes the SENG part of our method by generating synthetic nodes and edges in the node classification training process to mitigate class imbalance issues but excludes the FA component.
- *FA – GraphSAGE*: Utilizes the FA component of our method to improve the performance of node class classification on an imbalanced dataset by enriching node features.

- *SF – GraphSAGE*: Utilizes both SENG and FA components of our method to improve the performance of node class classification on an imbalanced dataset.

4.1.3. Configurations

All experimental trials were conducted within a consistent Google Colab environment, employing the ADAM optimization algorithm [61] for training the models. The learning rate for all models was set to 0.01. All models were trained until convergence, with the maximum training epoch set to 415. The OS was fixed at 1 for all datasets in 4.2.

4.1.4. Evaluation Metrics

We adopt two criteria for evaluating imbalanced classification, in line with previous studies: Area Under the Receiver Operating Characteristic curve (AUC-ROC) and Area Under the Precision-Recall curve (AUC-PR). AUC-ROC metric measures the ability of a classifier to distinguish between the positive and negative classes by comparing the true positive rate and false positive rate. AUC-PR illustrates a model’s ability to distinguish between positive and negative classes by comparing precision and recall.

4.2. Overall Performance of MSC Identification

The investigation of the overall performance of MSC predictions is conducted to answer Question (1). To mitigate the effects of randomness, each experiment is repeated on multiple occasions, with a minimum of three iterations. For link prediction tasks, according to Table 2, GCN consistently performs well across most evaluation metrics and datasets, which indicates that GCNs are a robust choice for link prediction tasks, as they can capture complex relationships in the graph structures effectively. FA-GraphSAGE and FA-GCN generally outperform their non-feature-aggregated counterparts. This suggests that incorporating additional features into the graph-based models can lead to improvements in link prediction. For node classification tasks, according to Table 3, the combination of both SENG and FA components in SF-GraphSAGE leads to the highest AUC-ROC and AUC-PR scores in most cases. This demonstrates that utilizing both SENG and FA can significantly improve performance on the imbalanced datasets. Besides, it is noticed that utilizing SENG without incorporating FA may not yield an improvement in performance due to the insufficiency of node features. On the contrary, using FA independently can greatly improve evaluation results

from the baseline by augmenting node features. Additionally, extra evaluations are conducted across varying Train-Test-Valid Ratios (8:1:1, 7:1.5:1.5, 6:2:2, 5:2.5:2.5) using two node classification models: GraphSAGE and FA-GraphSAGE. It is consistently observed that FA-GraphSAGE outperformed GraphSAGE across all metrics for each Train-Test-Valid Ratio. In summary, these results underscore the importance of our method in the context of inferring MSCs, particularly when integrated with node classification. As a result, in the subsequent experiments, we will concentrate on analyzing the variations in node classification.

Table 2: Comparison of Different Methods for Link Prediction

Datasets	Machining		Copper		Heat Treatment	
	AUC-ROC(%)	AUC-PR(%)	AUC-ROC(%)	AUC-PR(%)	AUC-ROC(%)	AUC-PR(%)
GraphSAGE	19.77	30.64	37.05	31.11	17.30	27.86
GCN	27.82	42.06	39.98	40.84	44.08	42.31
EdgeConv	34.67	33.01	39.78	31.94	20.06	28.20
SGC	22.83	31.10	35.18	31.25	34.14	30.69
APPNP	17.44	39.26	11.83	32.96	10.06	32.26
FA-GraphSAGE	30.40	43.00	44.34	43.12	40.95	42.12
FA-GCN	37.95	45.45	49.13	45.20	54.10	47.02

Table 3: Comparison of Different Methods for Node Classification

Datasets	Machining		Copper		Heat Treatment	
	AUC-ROC(%)	AUC-PR(%)	AUC-ROC(%)	AUC-PR(%)	AUC-ROC(%)	AUC-PR(%)
GraphSAGE	61.10	55.40	51.60	16.13	52.97	22.92
SENG-GraphSAGE	53.90	50.10	54.74	17.23	56.42	38.72
FA-GraphSAGE	78.70	71.80	73.56	43.48	78.47	52.51
SF-GraphSAGE	82.80	80.90	76.42	41.92	79.78	53.25

4.3. Influence of Classifier

To answer Question (2), we undertake an analysis of the performance variations of the models replacing GraphSAGE with GAT and GCN. All experiments have the same configuration settings as GraphSAGE, are implemented on the same GAT and GCN. “Machining” is used as the target manufacturing service, with its original class distributions and OS set to 1. The results presented in Table 4 reveal that both FA-FCN and SF-GCN, have good performance in evaluation metrics for identifying the capability of “Machining”. The difference in their mechanisms for aggregating information from neighbors in the graph could impact their performance in identifying MSC. FA-GCN also outperforms FA- GraphSAGE. The difference in

their mechanisms for aggregating information from neighbors in the graph could impact their performance in identifying manufacturing service capability. GAT has a lower performance in conjunction with SENG and FA compared to GCN and GraphSAGE. The potential reason is that it uses an attention mechanism to weigh and aggregate information from neighboring nodes in the graph. The attention mechanism’s performance depends on the specific dataset and whether the relationships between nodes benefit from such fine-grained weighting. It might not perform as well as the simpler neighborhood aggregation used in GCN.

Table 4: Comparison of Different Classifiers for Node Classification

Methods	AUC-ROC (%)	AUC-PR (%)
GAT	56.76	55.96
SENG-GAT	56.49	54.98
FA-GAT	73.81	73.98
SF-GAT	73.11	72.96
GCN	72.48	63.99
SENG-GCN	61.77	65.19
FA-GCN	84.59	75.07
SF-GCN	85.79	85.96

4.4. Influence of FA

To answer Question (3), firstly, we compare the performance of utilizing FA and traditional node feature engineering, which is to convert the names of nodes to node features. All the experiments are conducted with Doc2Vec and t-SNE except the ways of generating node features are different. GraphSAGE and GAT are selected as the classifiers. When applying node feature engineering on detecting the capability of “Machining” with GraphSAGE, AUC-ROC and AUC-PR are 57.48% and 62.12%, respectively. Applying the same method with GAT, AUC-ROC and AUC-PR are 56.76% and 55.96%. It is noticed that using the FA method significantly outperforms the traditional method in terms of both AUC-ROC and AUC-PR. FA aggregates neighbor service nodes’ names to manufacturer nodes, which takes into account the broader context of each manufacturer node, while the traditional approach considers each node’s name as its sole feature, more isolated in its context. The results highlight the importance of considering the broader context in graph-based node classification tasks.

In addition, we undertake an analysis of the performance variations of the algorithms replacing Doc2Vec with Bert. The evaluation metrics obtained from the experiments are reported in Table 5. All experiments have the same configuration settings except Doc2Vec is replaced by Bert in FA. “Heat Treatment” and “Copper” are selected as target manufacturing services, with the datasets’ original imbalance ratios and the OS set to 1. The results presented in Table 4 reveal that the implementation of Doc2Vec in FA exhibits a more substantial enhancement relative to the baseline in evaluation metrics when compared to the utilization of Bert. The utilization of Bert demonstrates a relatively inferior performance on the AUC-PR metric compared to the application of Doc2Vec. Given that the textual data utilized in FA does not consist of complete paragraphs with context-dependent sentences, the requirement for contextual understanding of words within a sentence is diminished, resulting in Bert being less appropriate for the task in comparison to Doc2Vec.

Table 5: Comparison of Different Solutions to Node Classification with Bert

Datasets	Copper		Heat Treatment	
	AUC-ROC (%)	AUC-PR (%)	AUC-ROC (%)	AUC-PR (%)
FA-GraphSAGE	63.99	22.61	54.22	23.57
SF-GraphSAGE	77.93	34.61	77.65	47.01

4.5. Influence of Oversampling Ratio

In this section, we undertake an analysis of the performance variations of two algorithms which include an oversampling process with respect to varying levels of oversampling, to address Question (4). The oversampling scale is manipulated to take on the values of $\{0.2, 0.4, 0.6, 0.8, 1.0, 1.2\}$. For all the experiments in this section, the dataset of “Machining” is used with its original imbalance ratio of 0.5866. In order to ensure statistical validity, each experiment was repeated more than 3 times, with the average results presented in Figure 3. For SENG-GraphSAGE, as the OS is smaller than 1, the evaluation metrics decrease slightly first, then increase and achieve optimal scores at 1. It indicates the improvement in the model’s ability to distinguish between Class 1 and Class 0 when synthesizing more samples for minority classes with an OS between 0.6 and 1, as well as shows the fluctuations in the model due to graph oversampling without applying content

embedding. When the OS exceeds 1, a degradation in the performance of SENG-GraphSAGE is observed. This phenomenon can be attributed to an excessive generation of synthetic nodes with redundant or similar information, which ultimately hinders the ability of GraphSAGE to learn effectively. For SF-GraphSAGE, as the oversampling ratio increases from 0.2 to 0.8, the evaluation metrics increase, indicating an improvement in the model’s ability to distinguish between Class 1 and Class 0. However, as the oversampling ratio increases from 1 to 1.2, the evaluation metrics decrease since too many synthetic nodes are generated with similar attributes, which ultimately impairs the ability of GraphSAGE to learn effectively.

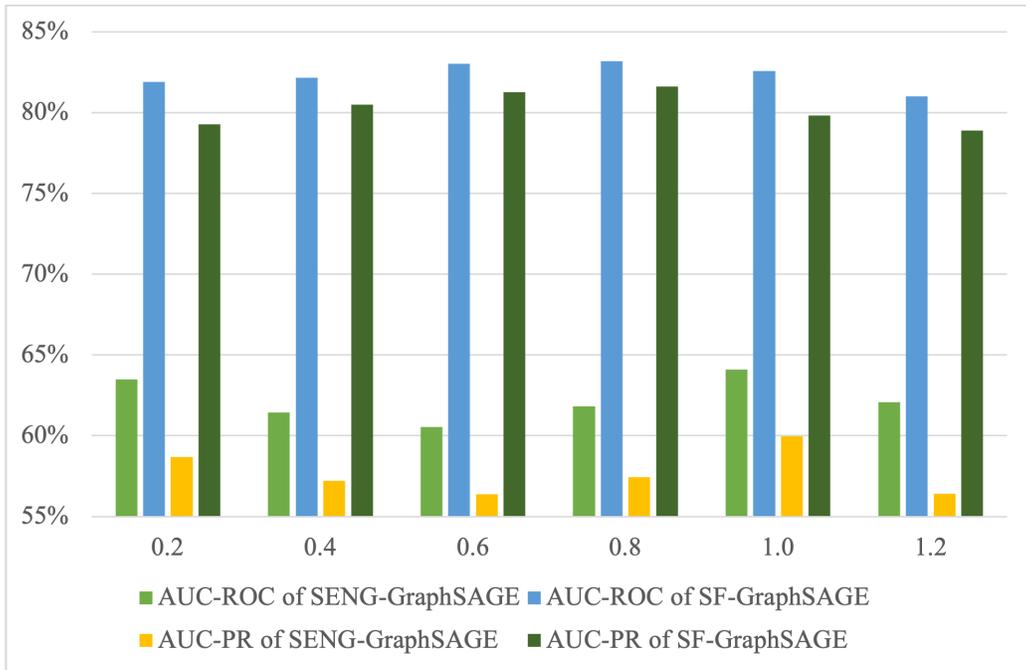


Figure 3: Evaluation Metrics under Different OSs

4.6. Influence of Imbalance Ratio

In this section, we undertake an analysis of the performance variations of various algorithms with respect to varying levels of imbalance ratio, to address Question (5). For all the experiments in the section, the “Machining” dataset is used as well as a fixed OS of 1 is applied to SENG-GraphSAGE and SF-GraphSAGE. The imbalance ratio scale is manipulated to take on

the values of $\{0.1, 0.2, 0.4, 0.5866\}$, where 0.5866 is the original imbalance ratio of the "Machining" dataset. In order to ensure statistical validity, each experiment was repeated more than 3 times, with the average results presented in Figure 4.

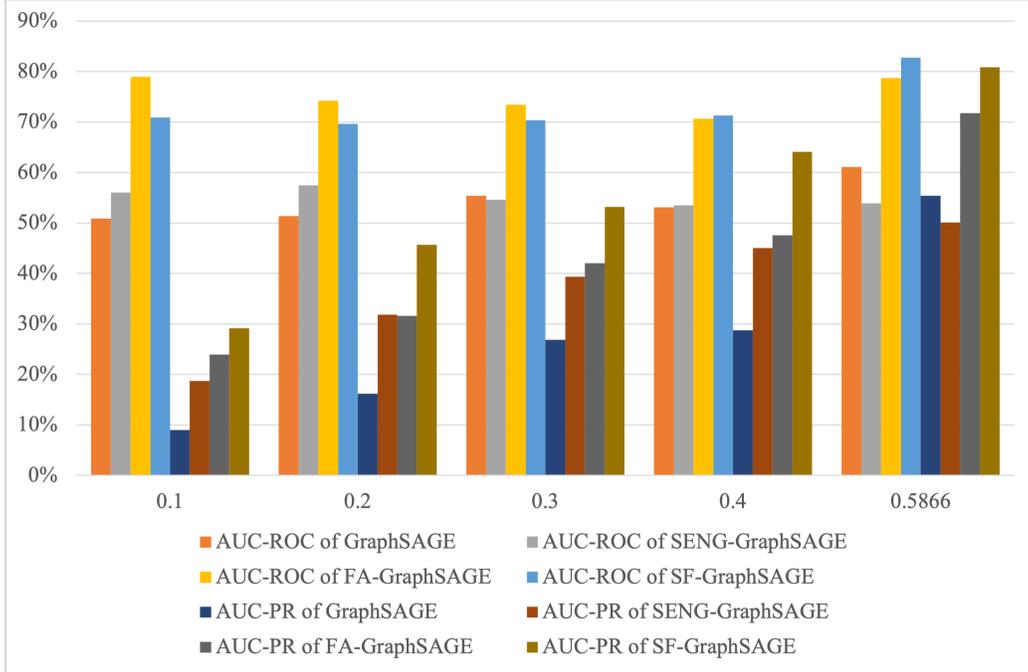


Figure 4: Evaluation Metrics under Different Imbalance Ratios

It is noticed that the SF-GraphSAGE method demonstrates a pervasive performance across various imbalance ratios. It not only excels at specific ratios but maintains a competitive edge throughout the range of presented ratios. SF-GraphSAGE exhibits a relatively stable and consistent behavior, especially highlighted in the AUC-PR progression. For FA-GraphSAGE, in terms of AUC-ROC and AUC-PR, FA-GraphSAGE showcases strong performance, especially at the extremes of the given imbalance ratio (0.1 and 0.2). There's a noticeable dip in performance as we move from an imbalance ratio of 0.1 to 0.4, but the method shows adaptability to varying degrees of imbalance, evidenced by its consistently high AUC-ROC and increasing AUC-PR values.

5. Conclusion And Future Work

In the realm of industrial engineering and logistics, understanding a manufacturer’s service capabilities is crucial for optimizing production efficiency and supply chain management. The current prevailing methods for identifying MSCs from manufacturers are predominantly based on keyword matching and semantic matching. However, these methods tend to either lose hidden information or misunderstand the information, which subsequently leads to incomplete identification of manufacturers’ capabilities. To mitigate the limitation, this study presents a novel GNN-based approach for effectively identifying MSCs within KGs. To enhance the accuracy and performance of this identification process, an innovative strategy is introduced, which involves aggregating information from neighboring nodes and oversampling the graph data. Our rigorous evaluations, conducted on MSKGs, along with subsequent ablation studies, provide unequivocal evidence of the effectiveness and robustness of our proposed approach. These advancements are applicable to a wide range of recommender systems.

Although the effectiveness of our method is demonstrated, some limitations and implications need further attention. In future work, it would be valuable to utilize manufacturing ontologies [62] for constructing a more comprehensive MSKG that includes other critical entities like accuracy requirements and material specialization. This enhanced approach in evaluating MSC will ensure that manufacturers are selected not just for their ability to provide manufacturing services, but also for their alignment with the specific and varied needs of different projects. Besides, the study mainly considers a single type of heterogeneous graph. It is imperative to broaden the scope of our method to encompass other heterogeneous graphs or bipartite graphs. Integrating our method with heterogeneous GNN models [63] or bipartite GNN models [64] can be developed. This work simplifies MSKGs and it does not take into account the diversity and directionality of relationships within the graph. The consideration of the difference between edges and the directionality of edges may lead to a more optimized representation of nodes and edges [65], which can benefit graph-based downstream tasks. Not only node classification but also other graph-based downstream tasks, such as link prediction and graph classification, are suffering from imbalance class issues. Our methods can be tailored to address other imbalanced class problems, enhancing its efficacy in accurately discerning MSCs.

References

- [1] M. Cai and J. Luo. Influence of COVID-19 on manufacturing industry and corresponding countermeasures from supply chain perspective. *Journal of Shanghai Jiaotong University (Science)*, 25:409–416, 2020. doi: 10.1007/s12204-020-2206-z.
- [2] O.Felix Offodile and Layek L Abdel-Malek. The virtual manufacturing paradigm: The impact of it/is outsourcing on manufacturing strategy. *International Journal of Production Economics*, 75(1):147–159, 2002. ISSN 0925-5273. doi: [https://doi.org/10.1016/S0925-5273\(01\)00188-8](https://doi.org/10.1016/S0925-5273(01)00188-8). URL <https://www.sciencedirect.com/science/article/pii/S0925527301001888>. Information Technology/Information Systems in 21st Century Production.
- [3] Dmitry Ivanov and Alexandre Dolgui. The shortage economy and its implications for supply chain and operations management. *International Journal of Production Research*, 60(24):7141–7154, 2022. doi: 10.1080/00207543.2022.2118889. URL <https://doi.org/10.1080/00207543.2022.2118889>.
- [4] Nancy M. Levenburg. Does size matter? small firms’ use of e-business tools in the supply chain. *Electronic Markets*, 15(2):94–105, 2005. doi: 10.1080/10196780500083746. URL <https://www.tandfonline.com/doi/abs/10.1080/10196780500083746>.
- [5] Wei Cao, Pingyu Jiang, and Kaiyong Jiang. Demand-based manufacturing service capability estimation of a manufacturing system in a social manufacturing environment. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231(7): 1275–1297, 2017.
- [6] ThomasNet. Thomasnet - product sourcing and supplier discovery platform, 2024. URL <https://www.thomasnet.com>. [Accessed: January 9, 2024].
- [7] Ying Cheng, Fei Tao, Dongming Zhao, and Lin Zhang. Modeling of manufacturing service supply–demand matching hypernetwork in service-oriented manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, 45:59–72, 2017.

- [8] Amine Belhadi, Karim Zkik, Anass Cherrafi, Sha’ri M. Yusof, and Said El fezazi. Understanding big data analytics for manufacturing processes: Insights from literature review and multiple case studies. *Computers & Industrial Engineering*, 137:106099, 2019. ISSN 0360-8352. doi: 10.1016/j.cie.2019.106099.
- [9] Aman Kumar and Binil Starly. “fabner”: information extraction from manufacturing process science domain literature using named entity recognition. *Journal of Intelligent Manufacturing*, 33(8):2393–2407, 2022.
- [10] Roberto Sala, Fabiana Pirola, Giuditta Pezzotta, and Sergio Cavalieri. Nlp-based insights discovery for industrial asset and service improvement: an analysis of maintenance reports. *IFAC-PapersOnLine*, 55(2):522–527, 2022. ISSN 2405-8963. doi: <https://doi.org/10.1016/j.ifacol.2022.04.247>. URL <https://www.sciencedirect.com/science/article/pii/S2405896322002488>. 14th IFAC Workshop on Intelligent Manufacturing Systems IMS 2022.
- [11] Junhyung Moon, Gyuyoung Park, Minyeol Yang, and Jongpil Jeong. Design and verification of process discovery based on nlp approach and visualization for manufacturing industry. *Sustainability*, 14(3), 2022. ISSN 2071-1050. doi: 10.3390/su14031103. URL <https://www.mdpi.com/2071-1050/14/3/1103>.
- [12] Imhade P Okokpujie, CA Bolu, OS Ohunakin, Esther T Akinlabi, and DS Adelekan. A review of recent application of machining techniques, based on the phenomena of cnc machining operations. *Procedia Manufacturing*, 35:1054–1060, 2019.
- [13] Manfred Peters, Jörg Kumpfert, Charles H Ward, and Christoph Leyens. Titanium alloys for aerospace applications. *Advanced engineering materials*, 5(6):419–427, 2003.
- [14] Madalina Simona Baltatu, Petrica Vizureanu, Andrei Victor Sandu, Nestor Florido-Suarez, Mircea Vicentiu Saceleanu, and Julia Claudia Mirza-Rosca. New titanium alloys, promising materials for medical devices. *Materials*, 14(20):5934, 2021.

- [15] Jiacheng Xu, Kan Chen, Xipeng Qiu, and Xuanjing Huang. Knowledge graph representation with jointly structural and textual encoding. *arXiv preprint arXiv:1611.08661*, 2016.
- [16] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. *arXiv preprint arXiv:2104.06378*, 2021.
- [17] Edward Elson Kosasih and Alexandra Brintrup. A machine learning approach for predicting hidden links in supply chain with graph neural networks. *International Journal of Production Research*, 60(17):5380–5393, 2022. doi: 10.1080/00207543.2021.1956697. URL <https://doi.org/10.1080/00207543.2021.1956697>.
- [18] *Design of Knowledge Graph in Manufacturing Services Discovery*, volume Volume 2: Manufacturing Processes; Manufacturing Systems; Nano/Micro/Meso Manufacturing; Quality and Reliability of *International Manufacturing Science and Engineering Conference*, 06 2021. doi: 10.1115/MSEC2021-63766. URL <https://doi.org/10.1115/MSEC2021-63766>.
- [19] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003.
- [20] Hamed Bouzary and F. Frank Chen. A classification-based approach for integrated service matching and composition in cloud manufacturing. *Robotics and Computer-Integrated Manufacturing*, 66:101989, 2020. ISSN 0736-5845. doi: <https://doi.org/10.1016/j.rcim.2020.101989>. URL <https://www.sciencedirect.com/science/article/pii/S0736584520302003>.
- [21] Hui Xiong, Yi Cheng, Wenhao Zhao, and Jianhua Liu. Analyzing scientific research topics in manufacturing field using a topic model. *Computers & Industrial Engineering*, 135:333–347, 2019. ISSN 0360-8352. doi: <https://doi.org/10.1016/j.cie.2019.06.010>. URL <https://www.sciencedirect.com/science/article/pii/S0360835219303377>.
- [22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings*

of Workshop at International Conference on Learning Representations (ICLR), 2013.

- [23] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*, 2016.
- [24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Bidirectional encoder representations from transformers. *arXiv preprint arXiv:1810.04805*, 2019.
- [25] Niklas Muennighoff. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*, 2022.
- [26] Xinyi Huang, Lianglun Cheng, Jianfeng Deng, and Tao Wang. Binocular attention-based stacked bilstm ner model for supply chain management event knowledge graph construction. In *Proceedings of the 2023 15th International Conference on Machine Learning and Computing, ICMLC '23*, page 40–46, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450398411. doi: 10.1145/3587716.3587723. URL <https://doi.org/10.1145/3587716.3587723>.
- [27] Ehsan Pourjavad and Arash Shahin. The application of mamdani fuzzy inference system in evaluating green supply chain management performance. *International Journal of Fuzzy Systems*, 20:901–912, 2018.
- [28] Diego A Carrera and Rene V Mayorga. Supply chain management: a modular fuzzy inference system approach in supplier selection for new product development. *Journal of Intelligent Manufacturing*, 19:1–12, 2008.
- [29] Youssef Tliche, Atour Taghipour, and Béatrice Canel-Depitre. An improved forecasting approach to reduce inventory levels in decentralized supply chains. *European Journal of Operational Research*, 287(2):511–527, 2020.
- [30] Sofia Berto Villas-Boas. Vertical relationships between manufacturers and retailers: Inference with limited data. *The Review of Economic Studies*, 74(2):625–652, 2007.

- [31] Dimitris Mourtzis, Michael Doukas, and Dimitra Bernidaki. Simulation in manufacturing: Review and challenges. *Procedia Cirp*, 25:213–229, 2014.
- [32] Boonserm Kulvatunyou, Yunsu Lee, Nenad Ivezic, and Yun Peng. A framework to canonicalize manufacturing service capability models. *Computers & Industrial Engineering*, 83:39–60, 2015.
- [33] Yong Zhang, Ming Sheng, Rui Zhou, Ye Wang, Guangjie Han, Han Zhang, Chunxiao Xing, and Jing Dong. Hkgb: an inclusive, extensible, intelligent, semi-auto-constructed knowledge graph framework for healthcare with clinicians’ expertise incorporated. *Information Processing & Management*, 57(6):102324, 2020.
- [34] Yueqin Zhu, Wenwen Zhou, Yang Xu, Ji Liu, Yongjie Tan, et al. Intelligent learning for knowledge graph towards geological data. *Scientific Programming*, 2017, 2017.
- [35] Yunqing Li and Binil Starly. Building a knowledge graph to enrich chatgpt responses in manufacturing service discovery. 2021. doi: 10.2139/ssrn.4517533. URL <https://ssrn.com/abstract=4517533>.
- [36] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212, 2017. URL <http://arxiv.org/abs/1704.01212>.
- [37] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *CoRR*, abs/1706.02216, 2017. URL <http://arxiv.org/abs/1706.02216>.
- [38] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):1–23, 2019.
- [39] Felix Wu, Tianyi Zhang, Amauri H. Souza Jr., Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying graph convolutional networks. *CoRR*, abs/1902.07153, 2019. URL <http://arxiv.org/abs/1902.07153>.

- [40] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds. *CoRR*, abs/1801.07829, 2018. URL <http://arxiv.org/abs/1801.07829>.
- [41] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Personalized embedding propagation: Combining neural networks on graphs with personalized pagerank. *CoRR*, abs/1810.05997, 2018. URL <http://arxiv.org/abs/1810.05997>.
- [42] Luana Ruiz, Fernando Gama, and Alejandro Ribeiro. Gated graph recurrent neural networks. *IEEE Transactions on Signal Processing*, 68:6303–6318, 2020. doi: 10.1109/tsp.2020.3033962. URL <https://doi.org/10.1109%2Ftsp.2020.3033962>.
- [43] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Graphgan: Graph representation learning with generative adversarial nets. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [44] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.
- [45] Ce Zheng, Matías Mendieta, Pu Wang, Aidong Lu, and Chen Chen. A lightweight graph transformer network for human mesh reconstruction from 2d human pose. *CoRR*, abs/2111.12696, 2021. URL <https://arxiv.org/abs/2111.12696>.
- [46] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.
- [47] Shunxin Xiao, Shiping Wang, Yuanfei Dai, and Wenzhong Guo. Graph neural networks in node classification: Survey and evaluation. *Machine Vision and Applications*, 33, February 2022. doi: 10.1007/s00138-022-01315-y. URL <https://link.springer.com/article/10.1007/s00138-022-01315-y>.
- [48] Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas. Link prediction techniques, applications, and performance:

A survey. *Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, 221-005, India*, Version of Record 4 June 2020, February 2020. doi: 10.1016/j.jocs.2020.05.030. URL <https://www.sciencedirect.com/science/article/pii/S1877056820302349>.

- [49] Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A fair comparison of graph neural networks for graph classification. *arXiv*, 2020. doi: 10.48550/arXiv.1912.09893. URL <https://arxiv.org/abs/1912.09893>. Extended version of the paper published at the International Conference on Learning Representations (ICLR), 2020. Additional results are shown in the appendix.
- [50] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Pick and choose: a gnn-based imbalanced learning approach for fraud detection. In *Proceedings of the web conference 2021*, pages 3168–3177, 2021.
- [51] Kushankur Ghosh, Arghasree Banerjee, Sankhadeep Chatterjee, and Soumya Sen. Imbalanced twitter sentiment analysis using minority over-sampling. In *2019 IEEE 10th international conference on awareness science and technology (iCAST)*, pages 1–5. IEEE, 2019.
- [52] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- [53] Tianxiang Zhao, Xiang Zhang, and Suhang Wang. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining, WSDM '21*, page 833–841, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450382977. doi: 10.1145/3437963.3441720. URL <https://doi.org/10.1145/3437963.3441720>.
- [54] Yongxu Liu, Zhi Zhang, Yan Liu, and Yao Zhu. Gatsmote: Improving imbalanced node classification on graphs via attention and homophily. *Mathematics*, 10(11), 2022. ISSN 2227-7390. doi: 10.3390/math10111799. URL <https://www.mdpi.com/2227-7390/10/11/1799>.

- [55] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [56] Felipe L. Gewers, Gustavo R. Ferreira, Henrique Ferraz de Arruda, Filipi Nascimento Silva, Cesar H. Comin, Diego R. Amancio, and Luciano da F. Costa. Principal component analysis: A natural approach to data exploration. *CoRR*, abs/1804.02502, 2018. URL <http://arxiv.org/abs/1804.02502>.
- [57] Durgesh Tamhane, Jinit Patil, Sauvik Banerjee, and Siddharth Tallur. Feature engineering of time-domain signals based on principal component analysis for rebar corrosion assessment using pulse eddy current. *IEEE Sensors Journal*, 21(19):22086–22093, 2021.
- [58] Christoph Bienefeld, Florian Michael Becker-Dombrowsky, Etnik Shatri, and Eckhard Kirchner. Investigation of feature engineering methods for domain-knowledge-assisted bearing fault diagnosis. *Entropy*, 25(9), 2023. ISSN 1099-4300. doi: 10.3390/e25091278. URL <https://www.mdpi.com/1099-4300/25/9/1278>.
- [59] Bin Wang, Angela Wang, Fenxiao Chen, Yuncheng Wang, and C-C Jay Kuo. Evaluating word embedding models: Methods and experimental results. *APSIPA transactions on signal and information processing*, 8: e19, 2019.
- [60] Yunqing Li. Entities and Relationships of the Manufacturing Service Knowledge Graph. 10 2023. doi: 10.6084/m9.figshare.24416059.v1. URL https://figshare.com/articles/dataset/Entities_and_Relationships_of_the_Manufacturing_Service_Knowledge_Graph/24416059.
- [61] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [62] Severin Lemaignan, Ali Siadat, J-Y Dantan, and Anatoli Semenenko. Mason: A proposal for an ontology of manufacturing domain. In *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06)*, pages 195–200. IEEE, 2006.
- [63] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. Heterogeneous graph neural network. In *Proceedings*

of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, page 793–803, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330961. URL <https://doi.org/10.1145/3292500.3330961>.

- [64] Zhao Li, Xin Shen, Yuhang Jiao, Xuming Pan, Pengcheng Zou, Xi-anling Meng, Chengwei Yao, and Jiajun Bu. Hierarchical bipartite graph neural networks: Towards large-scale e-commerce applications. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1677–1688. IEEE, 2020.
- [65] Guillaume Jaume, An-phi Nguyen, María Rodríguez Martínez, Jean-Philippe Thiran, and Maria Gabrani. edggn: a simple and powerful GNN for directed labeled graphs. *CoRR*, abs/1904.08745, 2019. URL <http://arxiv.org/abs/1904.08745>.