THESIS TITLE

PHAPHONTEE YAMCHOTE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY (COMPUTER SCIENCE)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2023

Thesis
entitled
THESIS TITLE

.....
Mr. Phaphontee Yamchote
Candidate

.....
Dr. Thanapon Noraset,
Ph.D. (Computer Science)
Major advisor

.....
Dr. Chainarong Amornbunchornvej,
Ph.D. (Computer Science)
Co-advisor

.....
Prof. .....
Dean
Faculty of Graduate Studies
Mahidol University

.....
Asst. Prof. Boonsit Yimwadsana,
Ph.D. (Electrical Engineering)
Program Director
Doctor of Philosophy Programme
in Computer Science
Faculty of Information and
Communication Technology
Mahidol University

Thesis
entitled
THESIS TITLE

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Doctor of Philosophy (Computer Science)
on
January 5, 2021

.....
Mr. Phaphontee Yamchote
Candidate

.....
.....,
Ph.D. (Computer Science)
Chair

.....
Dr. Thanapon Noraset,
Ph.D. (Computer Science)
Member

.....
.....,
Ph.D. (Computer Science)
Member

.....
Dr. Chainarong Amornbunchornvej,
Ph.D. (Computer Science)
Member

.....
Prof. .....
Dean
Faculty of Graduate Studies
Mahidol University

.....
.....,
Ph.D. (Computer Science)
Dean
Faculty of Information and
Communication Technology
Mahidol University

ACKNOWLEDGEMENTS

First

Phaphontee Yamchote

THESIS TITLE

PHAPHONTEE YAMCHOTE 6436198 ITCS/D

Ph.D. (COMPUTER SCIENCE)

THESIS ADVISORY COMMITTEE: THANAPON NORASET, Ph.D.,
CHAINARONG AMORNBUNCHORNVEJ, Ph.D.

ABSTRACT

Your abstract goes here.

IMPLICATION OF THE THESIS

Your thesis implication goes here.

KEY WORDS : MACHINE LEARNING / GRAPH NEURAL NETWORK

19 pages

CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

CHAPTER I
INTRODUCTION

[about data]

Nowadays, data plays an important role in both academia and business. They can find some business opportunities from their data. Then, the field of study that unifies engineering, mathematics, statistics, and computer science emerges, called data science. Data science is broadly used in business and industry to find the business opportunity hidden in data. Chatterjee et al. (2021) [**?**] concluded that using data science in an enterprise increases accuracy in innovative decisions and the potentiality of competition in business.

[tell that missing data is a problem often met]

However, not only mining or modeling data but also data preparation is a task that machine learning practitioners have to do. They spend much time in cleaning and organizing data. One of the common challenges is missing features which is the problem of absence for some values in the given datasets. This problem is relatively common in almost all research. It can significantly affect results obtained from the data [**?**] in sense of learning.

[causes of missing data]

There are many causes of missing data. First, the problem of measuring instruments is a cause that often occurs in a production line. Second, some missing data can emerge from the nature of data, such as different factors corrected between males and females in a clinical survey or the disappearance of a patient during the treatment process. Third, the lack of knowledge of the sample can also result in missing data. There may be the case of the sample giving a “don’t know” answer to an observer in some complicated question.

[**significance of missing data**]

Why is the missingness problem noteworthy, especially in machine learning and deep learning? The most trivial answer to this question is about computation of almost machine learning algorithms. Including deep learning emerging the state-of-the-art in the current era, they calculate the fixed size of the representation vector, sometimes called embedding. However, without preprocessing, missing data leads to the problem that we cannot feed the data directly into a model because of the absence of some values inside them.

[**Imputation: the most basic solution**]

One of the simple way is to basically impute these missing values by a constant value, mean or interpolation. However, these imputation may not reflect the actual dataset. Actually, we do not even know the actual dataset. In addition, some missing behavior is also informative such as unrating the product from a customer which may mean dissatisfaction but the customer avoids giving a reason directly. So imputation seems that we do not concern the hidden information about missing data. Moreover, Ma, et al. (2018) [**?**] commented that the imputation method ignores uncertainties of data which means as the same as hidden information said earlier.

[**More a bit Imputation but with learning from data**]

Not only basic imputation methods, but many work also investigated imputation based methods by learning from data. However, it still relies on the assumption that all missing values can be replaced as a number. But not all missing scenario can be imputed, for example, unrating the product. Also the case of some non measurable feature depending on others, the value 0 and a missing value are not equivalent. Moreover, these methods is independent from the learning process. In other words, the same model may be lead to different way by different imputation methods. The only way is to cross test all possible imputation methods to see whether which method is the best one for the desired model. This still requires much effort and time-consumption in the process of feature engineering.

[Tabular data: a format of data often used and also meet missing]
Tabular data is a format of data that displays in rows and columns. One of them corresponds to the collected features of each data sample. It is a ubiquitous data format used in practical applications in many domains such as enterprise operations, manufacturing, clinics, and surveys. For example, doctors or medical technologists may want to predict the chance of a specific disease for a patient from the patient's historical records that are stored in the form of a table. Missing data problem is also a challenge in modeling tabular data. Even though tabular data has its own structure represented by feature vector, missing values can eliminates the structure of tabular data so that model cannot learn or compute on such data.

[It's time to tell about graphs and GNNs: graph first]
So we need a learning algorithm or a data structure that is more flexible in computation rather than using the fixed length of feature vector fed into algorithm directly. This work concentrates on graphs data structures and graph neural network (GNN) algorithms. As we have ever seen in data structure, graphs are flexible in how to storage data via edges and nodes. Moreover, they are suitable to be flexibly computed by using edges.

[It's time to tell about graphs and GNNs: GNNs next]
GNNs are a class of neural based methods designed to perform inference on graph data. It has emerged as a promising approach for modeling structured and semi-structured data, such as social networks [**?**], molecular structures [**?**], and knowledge graphs [**?**]. Particularly when dealing with high-dimensional or incomplete datasets, GNNs have demonstrated superior performance in various applications, as they effectively capture complex dependencies among nodes and edges in a graph [**?** , Kipf & Welling 2017]. Moreover, GNNs have advantages over traditional machine learning methods as they do not rely on assumptions about independence of instances and features. This makes GNNs are more flexible in learning with respect to both data they use and computation they do.

[Conclude to our expectation to use graphs and GNNs to address missing data problem]

Therefore, the objective of this work is to investigate the new method to learn representations or embeddings of tabular data containing missing values by using GNNs. We also aim to find a new way to represent tabular data by a graph. The proposed algorithm will allow users to directly feed a data point from tabular data containing missing values without any preprocessing process. Moreover, the model can be trained in an end-to-end fashion.

1.1 Research Objectives

Problem Statement

Missing data, which often occurs in many domains of tabular data, can lead to biased or inaccurate models if not handled properly. Current methods for handling missing data, such as imputation, can introduce additional assumptions and may not capture the underlying structure of the data. Moreover, doing so in the process of feature engineering takes much time and effort of practitioners. We seek to develop an approach that can effectively handle missing data in datasets without relying on process of data preparation for missing values such as imputation.

Objectives

The primary objective of this research is to investigate a GNN-based framework that can address the challenge of missing data. The further detail of our objectives is as follows:

1. develop an algorithm to construct graph structures for the representation of data that can be used for computation of the corresponding GNN prediction algorithm,
2. develop a GNN algorithm that can be used for complete data, and

3. develop a GNN algorithm that can be trained by datasets containing missing values without any preprocessing, and also can predict the corresponding label of the given data that may contain missing values.

1.2 Outlines

In Chapter 2, we explain more detail related to our work including graph representation, GNNs and related work. Readers can find deep detail and also formal definition in the section preliminary of this chapter. And then, we describe how we perform this research in Chapter 3: methodology.

CHAPTER II
LITERATURE REVIEW

2.1 Preliminary

[Wrap-up 3 things and explain more about tabular data]

There are 3 major objects discussed in introduction chapter: (1) tabular data, (2) missing values and (3) GNNs. In brief, tabular data is the format of data that we concentrate on in this work. It is a format of data that displays in rows and columns whose one of them corresponds to the collected features of each data sample. Here, we use rows for samples. The another axis of table refers to the features or attributions of datasets, which are the fields or variables representing a specific aspect or characteristic of each data point within a structured table. Formally, we use the notation $T \in \mathbb{R}^{m \times n}$ for the table of data of m samples and n features. When we refer to the ith instance, which is in the row ith, we use $T_{i:}$.

[Missing data in tabular]

Next, missing data is one of the challenges in learning predictive model for tabular data. It is the scenario that some feature values of a sample (row) is absence by some reason such as quality of measuring equipment or dependency from value of other features. It can lead to biased model performance and inaccurate predictions. It can be a problem not only in tabular data but also in different data types as well such as image or time series. However, this work concentrates only tabular data. Since they are a data structure with high variability of data types and format, they possibly ambiguous and low quality due to missing values []. Not like an image, a small proportion of missing values in images may not significantly impact classification learning.

[GNNs for missing data in tabular data]

And the last main character is graph neural networks, which is a deep neural based framework designed for graph data structures. It can be seen as a generalization of traditional multilayer perceptron which nodes of graph are grouped into each level or layer where each node (neuron) in one layer is connected to every node in the adjacent layers with independent weights for edges [**?**]. GNNs are used to process graph data which may contains high level of prescribed dependency, in other words, not independent and identical distributed which is often used to be an assumption for almost machine learning algorithm. Moreover, values in tabular data, which is heterogeneous, may not independent, and also missing values as well. Transforming them to be graphs and using GNNs might be an interesting methods to be explored.

In this section, we will give you more precise details of missing data and GNNs.

2.1.1 Missing Data

Missing data is the scenario that some feature values of a sample is absence by some reason such as quality of measuring equipment or dependency from value of other features. For example in clinical scenario, some features values of a patient may not be determined, even cannot, due to the gender of the observed patient. It can be classified into three categories [**?**]: completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR).

2.1.1.1 Missing Values Mechanisms

According to [**?**], Missingness can be categorized by dependency of missing features into 3 mechanisms: MCAR, MAR and MNAR.

MCAR means that the missingness is random and has no relationship with the values of other variables in the dataset. For example, if we collect height data for a random sample of individuals, and some of them are absent from the data due to equipment malfunction, then the missingness is MCAR. In this case, the missing data can be safely ignored in statistical analysis without introducing bias.

MAR means that the missingness is not random but can be explained by other variables in the dataset. For example, in a study of school performance, some students may not complete a survey on mental health due to embarrassment or discomfort, and their absence can be explained by their mental health status. In this case, the missing data are MAR, and ignoring it may lead to biased results. Therefore, imputation methods can be used to estimate the missing values based on the observed values and other relevant variables.

MNAR means that the missingness is not random and cannot be explained by other variables in the dataset. For example, in a study of employee salaries, some high-income earners may refuse to report their salaries, and their absence is related to the variable of interest but not to other variables in the dataset. In this case, ignoring the missing data may lead to biased results, and imputation methods may not be effective since the missing values are not predictable.

Here is a formal definition of missingness mechanism from [**?**].

Definition 2.1. For precise describing, define the complete data matrix $X = (x_{ij})$ and the missingness indicator matrix $M = (m_{ij})$ where $m_{ij} = 1$ if x_{ij} is missing and $m_{ij} = 0$ if x_{ij} is observed. Assume that the rows (x_i) and (m_i) are independent and identically distributed over i. The missingness mechanism is characterized by the conditional distribution of m_i given x_i, say $f_{M|X}(m_i|x_i, \phi)$, where ϕ denotes unknown parameters. Let $x_{(1)i}$ denote the components of x_i that are missing for unit i.

1. **Missing Completely at Random** (MCAR) is the missing mechanism that missingness does not depend on the values of the data, missing or observed. That is, if for all i and any distinct values x_i, x_i^* in the sample space of X,

$$f_{M|X}(m_i|x_i, \phi) = f_{M|X}(m_i|x_i^*, \phi).$$

2. **Missing at Random** (MAR) is the missingness mechanism that missingness depends on x_i only through the observed components $x_{(0)i}$. Specifically, if for all i and any distinct values $(x_{(1)i}, x_{(1)i}^*)$ of

the missing components in the sample space of $x_{(1)i}$,

$$f_{M|X}(m_i|x_{(0)i},x_{(1)i},\phi) = f_{M|X}(m_i|x_{(0)i},x^*_{(1)i},\phi).$$

3. **Missing not at Random** (MNAR) is the missing mechanism that the distribution of m_i depends on the missing components of x_i.

2.1.1.2 Traditional Methods to Handling Missing Values

Content...

2.1.2 Graph Neural Network

Graph Neural Networks (GNNs) have emerged as a promising approach for modeling structured and semi-structured data, such as social networks [**?**], molecular structures [**?**], and knowledge graphs [**?**]. Traditional machine learning models often struggle to capture the complex dependencies in tabular data, particularly when dealing with high-dimensional or incomplete datasets. GNNs, however, have demonstrated superior performance in various applications, as they effectively capture complex dependencies among nodes and edges in a graph [**?** , Kipf & Welling 2017].

2.1.2.1 Deep Learning towards GNNs

Before going deep further to detail of GNNs, let us introduce the broad definition of deep learning which is the fundamental idea of GNNs. After explaining these, we then move to the often-used layers for GNNs called message passing and graph pooling.

content...

2.1.2.2 Message Passing

content...

2.1.2.3 Graph Pooling

content...

2.2 Related Work

2.2.1 Handling Missing Data

Many works investigated methods to deal with this problem. In addition to statistical imputation, some work developed new methods in machine learning approaches to impute missing data. For example, Samad and Harp dealt with missing values by self-organizing map (SOM) [**?**]. Many works use multilayer perceptron (MLP), for example, [**?** **?** **?**]. Moreover, there are also works using deep learning approaches, for example, [**?** **?**]. However, there also are works dealing with this problem by avoiding imputation. They try to learn a representation of data instead of imputation. These works are based on the hypothesis that missingness should be informative, and imputation is the method that ignores uncertainties in missing data.

content...

2.2.2 Graph Neural Networks for Missing Data

content...

2.2.3 Graph Neural Networks for Tabular Data

GNNs have been applied to diverse types of tabular data, including healthcare [**?**] and financial data [**?**]. Their ability to model complex interactions and dependencies between variables [**?** **?**] and handle missing values [**?**] make them advantageous for tabular data analysis. However, designing GNNs for tabular data requires careful consideration of factors such as graph structure and feature engineering [**?**]. The graph structure should reflect the relationships between variables, while feature engineering should effectively transform input features into graph structures suitable for GNN models.

content...

**********log**********

Missing data is significant because it can lead to biased model performance and inaccurate predictions. Handling missing data can be done through various methods such as statistical imputation, machine learning-based imputation, and deep learning-based imputation. Statistical methods such as mean imputation, regression imputation, and hot-deck imputation assume that the missing values follow a certain statistical distribution, but they might not be accurate if the missing values are not random. Machine learning methods like k-nearest neighbor (KNN) imputation [**?**] and decision tree-based imputation [**?**] can be effective when the missing values have some patterns, but they might not work well if the missingness is too extensive. Deep learning methods such as autoencoders [**?**] and Generative Adversarial Networks (GANs) [**?**] can be used to impute missing values, but they require a large amount of training data and computational resources, and their performance might not be better than simpler imputation methods [**?**].

However, A limitation of imputation methods is that they assume that the missing values can be imputed based on the observed values and other relevant variables. However, in practice, we may not know the exact values of the missing data, and imputing them may introduce errors and bias in the analysis. Moreover, in some cases, the missing data itself may be informative, and imputing them may lead to loss of information. Therefore, it is important to carefully consider the nature of missing data and the suitability of different imputation methods before making any assumptions or decisions.

Recent research has shown that deep learning techniques can also be applied to handle missing data. One such approach is GAIN [**?**], proposed by J. Yoon et al. in 2018, which utilizes Generative Adversarial Nets (GANs) to generate plausible values for missing data based on the observed data. Another study by M. Smieja et al. [**?**] in 2019 also explored the use of neural networks for processing missing data, where a deep network was used to predict data containing missing values based on the observed values without imputation. In 2020, Ghorbani et al. [**?**] proposed a novel method called Embedding for Informative Missingness

(EFIM), which uses a deep learning model to learn embeddings of the missing values that preserve their informativeness. The embeddings are then used to predict the corresponding label of input via the model that is learned together with the embedding. These recent studies suggest that deep learning-based methods can be effective for handling missing data, but they still ignore the dependencies of features on missing values.

In this work, we aim to develop a framework for handling missing data that does not rely on explicit imputation. Specifically, we want to design a model that can predict the label of a given data point even if some of its features are missing, without necessarily imputing the missing values. If necessary, the model should also be able to reconstruct the missing parts of the input based on the other nonmissing features. Our proposed approach will leverage deep learning techniques to capture the complex relationships between the input features and the label, as well as the dependencies between the missing and nonmissing features. By doing so, we hope to provide a more accurate and robust solution to the missing data problem, while also preserving the integrity of the original data.

According to Grinsztajn (2022) [**?**], deep learning models have generally not performed as well as traditional machine learning models, particularly tree-based models, in tabular data analysis. As discussed in the previous section, there are several challenges that contribute to this underperformance. Recent advances in deep learning for tabular data have incorporated neural networks with architectures specifically designed for tabular data, TabNet [**?**] for example. Some work attempts to capture feature interactions [**?**] which is simply an operation among two or more features with respect to the output variable such as multiplication between two features, and handle missing data [**?**]. However, these methods can be computationally expensive. Table2Graph, for instance, uses a reinforcement learning approach that requires significant amounts of data and computation power to address feature interactions. GRAPE, on the other hand, models a table of data as a graph to handle missing data but does not address feature interactions.

Numerous studies have compared the performance of deep learning and traditional machine learning methods for tabular data. Some studies found that

deep learning models outperform traditional machine learning models on certain datasets, such as those with high-dimensional and complex features, while others found that traditional machine learning models perform better on datasets with small to medium-sized features. For example, Ching et al. (2018) [**?**] found that deep learning models outperformed traditional machine learning models on a dataset with high-dimensional features. In contrast, other studies, such as Grinsztajn et al. (2022) [**?**], Olson et al. (2018) [**?**], and Fernández et al. (2014) [**?**] found that traditional machine learning models, such as random forests and gradient boosting machines, outperformed deep learning models on certain tabular datasets. Despite these mixed results, it is clear that both deep learning and traditional machine learning methods have their strengths and weaknesses when it comes to tabular data analysis, and the choice of method should depend on the specific characteristics of the dataset and the research question at hand.

Graph representation learning has emerged as a promising approach for handling missing data. Notably, You et al. (2020) [**?**] proposed GRAPE, a framework utilizing GNNs for feature imputation and label prediction with missing data. GRAPE constructs a bipartite graph from the data matrix and formulates feature imputation as an edge-level prediction task and label prediction as a node-level prediction task.

Danel et al. (2020) [**?**] presented an approach for processing incomplete images as graphs, employing a spatial graph convolutional neural network (SGCN) to handle missing data without imputation. However, this method only considers spatial graph convolutions based on Euclidean distance, which may limit its ability to capture complex relationships between pixels. Alternative graph convolutions, such as spectral or attention-based, could offer greater flexibility and expressiveness.

GNN-based approaches for handling missing data have advantages over traditional imputation methods, as they do not rely on assumptions about missing values and can capture complex dependencies between features and missing data. Furthermore, the learned node or graph representations can be used for downstream analysis without imputing missing values, potentially leading to more

accurate and interpretable results. Nevertheless, challenges remain in applying GNNs to handle missing data, including selecting appropriate graph construction methods, and GNN architectures, and evaluating their performance under various missing data scenarios.

CHAPTER III
METHODOLOGY

[In this section, you should remind your readers what the focus of your study is, especially the research aims. As we've discussed many times on the blog, your methodology needs to align with your research aims, objectives and research questions. Therefore, it's useful to frontload this component to remind the reader (and yourself!) what you're trying to achieve.]

some introduction some introduction

3.1 Research Methodology

To achieve our research objectives, we will use the following methodology:

3.1.1 Dataset

We will use benchmark datasets for regression and classification tasks that are commonly used in the literature. Specifically, we will use the datasets mentioned as benchmarks for tabular data in the work of Grinsztajn (2022) to ensure comparability with existing studies. We will also preprocess the data to handle missing values and normalize the features. Moreover, we also use synthetic datasets to be able to control the condition of interactions and missing values.

3.1.2 Graph Construction

Given a table of data $T \in \mathbb{R}^{n \times p}$ containing n instances (rows) and each instance has p attributes (columns). As depicted in Figure 3.1, the transformed graph of instance $i \in \{1, 2, \ldots, n\}$ is the graph $G_i = (V_{G_i}, E_{G_i})$ whose nodes $V_{G_i} = (F_1, \ldots, F_p)$ correspond to feature fields of the table. We call this graph a **feature graph**. The feature interactions are represented by edges.

In this work, defining E_{G_i} explicitly remains an open and under-explored question. As the ground-truth of feature interaction is typically unknown in real-world scenarios, we aim to create a model capable of learning feature interaction from the complete feature graph ($E_{G_i} = V_{G_i} \times V_{G_i}$) to enhance interpretability.

Figure 3.1: an example of construction of graphs from tabular data

3.1.3 Model Design

We will develop a novel GNN-based framework that takes datapoint-wise graphs as input. We will use PyTorch [**?**] and PytorchGeometric [**?**] to implement our proposed framework. The model will include attention mechanisms to identify important features and handle missing data. We will implement our proposed GNN framework and compare its performance with various baselines, including existing deep learning models and traditional machine learning models.

3.1.4 Performance Evaluation

We will use Mean Squared Error (MSE) as the evaluation metric for regression tasks and metrics from confusion matrices (e.g., accuracy, precision, recall, F1-score) as the evaluation metric for classification tasks. We will compare the performance of our proposed framework with existing deep learning-based models and traditional machine learning models.

We will conduct experiments to evaluate the performance of our proposed framework on benchmark datasets. Specifically, we will compare the performance of our proposed framework both without missing data and with missing data with the following models:

- Deep learning-based models: MLP (Multilayer Perceptron), CNN (Convolutional Neural Network), and DNN (Deep Neural Network).
- Traditional machine learning models: Random Forest, Gradient Boosting Machine, and Support Vector Machine. We will use five-fold cross-validation to ensure the reliability of the results.
- Graph-based deep models: GRAPE [**?**], Fi-GNN [**?**], Table2Graph [**?**]

3.2 Research Framework

REFERENCES

[1] Angkana Huang. Unofficial latex template of the mahidol unversity thesis, 2017. URL `https://github.com/hatio/MahidolThesis`.

BIOGRAPHY

NAME	Mr. Phaphontee Yamchote
DATE OF BIRTH	26 August 1992
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Chulalongkorn University, 2012–2016
	Bachelor of Science (Mathematics)
	Chulalongkorn University, 2017–2021
	Master of Science (Mathematics)
	Mahidol University, 2021–20..
	Doctor of Philosophy (Computer Science)
E-MAIL	yamchote_p@outlook.com