ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

A DEEP LEARNING ARCHITECTURE FOR MISSING METABOLITE CONCENTRATION PREDICTION

M.Sc. THESIS

Sadi ÇELİK

Department of Computer Engineering

Computer Engineering Programme

JULY 2024

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

A DEEP LEARNING ARCHITECTURE FOR MISSING METABOLITE CONCENTRATION PREDICTION

M.Sc. THESIS

Sadi ÇELİK
(504211530)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Ali ÇAKMAK

JULY 2024

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

EKSİK METABOLİT MİKTARI TAHMİNİ İÇİN BİR DERİN ÖĞRENME MİMARİSİ

YÜKSEK LİSANS TEZİ

Sadi ÇELİK
(504211530)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Assoc. Prof. Dr. Ali ÇAKMAK

TEMMUZ 2024

Sadi ÇELİK, a M.Sc. student of ITU Graduate School student ID 504211530 successfully defended the thesis entitled "A DEEP LEARNING ARCHITECTURE FOR MISSING METABOLITE CONCENTRATION PREDICTION", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Ali ÇAKMAK** Istanbul Technical University	.....
Jury Members :	**Assoc. Prof. Dr. Mehmet BAYSAN** Istanbul Technical University	.....
	Asst. Prof. Dr. Barış ARSLAN Acıbadem University	.....

Date of Submission : **24 May 2024**
Date of Defense : **12 July 2024**

To my beloved family,

FOREWORD

First of all, I would like to thank my advisor, Assoc. Prof. Ali Çakmak, for his guidance during my thesis and support in my academic studies, letting me join his team and giving me the opportunity to participate in various important events and competitions in Turkey. I would also like to thank my colleagues Barış Can and Aycan Şahin, and all the graduate and undergraduate research students working in the ITU Bioinformatics Lab, for their friendship and sharing. I would like to express my gratitude to my family, who have always been by my side and trusted me.

This thesis was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) [Grant number: 124N069] and the National Center for High-Performance Computing (UHEM) [Grant Number: 1009742021].

July 2024 Sadi ÇELİK

TABLE OF CONTENTS

ABBREVIATIONS

AE	**:** Autoencoder
BPCA	**:** Bayesian Principal Component Analysis
GC	**:** Gas Chromatography
HM	**:** Half Minimum
KNN	**:** K-Nearest Neighbors
KPCA	**:** Kernel Principal Component Analysis
LC	**:** Liquid Chromatography
LOD	**:** Limit of Detection
MAR	**:** Missing at Random
MCAR	**:** Missing Completely at Random
MICE	**:** Multivariate Imputation by Chained Equations
MNAR	**:** Missing Not at Random
MS	**:** Mass Spectrometry
NMDR	**:** National Metabolomics Data Repository
NMR	**:** Nuclear Magnetic Resonance
NRMSE	**:** Normalized Root Mean Squared Error
PCA	**:** Principal Component Analysis
QRILC	**:** Quantile Regression Imputation of Left-Censored Data
RF	**:** Random Forest
SVD	**:** Singular Value Decomposition
SVR	**:** Support Vector Regression
VAE	**:** Variational Autoencoder

SYMBOLS

μ	: Mean
σ	: Standard Deviation
λ	: Arbitrary Small Value of Data Transformers
J	: Jaccard Similarity
W^i	: Weights in a Neural Network
b^i	: Biases in a Neural Network
a	: Activation of a Neuron
z	: Non Linear Activation Function
t	: Time
α	: Learning Rate
β_1, β_2	: Decay Rate Parameters for the Moments
ε	: Random Number to Prevent Division Error
m_t	: Exponential Moving Average Moment
v_t	: Squared Gradient Moment
m_t, v_t	: Bias Corrected Moments
g_ϕ	: Encoder Function of Autoencoders
f_θ	: Decoder Function of Autoencoders
$p_\theta(\mathbf{z})$	: Prior Distribution of VAE
$q_\phi(\mathbf{z}\|\mathbf{x})$	: Parametric Posterior of VAE
$p_\theta(\mathbf{z}\|\mathbf{x})$	: True Posterior of VAE
SS_{res}	: Sum of Squares of Residuals
SS_{tot}	: Sum of Squares of Total

LIST OF TABLES

LIST OF FIGURES

A DEEP LEARNING ARCHITECTURE FOR MISSING METABOLITE CONCENTRATION PREDICTION

SUMMARY

In the last decade, the use of deep learning methods for the diagnosis and treatment of diseases has become a widespread practice in the field of bioinformatics. Metabolomics is an omics science dealing with the identification and measurement of all metabolites in an organism, and can provide a comprehensive analysis of the metabolic profile both in physiological and pathological conditions. Metabolomics data serve as a measure of metabolic function. In particular, relative ratios and perturbations that are outside of the normal range signify disease conditions.

The analysis workflow of metabolomics data involves the application of different bioinformatics tools. The quantification of metabolites is accomplished by the utilization of a wide range of different combinations of mass spectrometry (MS) in conjunction with liquid chromatography (LC), gas chromatography (GC), and nuclear magnetic resonance (NMR) techniques.

For a proper biological interpretation of metabolomic datasets and powerful data analysis, preprocessing is essential to ensure high data quality. In various studies containing metabolite measurements, missing values in the data may affect the performance of the analysis results significantly. In recent years, the application of deep learning-based generative models for the accurate imputation of missing values has gained popularity. Unsupervised generative models like variational autoencoders (VAE) can impute missing values to perform more powerful data analysis.

This work aims to develop effective models that can accurately predict missing metabolite values in metabolomics datasets. To this end, a number of human metabolomics studies from the Metabolomics Workbench and MetaboLights databases are collected. These datasets are heterogeneous in terms of their metabolite sets and the underlying experimental technologies that generated them. Hence, it is challenging to utilize these diverse datasets together to train imputation models. To tackle this challenge, we propose three different models and dataset merging strategies, namely, *Union-based Merging*, *Iterative Similarity-based Merging*, and *Model-guided Agglomerative Merging*.

We perform several experiments to determine the optimal setup configuration for the training pipeline. This includes finding the best initial missing value imputation approach and the most effective data pretreatment scheme. After handling the original missing values and applying a preprocessing pipeline to the input data, k-fold cross-validation is carried out to ensure consistent and reliable model evaluation. Before training, random missingness simulations are performed to mimic different missing value patterns in clinical datasets, and the models are trained with those patterns.

During our empirical evaluation, we observe that the complexity drawback of *IterativeImputer* with *RandomForestRegressor* is more evident in larger datasets. For this reason, *KNNImputer* method is chosen as the standard missing filling method for initial missing values in our proposed merging approaches. Moreover, the application of log transformations with different bases and the Yeo-Johnson transformation of the datasets results in improved VAE model performances. Furthermore, our experimental results show that the performance of the proposed framework scales over large datasets to create accurate metabolite- and dataset-independent imputation models to predict missing values in metabolomics studies.

EKSİK METABOLİT MİKTARI TAHMİNİ İÇİN BİR DERİN ÖĞRENME MİMARİSİ

ÖZET

Geçtiğimiz on yıl içerisinde hastalıkların tanı ve tedavisinde derin öğrenme yöntemlerinin kullanımı bioinformatik alanında artmıştır. Omik bilimler, DNA'nın işlevlerini inceleyen ve hastalığa bağlı genetik varyantları tanımlamaya odaklanan genomik, RNA seviyelerinin genom bazında niteliksel ve miktarlı incelenmesine olanak sağlayan transkriptomik, organizmadaki protein ağlarını ve hücre, doku içindeki protein miktarını inceleyen proteomik ve metabolomikten oluşmaktadır. Metabolomik, bir organizma içinde var olan tüm metabolitlerin tanımlanması ve ölçülmesini inceleyen, fizyolojik ve patolojik durumlarda kapsamlı bir analiz sağlayan bir omik bilimidir. Metabolomik veriler metabolomik fonksiyonun bir ölçüsüdür. Normal ölçüm aralığının dışındaki sapmalar kişideki hastalık durumunu ifade etmektedir.

Metabolomik verilerin analizi birden fazla farklı bioinformatik araçlarının kullanılmasını içerir. Metabolitlerin ölçümleri, sıvı kromatografisi (SK) ve gaz kromatografisi (GK) gibi ön filtreleme işlemlerini içeren çeşitli kütle spektrometrisi (KS) ve nükleer manyetik rezonans (NMR) cihazları ile gerçekleşmektedir. Hastalardan alınan kan, tükürük, idrar gibi çeşitli vücut sıvıları, yapılan çalışmanın hedefine ve hipotezine göre analiz edilmektedir.

Metabolomik veritabanlarının biyolojik açıdan etkili bir biçimde yorumlanması ve güçlü veri analizi için, veri kalitesinin yüksek olması ve ön işleme yapılması gereklidir. Ölçüm cihazlarından alınan ham veriler çeşitli işlemlere tabi tutulur. Bu işlemler, cihazdaki ham sinyallerin işlenmesi, sinyallerin gruplandırılması ve filtrelenmesi aşamalarıdır. Bu işlemler sırasında veri kaybı yaşanması olasıdır. Metabolitlerin göreceli miktar ölçümlerini içeren çeşitli çalışmalarda, verilerde tespit edilen eksik değerler, analiz sonuçlarının performansını olumsuz etkileyebilir. Son yıllarda, eksik değerlerin doğru hesaplanması için derin öğrenme tabanlı üretken modellerin kullanılması oldukça yaygınlaşmıştır. Daha güçlü veri analizi gerçekleştirmek için bu eksik değerleri Değişimli Otomatik Kodlayıcı (DOK) gibi denetlenmemiş üretken modeller kullanarak hesaplayabiliriz. Böylece yeniden üretilen değerler, eksik değerleri de gidererek güçlü bir performans seviyesine ulaşabilir.

DOK modeli, kodlayıcı ve çözücü adında iki ayrı yapay sinir ağı modelinin birbirleriyle dönüşümlü çalışmasıyla oluşur. Kodlayıcı, aldığı veriyi örtülü alan denilen daraltılmış bir uzaya sıkıştırır. Çözücü ise örtülü alandan örnekler alarak ilk veriyi yeniden inşa etmeye çalışır. Örtülü alan, normal otomatik kodlayıcılardan farklı olarak rastsaldır. Bir başka deyişle, bu olasılıksal alandan üretilen örnekler rastgeledir. Örneklerin rastgele olması, geri yayınımlı öğrenim algoritmasının

çalışmamasına, ortalama ve standart sapma cinsinden kısmi türevlerin alınamamasına, dolayısıyla modelin öğrenmemesine sebep olmaktadır. Bu sorunu çözebilmek için parametre değiştirme hilesi uygulanır. Ortalaması sıfır, standart sapması bir olan normal dağılımdan rastgele üretilmiş bir epsilon değeri alınarak rastgele yapılan örnekleme kararlı bir şekle çevrilir. Normalde görüntü üretmek için kullanılan DOK modelleri, çalışmamızda eksik metabolit ölçüm değerlerini tahmin etmek için yeniden tasarlanmıştır. Görüntü üretiminde kodlayıcı ve çözücü ağları evrişimli sinirsel ağ cinsindeyken eksik değer üretiminde tam bağlantılı ağ cinsindedir. Zarar fonksiyonu olarak model, orjinal veri ve yeniden inşa edilen veri arasındaki kayıp ve standart normal dağılım ile örtülü alanda oluşan dağılımın arasında farklılığı ölçen Kullback-Leibler diverjansının toplamını kullanır. Model, zarar fonksiyonunu minimize ederek nöronların ağırlık ve sapmalarını optimize etmeye çalışır.

Bu çalışma, Metabolomik veri kümelerindeki eksik metabolit değerleri doğru bir şekilde tahmin edebilecen etkili modeller geliştirmeyi amaçlamaktadır. Bu amaçla, Metabolomics Workbench ve MetaboLights veritabanlarından çok sayıda metabolomik çalışma topladık. Bu veri setleri, metabolit kümeleri ve bunları üreten temel deneysel teknolojiler açısından heterojendir. Dolayısıyla imputasyon modellerini eğitmek için bu çeşitli veri kümelerini bir arada kullanmak zordur. Bu zorluğun üstesinden gelmek için üç farklı model ve veri seti birleştirme stratejisi öneriyoruz: *Küme Birleşimi Temelli Birleşme*, *İteratif Benzerlik Temelli Birleşme* ve *Modelle Yönetilen Toplu Birleşme*. Bu yöntemler kapsamında verisetleri birleştirilip çeşitli DOK modelleri eğitilmiştir.

Küme Birleşimi Temelli Birleşme yöntemiyle, çalışmalar kendi veritabanlarında ortak metabolit düzlemine getirildikten sonra sanal bir şekilde birleştirilmiştir. Kesişmeyen değerler boş kabul edildiğinden günün sonunda çok seyrek bir veri matrisi eğitime hazırlanmıştır. Her bir veritabanı (Metabolomics Workbench ve Metabolights) için birer DOK modeli eğitilmiştir. *İteratif Benzerlik Temelli Birleşme* yöntemiyle, her bir çalışma için optimal birleştirilebilecek çalışmalar, ölçülen metabolitlerin Jaccard benzerliği üzerinden hesaplanıp, belirli bir seyrekliliği geçmemek kaydıyla kaydedilmiştir. Her bir çalışma için bir tane optimal birleşme seti mevcuttur. Her bir birleşme seti için bir tane DOK modeli eğitilmiştir. Aynı çalışmaları içeren birleşme setlerinden üretilen modeller, tekrarı önlemek için silinmiştir. *Modelle Yönetilen Toplu Birleşme* yöntemiyle ise çalışmalar kendi veritabanlarında ikili ikili birleştirilip, birleşimde oluşan eksikler daha önceki modeller kullanılarak tahmin edilmiştir. İkili birleştirme işlemi, büyük tek bir model oluşturulana kadar devam etmiştir. Toplamda $N_{\log_2 N}$ tane özgün model oluşturulmuştur (N = çalışma sayısı). İlk aşamalarda üretilen modellerin güvenirliliği son modellere göre daha yüksektir.

DOK modellerimizin optimum yapılandırmasını belirlemek için çeşitli ön deneyler yapılmıştır. Bu ön deneyler, verilerde gerçekten var olan eksik ölçümleri hangi yöntemle dolduracağımız ve hangi en etkili veri ön işleme planını uygulayacağımız konusunda sonuçlar vermiştir. Verideki gerçek boşluklar doldurulup, veriler ön işleme hattına sokulduktan sonra tutarlı ve güvenilir bulgular sağlamak için k-katmanlı çapraz doğrulama tekniği kullanılmıştır. Eğitim öncesinde, klinik verisetlerindeki eksiklik örüntülerini taklit etmek için dolu olan verilerde rastgele boşluklar oluşturulmuştur ve

modeller bu eksik verilerle eğitilmiştir. Bu işlemin oluşturulma amacı, ideal koşullarda gelecek yeni bir veride potansiyel olarak bir eksiliğin olmasıdır.

Veri setlerinin boyutu büyüdükçe RastgeleOrmanRegresör'ü daha yavaş çalıştığından KNN doldurma yöntemi, orijinal eksik verileri doldurmada standart yöntem olarak seçilmiştir. Veri kümelerine yapılan log dönüşümlerinin farklı tabanlarla uygulanması ve Yeo-Johnson dönüşümü, DOK modellerinin performansını artırmıştır. Deneysel sonuçlarımız, önerilen çerçeve ile eğitilmiş modellerimizin metabolomik çalışmalardaki eksik verileri doğru bir şekilde hesaplayabildiğini göstermektedir.

1. INTRODUCTION

Metabolomics is a discipline within the field of omics sciences that specifically examines the small-molecule elements of a biological system. It involves the comprehensive assessment of all the metabolites found in cells, tissues, or biofluids. The metabolome refers to the entirety of metabolites present at a specific moment in time. It may reveal the interaction between an organism's genome and its environment, as well as affect cellular physiology by modulating other omics (genomics, transcriptomics, and proteomics) [6]. Factors such as genetic background, epigenomics, diet, and the gut microbiome may change the metabolome. Metabolites are in dynamic equilibrium, may directly reflect the underlying biochemical activity, and are well correlated with an organism's phenotype. Metabolomics reflects the metabolic profile in both healthy and diseased conditions, thus enabling the identification of specific metabolome patterns that may serve as clues for underlying causes of diseases. The non-invasive nature of metabolomics and its relation to the phenotype enable useful applications for studying physiological conditions and disorders. For instance, biomarkers based on altered metabolic pathways may be identified, new physiologic and pathophysiologic mechanisms may be defined to map disease risks, and individual variations in drug response phenotypes may be predicted.

Several bioinformatics tools are used while analyzing metabolomics data. The development of new analytical techniques makes it possible to assay and quantify thousands of metabolites in very small amounts of biological samples. The commonly used technologies for capturing metabolomics data are mass spectrometry (MS) and nuclear magnetic resonance (NMR) [7].

To enhance the biological interpretability of metabolomics datasets and perform more powerful data analysis, raw metabolomics data must be cleaned and preprocessed to ensure high data quality (quality control). Data pre-processing is the set of steps conducted before data pre-treatment, after obtaining raw metabolomics data from

instruments. It involves several tasks like correcting the baseline, aligning the peaks, binning (spectral bins), and filtering out the noise [8]. Data pre-treatment steps include outlier elimination, missing value imputation, normalization, centering, scaling, and transformation. The sequence of pre-treatment steps may vary depending on the existing data and the hypothesis to be tested.

Metabolomics data usually exhibits a combination of missing data patterns [9] categorized as missing not at random (MNAR), missing at random (MAR), and missing completely at random (MCAR) [10]. Missing values occur generally due to either systematic or random loss of data. They are observed as gaps in the data matrix resulting from either the real biological absence of the metabolite in the given sample or from technical challenges.

In the literature, various approaches are presented for imputing missing data for general purposes [11]–[16].Besides the general-purpose missing value imputation strategies, there are specific imputation approaches to handle missing values in metabolomics datasets [9,10,17]–[20]. General-purpose imputation algorithms fail to sufficiently account for natural outliers present in metabolomics datasets. To overcome this challenge, researchers developed specific methods only for metabolomics data. In general, missing value imputation techniques for metabolomics are classified as *Statistical model-based* [9,10], *Machine learning-based* [17,18], and *Generative model-based* [19,20].

Dekermanjian et al. [17] created a missing imputation pipeline for metabolomics that first classifies missingness types, then applies suitable imputation algorithms to fill in the missing values. Jin et al. [18] introduced a method called NetSVR, which utilizes support vector regression (SVR) to forecast absent values in LC/MS metabolomics data by adding network ion interactions. Gomari et al. [19] used an advanced version of autoencoders, variational autoencoder (VAE), a type of neural network architecture, to impute missing values in metabolomic datasets. Zhao et al. [20] proposed a multi-view VAE framework incorporating genomics with metabolomics to extract meaningful relations between multi-omics data and impute missing values in the metabolomics datasets.

None of the existing studies discussed above attempt to integrate various sources of metabolite studies; instead, they target single large, or multiple homogeneous datasets that have the same metabolite set with varying sizes.

An imputation model derived from a single dataset can predict limited numbers of metabolites. That is, a trained model's prediction power is dependent on the breadth of the metabolite set that it is trained with. When new unseen data comes in, the model may not be able to cover and impute all of the metabolites in the dataset. In this thesis, to overcome this limitation, we adopted the integration of diverse datasets to expand the metabolite feature base. In particular, we aim to build effective models that predict accurate values for the missing metabolites. We propose three merging approaches to combine multiple datasets targeted to improve the performance of the imputing system and augment its predictive efficiency. These approaches are *Union-based Merging*, which merges studies with their own databases to provide a large dataset; *Iterative Similarity-based Merging*, which generates an optimal merge set for each study that stays below a specified sparsity threshold; and *Model-guided Agglomerative Merging*, which combines datasets in pairs to create a single large dataset model and imputes missing values using previously trained models to effectively combine metabolite studies while minimizing the likelihood of gaps created by non-overlapping metabolites.

To build and test our models, we obtained and filtered 490 datasets from the Metabolomics Workbench [21] and 66 datasets from MetaboLights [22]. A variety of experiments are performed to evaluate the proposed approaches, identify optimal configurations, and compare their performance to the state of the art. Our experimental results show that to effectively fill original missing values in datasets, *knn* was a robust imputation technique. Log transformations with different bases and Yeo-Johnson transformation helped VAE models capture the patterns in the datasets effectively. Different configurations of our proposed approaches demonstrated fast and robust missing imputation methods for metabolomics research. The best performing approach was the *Model-guided Agglomerative Merging* with an average *Sim-*R^2 score of 0.72.

The contributions of this thesis are summarized as follows:

- We gather and analyze a multitude of metabolomics studies conducted on humans across two different major data platforms, i.e., Metabolomics Workbench [21] and MetaboLights [22]. This is important to demonstrate that the proposed approaches generalize across data sources and datasets. This aspect differentiates our work from the previous metabolomics dataset imputation approaches, which consider only homogeneous datasets from a single source with the same set of metabolites.

- To reconcile the metabolite names included in different datasets, we map metabolites to a common platform. This is important to ensure interoperability between diverse datasets. This way, we can identify shared metabolites in different datasets even if they are referred through different names.

- We develop three different dataset merging strategies: *Union-based Merging*, *Iterative Similarity-based Merging*, and *Model-guided Agglomerative Merging* to combine datasets effectively and reduce the gaps that occurred during the merging process. Union-based merging represents the state-of-the-art approach and forms our baseline. The next two approaches are novel methods proposed in this work.

- We identify and present an optimal setup configuration for the proposed pipeline. This included finding the best initial missing imputation approach for the existing original missing values and the most effective data pre-treatment scheme.

- We build several Variational Autoencoder models for metabolite value imputation by combining several datasets.

- We conducted experiments on our proposed approaches and compared the training and prediction times of their different configurations.

This thesis is organized as follows: In Section 2, we present background information on metabolomics data, the acquisition methods, data pretreatment steps, and a literature review on missing imputation methods. In Section 3, we briefly explain our proposed merging methods and the structure of the VAE model that we implemented. Section 4 gives the results and discussion of our designed experiments. Finally, Section 5 concludes the thesis and provides ideas for future work.

2. BACKGROUND AND LITERATURE REVIEW

2.1 Metabolomics in the Context of the Central Dogma of Molecular Biology

In 1958, Francis Crick developed the theory of Central Dogma, describing how the flow of genetic information in living systems is transferred from DNA to mRNA by transcription and then to protein by the process of translation. In the last century, progressive advances in the methods to study genes, transcripts, proteins, and metabolites led to the development of the "omics" sciences [23] (Figure 2.1).

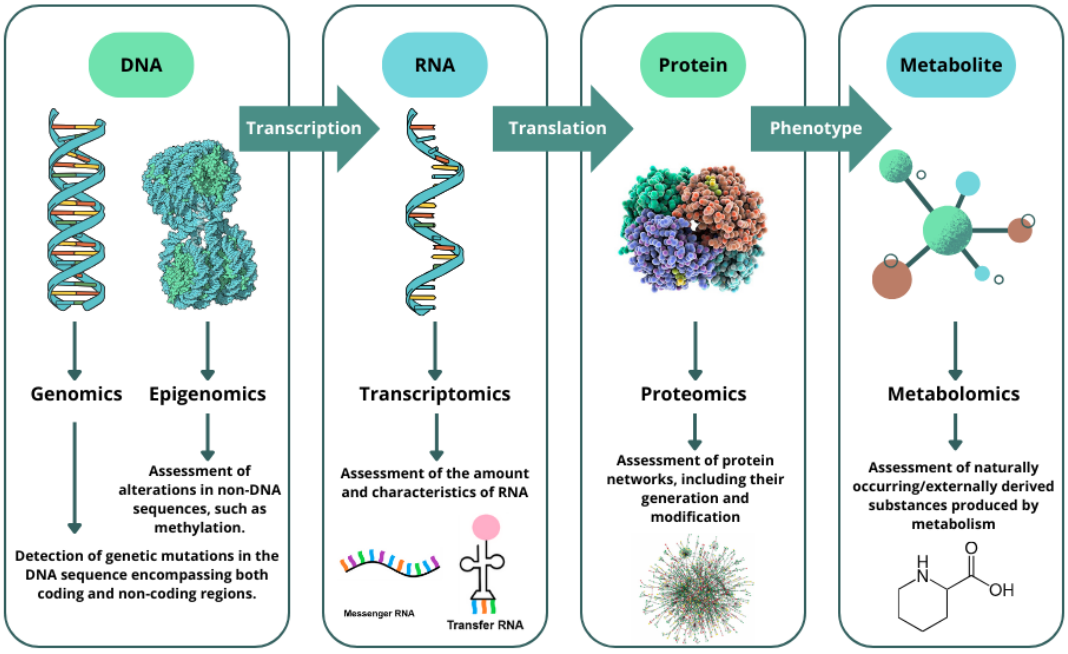

Figure 2.1 : The study of omics sciences and the interconnections between different layers. There may be other non-direct connections and complex links. Metabolomics is the end layer, which is closely linked with phenotype.

Metabolomics is one of the most recent omics sciences, that came later than genomics, transcriptomics, and proteomics. It is a way of "metabolic profiling" for the biochemical processes that take place within an organism, and deals with the identification and measurement of all metabolites in cells and tissues under various conditions [24]. Metabolomics data complements information obtained from other omics fields and provides a map of the metabolic pathway regulation. This

way, it contributes to a better understanding of interactions of the genome with the environmental factors [25]. Thus, it serves as a measure of metabolic function and can represent the overall physiologic status of an organism. The metabolome is the group of all small molecules; amino acids, carbohydrates, and lipids present in cells, tissues, organs, and biological fluids such as blood, urine, etc. Metabolome can affect cellular physiology by modulating other omics [6]. Besides being produced by the organism itself endogenously, metabolites can be the products of microorganisms (e.g., gut microbiota) and may be ingested exogenously from the diet. The metabolome changes rapidly in a dynamic manner based on factors such as epigenomics, diet, nutrition, exercise, the gut microbiome, age, and sex [26].

While classical biochemical methods focus on single metabolites and single metabolic reactions, metabolomics involves the collection of quantitative data on a broad series of metabolites, thus gaining an overall understanding of metabolism [25]. Metabolites, besides reflecting biochemical activities, well correlate with the phenotype of an organism [27]. Metabolomics also aims to define metabolome signatures to understand disease mechanisms. Relative ratios and perturbations that are outside the normal ranges signal disease conditions and can serve as a tool for early diagnosis. Metabolomics has useful applications for studying metabolic disorders. Markers for altered metabolic pathways can be identified, and new physiologic and pathophysiologic mechanisms can be defined that can help map disease risks [24]. Similarly, we can predict individual variations in drug response phenotypes.

The development of new analytical techniques makes it possible to assay and quantify many components of the metabolome. Today, it is possible to measure thousands of metabolites in very small amounts of biological samples.

2.2 Technologies for Measuring Metabolites

Over the course of the last ten years, advancements in technology have sped up the process of identifying and measuring metabolites. Currently, the primary forms of technology used for capturing metabolomics data are mass spectrometry (MS) and nuclear magnetic resonance (NMR) [7].

Mass spectrometry is an analytical method that consists of several steps to identify the fingerprints of the sampled compounds. First, molecules go through ionization with the removal or addition of an electron, resulting in compounds being positively or negatively charged. Then, a spectrometer captures distinct signal peaks, and ions are separated with respect to their charge and mass in a magnetic fielded mass analyzer. At the end, the detector captures and quantifies ions as a spectrum of mass-to-charge ratios (m/z) and the relative intensity of the measured compound. Before the MS analysis steps, liquid chromatography (LC) and gas chromatography (GC) techniques are employed in a percolation column to determine the retention time (the time it takes for a compound to pass the column) of input compounds, aiding the separation of molecules based on their physical and chemical properties. As a rule of thumb, MS and LC/GC methods are used in conjunction for the identification and quantification of metabolites [28]. The discussed steps of mass spectrometry and an example MS instrument are illustrated in Figure 2.2.

NMR, on the other hand, is a powerful method that relies on the absorption and release of energy by atomic nuclei in response to variations in an external magnetic field for the identification of the chemical structure of a compound [29]. The process is repeated several times to reduce interference and optimize signal frequency. Finally, the captured signals are Fourier transformed to show individual radio frequencies that make up the signal and collected frequencies form the NMR spectrum for interpretation. With NMR, one can target different atomic nuclei and measure different types of metabolites. In samples with biological origins, hydrogen (^{1}H-NMR) is the most common targeted nucleus thanks to its abundant occurrence. Besides, phosphorus (^{31}P-NMR) and carbon (^{13}C-NMR) nuclei can be targeted to retrieve additional

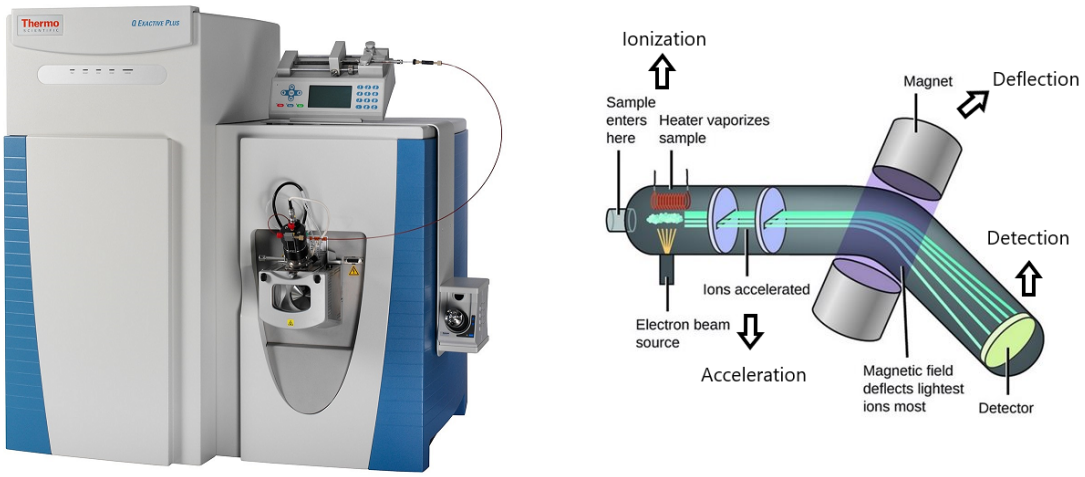

Figure 2.2 : An example MS instrument: The Thermo Scientific Q Exactive Plus Hybrid Quadrupole Orbitrap [1] on the left. Schematic of a general mass spectrometry instrument adapted from [2] on the right.

metabolite measurements [30]. An example NMR instrument, Bruker 400 MHz spectrometer kit, and a schematic of an NMR measurement pipeline are illustrated in Figure 2.3.

Metabolomics measurements can be performed in a targeted or untargeted approach [31]. In targeted studies, the set of metabolites to be measured is decided before the extraction and analysis of the samples. These studies are typically guided by a specific biochemical discovery or hypothesis that prompts the examination of a specific pathway. More reliable, structurally known, and annotated metabolite measurements can be captured within these studies. On the contrary, untargeted studies are done to achieve the global metabolic profile of a sample, but since the chemical identity of almost half of the measured metabolites remains unidentified, annotating and quantifying metabolites is more difficult and takes more time than targeted methods. Despite these challenges, it has huge potential to capture the signals of novel metabolites and provide insights into biological processes.

2.3 Data Pre-processing and Pre-treatment

Metabolomics data may be affected by various factors, such as the type of sample, the sample preparation steps, and the quality of the measurement devices. In addition, it may be influenced by internal biological reasons, such as variations

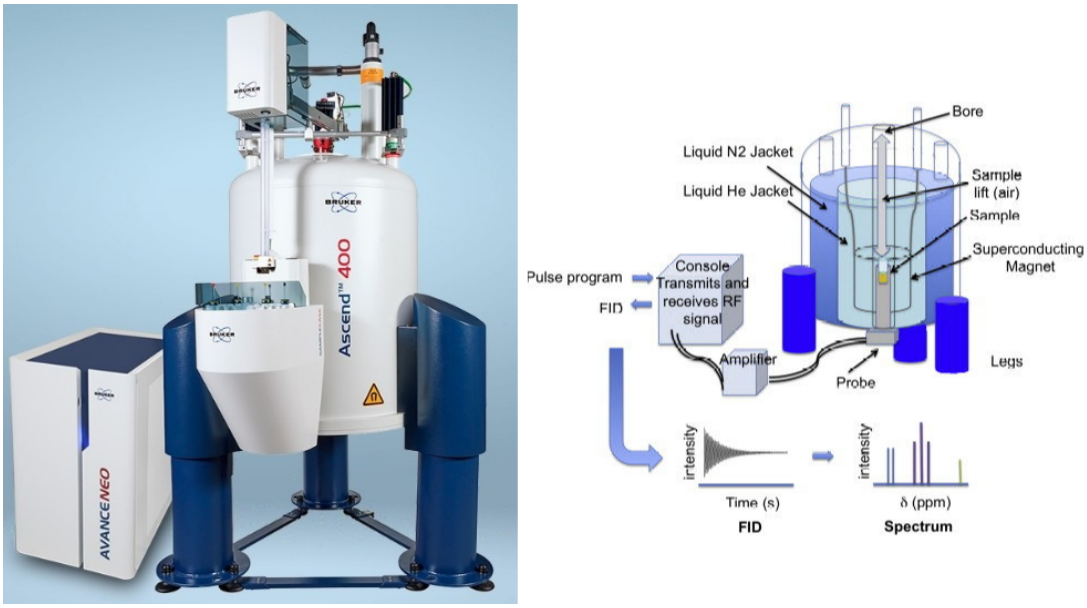

Figure 2.3 : An example NMR instrument [3]: Avance 400 MHz NMR spectrometer, consisting of a 400 MHz Ascend magnet, an Avance Neo console, and a Bruker CryoProbe on the left. Schematic of an NMR measurement pipeline [4] on the right.

in orders of magnitude between measured metabolites and variations in the fold changes of metabolite concentrations [32]. To enhance the biological interpretability of metabolomics datasets and acquire more powerful data analysis, raw metabolomics data must be cleaned and pre-processed. Data pre-processing and data pre-treatment are two separate processes. Data preprocessing is the set of steps conducted before data pre-treatment, after obtaining raw metabolomics data from instruments, such as correcting baseline, aligning the peaks, binning (spectral bins), and filtering out the noise [8]. On the other hand, conventional data pre-treatment steps include outlier elimination, missing value imputation, normalization, centering, scaling, and transformation, which can be seen in Figure 2.4. The sequence of pre-treatment steps may vary depending on the existing data and the hypothesis to be tested. In the following subsections, we explain the data pre-treatment steps, which also involve data imputation.

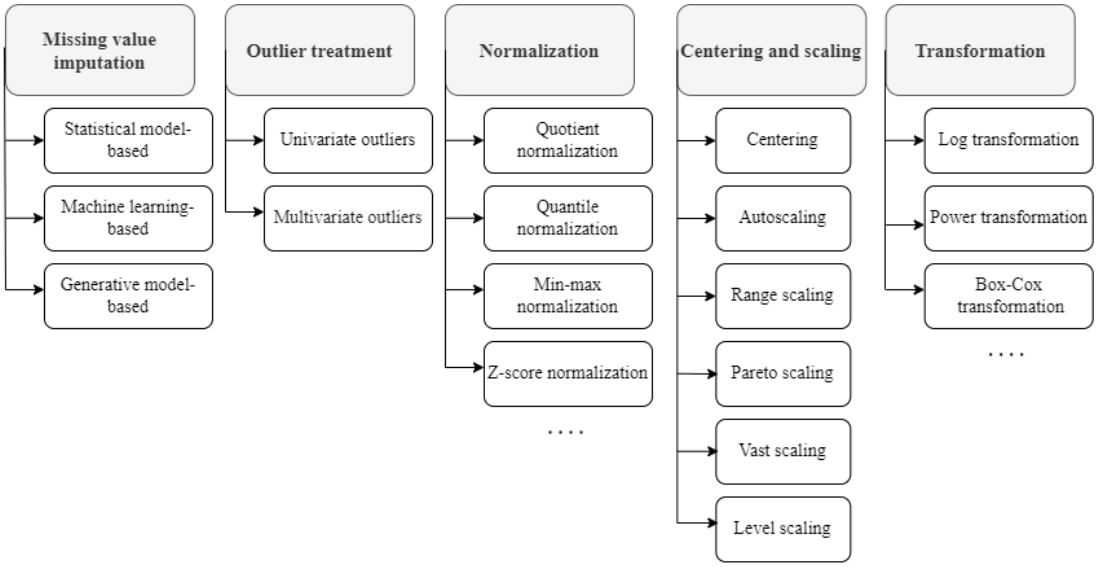

Figure 2.4 : Common data pretreatment steps for MS and N̈M̈R-based metabolomics data. The sequence of pretreatment steps may vary depending on the existing data and the hypothesis to be tested.

2.3.1 Outlier treatment

An outlier can be defined as either a single data point that falls beyond the expected range of values (known as univariate outliers) or a group of data that shows unexpected deviations from the overall variability of the data (known as multivariate outliers). Univariate outliers in metabolomics data can be identified by examining the overall distribution of measurements. For each metabolite, the data point can be eliminated from the analysis if the concentration is more than three or four standard deviations away from the mean in both positive and negative directions. Alternatively, one can extract descriptive statistics such as the first quartile (Q1), median, and third quartile (Q3) of a metabolite and discover outliers that go outside of the upper and lower fences. In some cases, outliers are only noticeable when examining the correlation between data across multiple features. Multivariate outliers are based on multiple factors and features simultaneously. To eliminate this type of outlier, one can use multivariate statistical techniques such as Mahalanobis [33] and Jackknife distance [34] or K-Nearest Neighbors [35] to compare the sample with the distributions of the

other variables. When the elimination of outliers leads to a substantial loss of data, they are replaced with reasonable values within normal ranges.

2.3.2 Normalization

Data normalization can refer to both patient/sample-wise and metabolite-wise operations. Patient-wise operations, which are referred to as normalization, aim to reduce sample variation. On the other hand, metabolite-wise operations, which include centering, scaling, and transformation, adjust the variance of metabolites to reduce heteroscedasticity [8]. These pre-processing steps are typically performed after handling the outliers and imputing missing metabolite values. Variations in metabolomics data may occur for a variety of reasons. Samples collected from different parts of the body, such as urine or saliva, may be affected by diet and water intake. Certain studies may take place at different times of the year, and external conditions such as temperature. Besides, the storage environment can have an impact on the measurement device's performance. Thus, normalization is performed to eliminate variance in the data. Probabilistic quotient normalization (PQN) [36], quantile normalization (QN) [37], which turns every mean value into the same variable for general use, and constant sum normalization (CSN), which is specifically used for NMR metabolomics, are the most common normalization methods used in the field of metabolomics.

2.3.3 Centering and scaling

Captured metabolites within the same studies can have different value ranges. The main purpose of centering and scaling is to reduce each metabolite in the same unit range, enabling accurate assessment while accounting for heteroscedasticity, and facilitating analysis of different subsets of data with variable sizes [38]. Centering involves redistributing all metabolites around zero by subtracting their mean levels. Scaling involves the process of multiplying each metabolite with a scaling factor. This scaling factor can be determined by either using the mean value of a metabolite, which involves level scaling and linear baseline scaling, or by using the standard deviation value of a metabolite, which involves range scaling [39], autoscaling (unit variance

scaling) [40], vast scaling [41] and Pareto scaling. Every method features a distinct set of advantages and disadvantages, which vary depending on the hypothesis being tested. The detailed centering and scaling formulations are presented in Table 2.1.

2.3.4 Transformation

Transformation, however, takes the input data and transforms it without considering any internal mean or standard deviation value. It is commonly performed to convert the skewed data distributions into a more normal-like distribution and stabilize the variance due to the outliers. Log transformation [42] using the natural logarithm $\ln(x)$ is a frequently selected method in metabolomics studies. Alternatively, one might employ a power transformation [43] as an alternative approach. The Box-Cox transformation [44] is a suitable transformation for positive metabolite variables, where a λ parameter is estimated with MLE (maximum likelihood estimation). Also, the Yeo-Johnson transformation [45] can be used with additional optimization for positive metabolite variables. Since two of the methods can be transformed with different λ values, they add extra complexity to the interpretation by finding the best value for λ value. The detailed transformation formulations are in Table 2.1.

An optimal λ value that maximizes the log-likelihood objective function (equation 2.1) within a given bound, can be estimated starting with an arbitrary small λ value, where x_i is the raw metabolite value, y_i is its transformed version with the initial λ, and n is the number of patients in the study.

$$L(\lambda;\mu,\sigma) = \frac{-n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu) + (\lambda-1)\sum_{i=1}^{n}\log(x_i) \quad (2.1)$$

The symbol $\bar{x}_i$ represents the average value of a metabolite concentration, as defined by equation 2.2. Similarly, the symbol σ_i represents the measure of variability in a metabolite, known as the standard deviation, as defined by equation 2.3. The variable x_{ij} denotes the concentrations of individual metabolites in a tabular research dataset, while y_{ij} represents their corresponding modified versions. All logarithmic expressions represent the natural logarithm to the base e.

Table 2.1 : Formulations of the discussed data pre-treatment steps.

Method	Formulation
Probabilistic Quotient Normalization	1. Perform integral normalization 2. Calculate the quotients based on reference spectrum 3. Calculate the medians of the quotients 4. Divide all variables by this factor
Quantile Normalization	1. Assign a rank to each metabolite in a column 2. Order each column according to ranks 3. Calculate means for each row 4. Replace each value according to the ranks assigned (First rows mean become rank one's value and so on). If there are the same ranks, take the average of the corresponding. 5. Restore the order of the metabolites before ranking
Min-max normalization	$y_{ij} = \frac{x_{ij} - x_{i_{\min}}}{(x_{i_{\max}} - x_{i_{\min}})}$
Mean centering	$y_{ij} = x_{ij} - \bar{x}_i$
Unit variance scaling	$y_{ij} = \frac{x_{ij} - \bar{x}_i}{\sigma_i}$
Pareto scaling	$y_{ij} = \frac{x_{ij} - \bar{x}_i}{\sqrt{\sigma_i}}$
Range scaling	$y_{ij} = \frac{x_{ij} - \bar{x}_i}{(x_{i_{\max}} - x_{i_{\min}})}$
Vast scaling	$y_{ij} = \frac{(x_{ij} - \bar{x}_i)}{\sigma_i} \cdot \frac{\bar{x}_i}{\sigma_i}$
Level scaling	$y_{ij} = \frac{x_{ij} - \bar{x}_i}{\bar{x}_i}$
Log transformation	$y_{ij} = \log(x_{ij})$
Power transformation	$y_{ij} = \begin{cases} x_{ij}^{\lambda} & \lambda \neq 0 \\ \log(x_{ij}) & \lambda = 0 \end{cases}$
Box-Cox transformation	$y_{ij} = \begin{cases} \frac{x_{ij}^{\lambda} - 1}{\lambda} & \lambda \neq 0 \\ \log(x_{ij}) & \lambda = 0 \end{cases}$
Yeo and Johnson transformation	$y_{ij} = \begin{cases} \frac{(x_{ij}+1)^{\lambda} - 1}{\lambda} & x_{ij} \geq 0, \lambda \neq 0 \\ \log(x_{ij} + 1) & x_{ij} \geq 0, \lambda = 0 \\ -\frac{(-x_{ij}+1)^{2-\lambda} - 1}{2-\lambda} & x_{ij} < 0, \lambda \neq 2 \\ -\log(-x_{ij} + 1) & x_{ij} < 0, \lambda = 2 \end{cases}$

$$\bar{x_i} = \frac{1}{N}\sum_{j=1}^{N} x_{ij} \tag{2.2}$$

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(x_{ij} - \bar{x_i})^2} \tag{2.3}$$

2.4 Missing Value Imputation

Missing data patterns are generally categorized as missing not at random (MNAR), missing at random (MAR), and missing completely at random (MCAR) [10]. Metabolomics data usually exhibits a combination of various kinds of missing data patterns [9]. In the context of metabolomics, not at random values are influenced by an unobserved factor, such as when metabolite signals fall below a particular instrument's detection threshold. On the other hand, at-random values are affected by observed factors, due to insufficient data processing or technical problems like overlapping peaks in chromatography-based analysis. This can lead to chemically similar metabolites having closely matching retention times, making their peaks indistinguishable and resulting in only one metabolite being detected in each sample. Finally, completely at-random values exhibit a random distribution (i.e., the likelihood of being missing is consistent across all variables) and are affected by neither the observed nor unobserved missing values.

In the literature, there are several works describing approaches for imputing missing values into metabolomics datasets [9,10,17]–[20]. Some researchers conduct simulations and evaluations of the currently existing missing imputation strategies [9,10], while others come up with novel imputation approaches to handle missing values in real-world metabolomics studies [17]–[20]. Missing imputation techniques can be categorized into three subgroups: statistical model-based, machine learning-based, and generative model-based. Our proposed work presented in this thesis falls under the category of generative model-based approaches.

2.4.1 Statistical model-based

Wei et al. [9] benchmark a total of eight imputation methods for various missing value types using four real-world GC/MS-LC/MS metabolomics datasets. Initially, a random missing value creation procedure is performed on the datasets. The missingness rate ranges from 2.5% to 50% with increments of 2.5% for MAR/MCAR categories, and from 4% to 80% with increments of 4% for MNAR category. Additionally, a random quantile cut between 30% and 60% is used to generate missing data for MNAR. Then, they apply zero, half minimum (HM), K-Nearest Neighbors (KNN), Singular Value Decomposition (SVD), Random Forest (RF), and Quantile Regression Imputation of Left-Censored Data (QRILC) for MNAR, and mean, median, KNN, SVD, and RF for MAR/MCAR. They evaluate the performance of the approaches by calculating the normalized root mean squared error (NRMSE) and sum of ranks (SOR) based on NRMSE between the imputted and actual values. The RF algorithm demonstrates superior performance in handling missing data with MAR/MCAR, whereas the QRILC algorithm excels in dealing with left-censored data (caused by the device detection limit), which can be categorized as MNAR.

Do et al. [10] perform another comparative study of existing missing value imputation methods on metabolomics datasets. Initially, they study the arrangement of absent information in serum samples obtained from the German KORA F4 cohort in a metabolomics experiment. The researchers assess 31 imputation techniques inside a simulation framework to create synthetic data that replicates the missing data patterns found in actual datasets and employ these techniques on clinical metabolomics data for biological validation. The study's primary findings indicate that missing data are caused by the limit of detection (LOD) effects, which vary depending on the day of analysis. The study also reveals that multivariate imputation techniques, such as KNN-based imputation and MICE, outperform other methods in predicting missing data. Since MICE is computationally challenging, authors suggest KNN to be a more robust imputation method for untargeted metabolomics studies.

2.4.2 Machine learning-based

Dekermanjian et al. [17] introduce a two-step methodology, MAI (a missing mechanism-aware) that takes into account the missingness type when estimating missing values in metabolomics datasets. A random forest classifier was used to classify the missing mechanism for each absent value in the dataset. In order to complete the missing information, a combination of imputation methods tailored to the anticipated type of missingness mechanism (i.e., MNAR or MAR/MCAR) is performed. Simulations suggest that utilizing a combination of single imputation, a method they had developed for estimating the expected value of a metabolite with left-censoring for MNAR values, and RF imputation [12] for MAR/MCAR values, works best in most situations.

An inherent limitation of the MAI framework is its dependence on the precise classification of missing values. Furthermore, the evaluation of different multi-class classifiers has not been attempted. Since artificially classifying missing data may lead to neglecting the genuine underlying causes, we choose not to pre-classify missing values in our proposed frameworks.

Jin et al. [18] proposed an algorithm, NetSVR, that employs support vector regression (SVR) to predict missing values in LC/MS metabolomics data by considering features in the network's neighborhood and adduct ion interactions. The *NetSVR* method is compared with other commonly used (KNN, bayesian principal component analysis (BPCA), simple linear regression (SLR), singular value decomposition (SVD), and imputation by inserting single values such as mean, median, and half-minimum) imputation algorithms, and is shown to outperform them in terms of accuracy based on NRMSE.

2.4.3 Generative model-based

Another group of studies adopt the generative model approach and attempt to reconstruct novel data from trained samples. In this thesis, we also employ generative models to impute missing values.

Gomari et al. [19] employ a more advanced version of autoencoders, variational autoencoder (VAE) [46]. VAE is a type of neural network architecture with fully connected layers. The authors exploit VAE as an unsupervised dimensionality reduction strategy to capture nonlinear changes and impute missing values in metabolomics datasets using a model trained with 217 metabolite measurements obtained from 4644 patients in the TwinsUK study [47]. A comparison is made between the employed VAE model and different variations of the PCA [48] and KPCA [49] models, including cosine, polynomial, RBF, and sigmoid KPCA. The authors show that non-linear dimensionality reduction techniques outperform the linear PCA method in accurately reconstructing the original data. Additionally, VAE latent dimensions demonstrate stronger connections with diseases such as *acute myeloid leukemia*, *schizophrenia*, and *type 2 diabetes*.

Gomari utilizes a uniform, singular dataset including a known set of metabolites. That is, it does not consider cases where data comes from different datasets and data sources. It also does not consider that datasets may have different metabolite sets. Our work, on the other hand, integrates different studies from different data sources to enhance the diversity of our training database.

Zhao et al. [20] propose a Multi-View Variational Autoencoder (MVAE) framework that combines genomics and metabolomics datasets to extract meaningful relationships between multi-omics data and impute missing values in the metabolomics datasets. They collected both whole-genome sequencing (WGS) and metabolomics data from the same individuals. Their cohort includes 1110 subjects from the Louisiana Osteoporosis Study (LOS) [50]. MVAE integrates three different genomic aspects: burden scores derived from template metabolites, polygenic risk scores (PGS), which offer a quantification of an individual's susceptibility to a certain disease based

on their genetic makeup, and linkage disequilibrium (LD)-pruned single nucleotide polymorphisms (SNPs), which eliminate repetitive genetic mutations from collected samples. The MVAE model's performance is evaluated using the mean absolute percentage error (MAPE) and the R^2 score, using various pre-determined cut-off levels. Then, the authors compare the robustness of their metabolic data imputation method with the existing approaches such as KNN, ridge regression, RF, gradient boosting regression (GBT), and SVM.

Zhao integrates multiple types of omics data to increase the imputation method's power. Integrating different aspects of human biology is a novel research area. However, obtaining both genomic WGS data and metabolomic data from the same individuals is often impractical, costly, and hard to achieve in clinical settings.

In addition to the metabolomics domain, generative models are also employed for data imputation in a variety of other fields. Zheng *et al.* [14] employ a diffusion model approach called TabCSDI for the imputation of time-series data and demonstrate the effectiveness of this method for imputing numerical variables. Gondora and Wang [15] develop an overly complete version of denoising autoencoders (unlike normal auto-encoders, in a structure that expands and shrinks) in a multiple imputation framework (MIDA) to reconstruct noise-free data. Their results demonstrate MIDA's superiority over MICE. Yoon *et al.* [16] build Generative Adversarial Imputation Nets (GAIN) with an extra hint generator to effectively represent the original data distribution, and compare its variations.

3. MATERIALS AND METHODS

In this section, we discuss metabolomics data sources, data preprocessing steps, imputation methods for imputing real missing values, three merging approaches to efficiently integrate diverse datasets, and the baseline variational autoencoder (VAE) imputation model in detail.

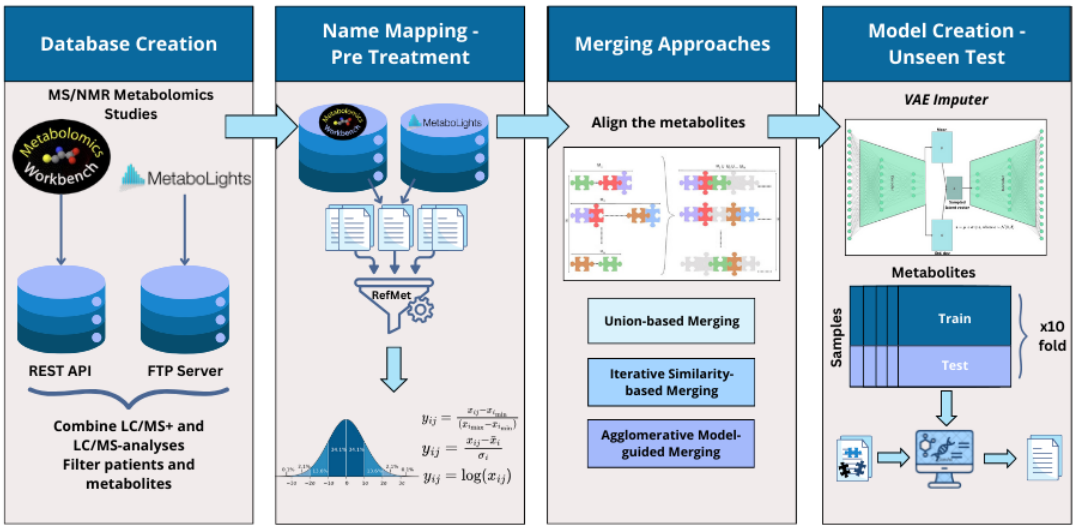

Figure 3.1 : Illustration of the proposed deep learning-based pipeline for imputing missing metabolite values.

3.1 Database Construction

Metabolomics Workbench [21] is a public repository that provides a platform to analyze and share large numbers of metabolomics studies belonging to various species. The metabolites included in studies are linked to records in a database that enables researchers to cross-reference for comparative analyses. The URL-based REST API of Metabolomics Workbench allows researchers to explore and download studies, analyses, genes, and compounds along with metadata analysis, phenotype filtering, and metabolite identification. Metabolomics Workbench contains 490 human studies out of 2612 total publicly available studies (as of January 2024). Hence, in our database, 490 studies are included from Metabolomics Workbench.

MetaboLights [22] is another publicly available data repository for metabolomics studies founded in 2012 by the European Bioinformatics Institute (EMBL-EBI). The database is structured in two main layers: the repository and reference layers. The repository layer contains original research data from various metabolomics investigations, including unprocessed experimental raw data and related metadata. On the other hand, the reference layer stores information about individual compounds and metabolites, including their chemical, analytical, and biological properties. In our database, a total of 66 out of 1345 studies that contain only human metabolites (as of October 2023) are retrieved from MetaboLights.

Patients of different ages, genders, and phenotype characteristics are included in our database. Metabolomics datasets are generated worldwide by various laboratories using a range of instruments and extraction techniques. Our database statistics are summarized in Table 3.1. RefMet mapping and filtering steps are explained next.

Table 3.1 : Our dataset summary statistics.

Database	Statistics		
	Studies	Patients	Metabolites
Workbench (Original)	490	43045	87774
Workbench (RefMet)	490	43045	18552
Workbench (RefMet + Filter)	437	40005	18045
MetaboLights (Original)	66	10652	22326
MetaboLights (RefMet)	66	10652	7295
MetaboLights (RefMet + Filter)	59	9700	7265

3.2 Data Processing

3.2.1 Creating study datasets

A metabolomics study may involve the analysis of different types of data, including positive ion mode (LC/MS+) and negative ion mode (LC/MS-) reverse phase LC-MS metabolite amounts. Due to the lack of consistency in the investigations and analyses, it is necessary to combine the analysis files in order to obtain reproducible study datasets. Study files are constructed by merging these sub-measurements based on patients, with an initial filtering applied during the process. More specifically, in

cases where two or more identical metabolites existed within the same study, their values were averaged. For positive and negative reverse phase concentrations, the maximum value was taken element-wise if the same metabolites had different values. Any uncaptured or blank metabolites, as well as studies with less than 50 metabolites, are discarded from the study database. Consequently, each study yielded a single dataset after this process.

3.2.2 Metabolite mapping

Merging metabolomics datasets from several organizations and platforms presents a substantial challenge for performing comparative analyses involving all studies. The primary reason is that many metabolites lack a universally recognized common name used across all research groups. As one solution, metabolites in each study are mapped to RefMet [51]. RefMet is a reference metabolite name database that is created by utilizing several analyses and combinations of metabolomics research collected from the Metabolomics Workbench. RefMet can be accessed online. It offers a wide range of browsing and searching capabilities, along with API support. Patients with fewer than 50 metabolites were removed after being mapped to a RefMet standardized name. If there are no remaining patients in a dataset, the corresponding study is excluded from our database.

3.2.3 Missing values and imputation methods

Some metabolite measurements may be missing in a given dataset. Typically, researchers choose to exclude such metabolites from the data or fill them with estimated values. Both approaches can potentially create bias and decrease the statistical power in subsequent analyses, such as examining the relationships between metabolites and clinical factors [17]. Metabolomics Workbench studies exhibit a median missingness value of 1.86%, with the range spanning from a minimum of 0.00% to a maximum of 75.22%. The first quartile is 0.00%, while the third quartile is 12.72%. MetaboLights studies exhibit a median missingness value of 0.00%, with the range spanning from a minimum of 0.00% to a maximum of 80.39%. The first quartile is 0.00%, while the third quartile is 11.62%. Missing values in the studies

were imputed using the scikit-learn *KNNImputer* method with a neighborhood size of 5. Data that was originally missing is masked while calculating the loss function in the VAE model.

3.3 Dataset Merging Approaches

The merging approaches proposed in this thesis aim to place all studies in a given database to a similar feature space, as illustrated in Figure 3.2. First, the union of metabolites in all studies is obtained. Subsequently, the feature space is expanded based on the provided missing values for each study. After incorporating all new metabolites into a study, the metabolites are sorted in a standard manner, continuing this process until all studies are aligned in the same feature space, as depicted on the right side of Figure 3.2.

The merging process relies on metabolite-level integration to solve the measurement sensitivity problem. The patient-based merging, which contributes to expanding the dataset, is explained. In patient-based merging, datasets are normally concatenated without ever considering the gaps that may occur during the merge procedure. In metabolite-level integration, the approaches described below handle the missing values created by non-overlapping metabolites differently. For the first two approaches, these values are filled with random uniform values between 0 and 1. In the last merging approach, they are filled with cross-trained model predictions.

3.3.1 Union based merging

Models trained on specific, small datasets had a limited number of metabolites. In order to create more comprehensive models that include a wider range of metabolites, multiple data sets need to be combined. Union-based merging constructs an expansive training dataset that covers a wide variety of metabolites and patients, much larger than the datasets used in state-of-the-art research. In particular, first, missing values in each individual dataset are filled (using *KNNImputer*). Then, we merge every possible study from each source database and create a single big study for both Metabolomics Workbench and MetaboLights. Next, patients with less than 50 metabolites are filtered

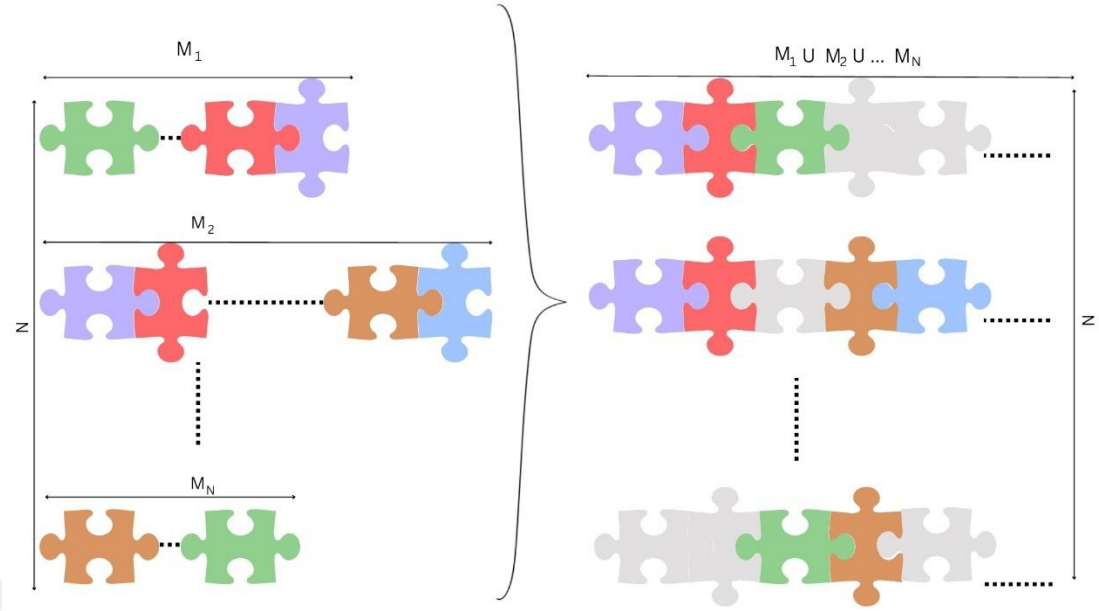

Figure 3.2 : Illustration of the merging problem. The feature space is aligned and expanded based on the available metabolites across the studies.

out. Furthermore, the metabolites that are measured in less than 50 patients are also excluded.

The merged Metabolomics Workbench data exhibits a sparsity of 97.47%, covering 10,563 unique metabolites. Among the metabolites, *Phenylalanine*, *Tryptophan*, and *Tyrosine* have the highest levels of completeness, with sparsity values of 24.53%, 26.76%, and 28.48%, respectively. The merged MetaboLights data exhibits a sparsity of 94.15%, covering 6,047 unique metabolites. Among the metabolites, *Tryptophan*, *Phenylalanine*, and *Hippuric acid* have the highest levels of completeness, with sparsity values of 12.88%, 19.26%, and 23.32%, respectively.

Due to the broad scope of metabolomics research, studies target different pathways and reactions. Thus, their combination results in large and sparsely populated data. Populating this sparse data with synthetic values and constructing a model from it does not yield viable outcomes that are relevant to clinical analysis. This motivated the development of more sophisticated algorithms that aim to decrease the sparsity of merged studies.

3.3.2 Iterative similarity based merging

The objective of the Iterative Similarity-based Merging algorithm is to identify studies that can be merged with the smallest sparsity possible. In particular, for each study, an optimal merge set is constructed and recorded (Figure 3.3). Algorithm 1 provides a pseudo-code for Iterative Similarity-based Merging. First, pairwise Jaccard similarities (Eq. 3.1) are calculated between each study pair based on shared metabolites (lines 2-5).

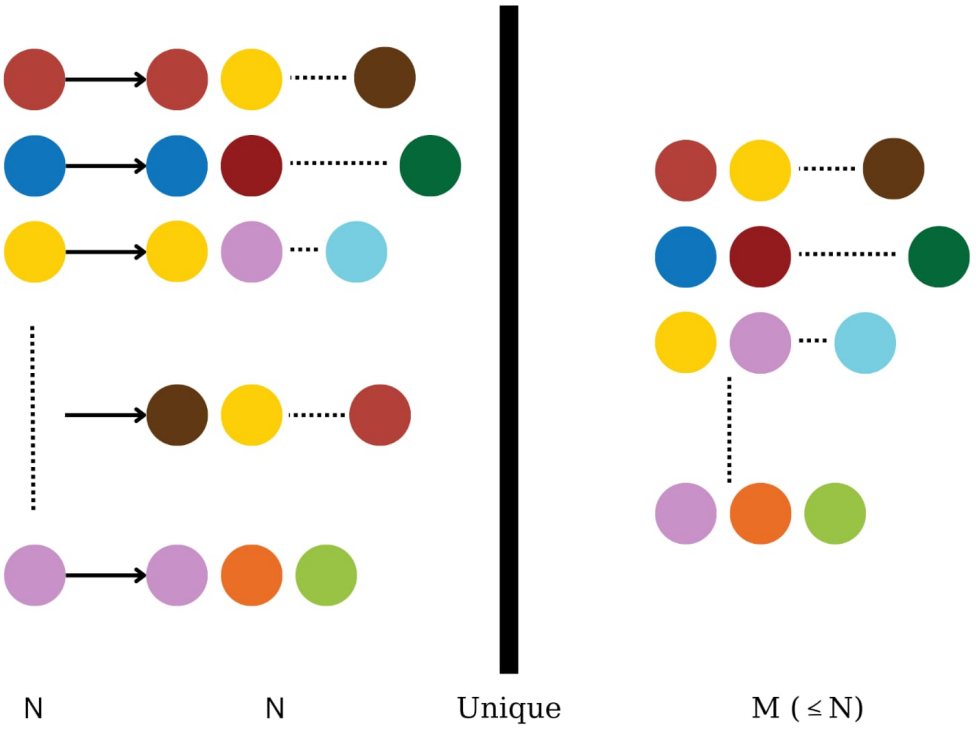

Figure 3.3 : Illustration of the dataset merging phase of *Iterative Similarity-based Merging*. An optimal merge set for each study is created. Unique merge sets are obtained, and models are trained with them.

Then, the algorithm chooses an initial study and ranks other studies based on their similarity score (line 7). Next, the merge operation is performed in the order of dataset similarity scores (lines 12-24). After each iteration, the sparsity of the merged dataset is calculated (line 16). If the sparsity exceeds a predefined threshold, the algorithm stops and returns the currently merged dataset. In cases where the sparsity exceeds the threshold after the first merge attempt, the algorithm returns the initial study as the merge set (lines 17-21). For each study in the database, a model is trained on its

Algorithm 1: Iterative Similarity-based Merging - Generate Models

Input : Metabolite studies in database: *StudyList*,
Sparsity threshold: θ
Output: Similarity based models: M_{iter_N}

Initialize: *mergedStudyDB*;
foreach *pair of studies* (S_1, S_2) *in StudyList* **do**
    $m_1 \leftarrow S_1.metabolite\_names$;
    $m_2 \leftarrow S_2.metabolite\_names$;
    Calculate $jaccard\_similarity(m_1, m_2)$;
foreach *study S in StudyList* **do**
    rankedList $\leftarrow$ **Rank** studies with Jaccard sim ;
    initialStudy $\leftarrow$ `S.data`;
    mergedStudy $\leftarrow$ `S.data`;
    previousMergedStudy $\leftarrow$ `None`;
    sparsity $\leftarrow$ *calculate_sparsity(mergedStudy)*;
    foreach *study* S_{MS} *in rankedList* **do**
        $S_{MS} \leftarrow$ **Find** most similar study with S;
        vals $\leftarrow$ define zeroes array;
        $mergedStudy \leftarrow MergeStudies([S, S_{MS}], vals)$;
        sparsity $\leftarrow$ *calculate_sparsity(merged study)* ;
        if *sparsity* $> \theta$ **then**
            if *previousMergedStudy is not* None **then**
                yield *previousMergedStudy*
            else
                yield *initialStudy*
        previousMergedStudy $\leftarrow$ *mergedStudy*;
    yield *mergedStudy*
    Add: yielded study to *mergedStudyDB*;
Remove: duplicate entries in *mergedStudyDB*;
foreach *mergedStudy in mergedStudyDB* **do**
    $M_{iter_N} \leftarrow GenerateVAEModel(mergedStudy)$;
    Store: M_{iter_N} to be used for imputing;

merge set and stored to impute new unseen data (lines 26-28). Due to the possibility of overlapping merge sets, identical merge sets are removed (line 25).

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.1}$$

In order to enhance the process of creating merge sets, we have suggested the utilization of a weighted Jaccard score. The score considers the contribution of the metabolites within the given studies. For Jaccard similarity, the contribution of intersecting metabolites to the numerator is 1. However, we adjust this contribution based on the range of each individual metabolite in the weighted Jaccard score as expressed in Eq. 3.2.

$$\begin{aligned} WJ(A,B) &= \frac{(|A \cap B|)_{cont}}{|A \cup B|} \\ cont &= 1 - \frac{|min_1 - min_2| + |max_1 - max_2|}{max(max_1 - max_2) - min(min_1 - min_2)} \end{aligned} \tag{3.2}$$

When new, unseen data arrives, models are scored and sorted based on the similarity between the new data and existing merge sets. The new data is imputed using a predefined number of most similar models, weighted averaging the cells with the given similarity scores of the models. For each missing value y_{ij} in the dataset, the weighted averages of imputation results are calculated in Eq. 3.3. The models that have predictions of NaN, which do not contribute, are excluded from the calculation of the weighted average.

$$y_{ij}^{final} = \frac{\sum(y_{ij}^{pred} * sim\text{-}score)}{\sum sim\text{-}score} \tag{3.3}$$

3.3.3 Model guided agglomerative merging

The integration of diverse studies significantly increases the number of models to be utilized. However, it introduces the challenge of sparsity. To address the sparsity problem, we propose another method to fill the non-intersecting metabolites in the study, treating them as observed values based on the output of cross-study models.

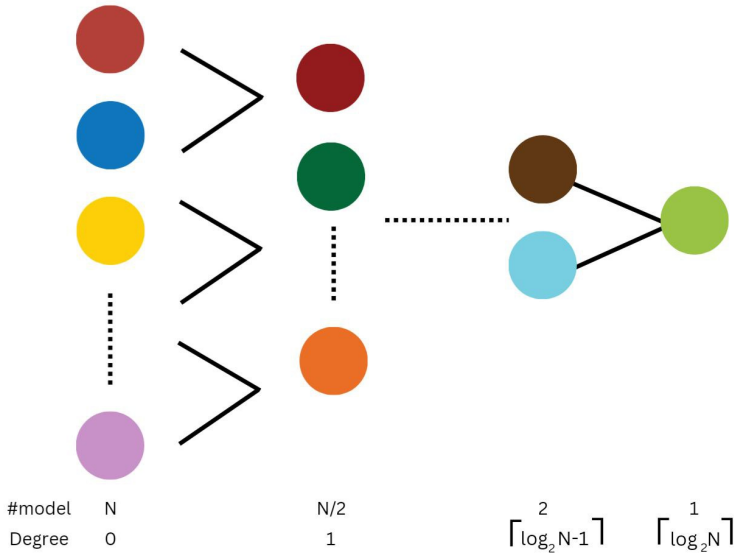

Figure 3.4 : Illustration of the dataset merging and model creation phases of Approach 3. Studies are artificially merged into pairs. The predictive power of the models decreases when going to a single model.

In this approach, studies are combined in pairs at each step. The merging process continues until all studies are used in a single model (see Figure 3.4). Algorithm 2 outlines the steps carried out in this approach. The data structure of binary trees has a height of 0 as an initial. As they grow, their depth increases linearly. We were inspired by the definition and proposed an imputation idea. The models obtained at each step are dependent on the models from the previous step. Therefore, the number of variables dependent on the models increases at each step. The models created initially are considered to have a degree of 0, and they are not dependent on any other models. At each step, imputation models are created for all studies in the study pool, and the resulting models are kept in the relevant degree. Subsequently, Jaccard similarity is used to calculate metabolite-based similarity values between studies.

As shown in lines 19-21, models $S1$ and $S2$ are used reciprocally to fill the non-overlapping metabolites in the other data set. Then, the datasets are merged. The merged studies are added to the new pool, which will be used in the next step. $S1$ and $S2$ studies are then removed from the current pool as well as the similarity matrix. This process continues until no studies remain to be merged. If there is only one unmergeable study left in the pool, it is added to the new pool to be used in the next step. Afterward, the current study pool is updated, and the degree is increased by

Algorithm 2: Model-guided Agglomerative Merging - Generate Models

Input : Metabolite studies in database: *StudyList*
Output: Agglomerative models: M_{Agg_N}

$degree \leftarrow 0$;
$vaeModels \leftarrow$ define array;
while *StudyList has element* **do**
    foreach *study S in StudyList* **do**
        $model \leftarrow GenerateVAEModel(study)$;
        $vaeModels[degree][S] \leftarrow model$;
    $sims \leftarrow$ define list;
    foreach *unique pair of studies* (S_1, S_2) *in StudyList* **do**
        $m_1 \leftarrow S_1.metabolite\_names$;
        $m_2 \leftarrow S_2.metabolite\_names$;
        $similarity \leftarrow jaccard\_similarity(m_1, m_2)$;
        $sims[S_1, S_2] \leftarrow similarity$;
    $nextStudyList \leftarrow$ define list;
    while *sims has 2 or more elements* **do**
        $vals \leftarrow$ define array;
        $S_1, S_2 \leftarrow$ Most similar pair in *sims*;
        $M_1 \leftarrow vaeModels[degree][S_1]$;
        $M_2 \leftarrow vaeModels[degree][S_2]$;
        $vals[S_1] \leftarrow M_2(S_1)$;
        $vals[S_2] \leftarrow M_1(S_2)$;
        $S_{1,2} \leftarrow MergeStudies(list(S_1, S_2), vals)$;
        $nextStudyList.append(S_{1,2})$;
        Remove: S_1, S_2 from *StudyList*;
        Remove: all pairs of S_1, S_2 from *sims*;
    if *StudyList has a element which is S* **then**
        $nextStudyList.append(S)$;
    $StudyList \leftarrow nextStudyList$;
    $degree \leftarrow degree + 1$;
Store: *vaeModels* for all degree and all studies

one. All the obtained models are saved with their degrees. As shown in Figure 3.4, the maximum degree will be $\lceil \log_2 N \rceil$, resulting in $2N-1$ unique models.

The imputation of unseen data is based on multiple models that belong to different degrees (Eq. 3.4). A single model with most similar metabolites is used from the each degree. For each missing value y_{ij} in the dataset, the weighted averages of imputation results are calculated based on *degree* and *scale* parameters, and a final prediction for each missing cell y_{ij}^{final} is dynamically calculated. The *scale* parameter is a hyperparameter that determines the impact of the merged models' imputation values. It can take a minimum value of 1, meaning all models have equal influence. As the value increases, the impact of the merged model on the final result decreases. When it goes to infinity, only the model obtained with degree 0 is used. The models that have predictions of NaN, which do not contribute, are excluded from the calculation of the weighted average.

$$y_{ij}^{final} = \frac{\sum(y_{ij}^{pred} * scale^{-degree})}{\sum_{0}^{maxDegree} scale^{-degree}} \tag{3.4}$$

where *maxDegree* can change based on the trained models

Each study, integrated at the metabolite level, is divided into k-folds (with k=10 in this study). The minimum and maximum values from each fold's training data are obtained, and then the training and validation data are normalized and returned for training and evaluation purposes.

3.4 Imputation Model

Artificial neural networks (ANNs) are fundamental model architectures in deep learning that mimic the function of signal transmission in human neuron cells, which begins from the dendrites and ends at the axons of neuron cells in the human brain. The input layer takes the initial input data (image, text, audio, etc.) and passes it through one or many hidden layers, and finally to those connected to the output layer at the end (Figure 3.5).

In each neuron cell, an activation occurs during the forward pass of the network. Activation can be formulated as $a^i = W^i a^{i-1} + b^i$, where W^i represents the weights

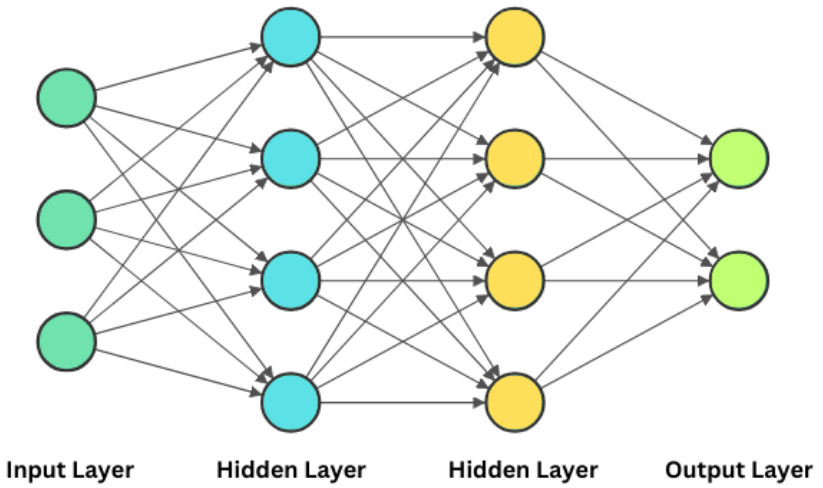

Figure 3.5 : Illustration of a fully connected neural network architecture.

and b^i represents biases, which are both randomly initialized for each neuron. The activation of previously passed cells a^{i-1} is used within the next cells as a^i. Then, the activation is transformed with a non-linear function such as *ReLU*, which is $z = max(0,a)$, to decide whether a neuron will be activated or not. The prediction of the output is compared with true values, and the loss for the model is calculated. During the back-propagation stage, the model optimizes the initialized weights and biases by minimizing that loss by calculating the partial derivatives of upstream gradients and local gradients, using them based on the chain rule. The derivatives can be tracked by a computational graph in recent deep learning frameworks. Partial derivatives can be defined with a simple setting as follows:

$$
\begin{aligned}
z^{(L)} = W^{(L)} a^{(L-1)} + b^{(L)} \quad a^{(L)} = ReLU(z^{(L)}) \quad Loss = (a^{(L)}, y)^2 \\
\frac{\partial Loss}{\partial W^{(L)}} = \frac{\partial z^{(L)}}{\partial W^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial Loss}{\partial a^{(L)}} = a^{(L-1)} ReLU'(z^L) 2(a^{(L)} - y) \\
\frac{\partial Loss}{\partial b^{(L)}} = \frac{\partial z^{(L)}}{\partial b^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial Loss}{\partial a^{(L)}} = ReLU'(z^L) 2(a^{(L)} - y) \\
\frac{\partial Loss}{\partial a^{(L-1)}} = \frac{\partial z^{(L)}}{\partial a^{(L-1)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial Loss}{\partial a^{(L)}} = W^{(L)} ReLU'(z^L) 2(a^{(L)} - y)
\end{aligned}
\tag{3.5}
$$

Optimization algorithms such as stochastic gradient descent (SGD) or its improved versions, like Adam [52] can be integrated into training procedures. For each layer in the neural network, parameters are updated with SGD in a mini-batch setup as in

equation 3.6, where new weights and biases at time t are updated based on parameters at time $t-1$ with the gradient of the loss function and pre-defined learning rate α.

$$\begin{aligned} \theta^{i(t)} &= \theta^{i(t-1)} - \alpha \nabla_{\theta} \mathscr{L}^{(t-1)} \\ W^{i(t)} &= W^{i(t-1)} - \alpha \nabla_{W^i} \mathscr{L}^{(t-1)} \\ b^{i(t)} &= b^{i(t-1)} - \alpha \nabla_{b^i} \mathscr{L}^{(t-1)} \end{aligned} \tag{3.6}$$

The improved optimization technique Adam combines the advantages of RMSProp [53] and SGD with momentum, improving the efficiency of finding the objective function's global minimum. The parameter update of Adam on parameter θ can be defined as shown in equation 3.7, where α is the learning rate, β_1 and β_2 are the decay rate parameters for the moments, and ε is the random number for preventing division by zero with default values $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$. The gradients of exponential moving averages and squared gradient moments are denoted as m_t and v_t.

$$\begin{aligned} \theta_t &= \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \\ \hat{m}_t = \frac{m_t}{1-\beta_1^t} \qquad & \hat{v}_t = \frac{v_t}{1-\beta_2^t} \\ m_t = \beta_1 m_{t-1} + (1-\beta_1) g_{\theta_t} \qquad & v_t = \beta_2 v_{t-1} + (1-\beta_2) g_{\theta_t}^2 \end{aligned} \tag{3.7}$$

An autoencoder (AE) is a dimensionality reduction-based neural network architecture that is specifically built to capture patterns in an unsupervised manner [54]. The primary objective is to rebuild the original input while simultaneously compressing the data. This compression allows the autoencoder to discover a more efficient and condensed version of the input. The architecture is made up of two neural networks: the encoder network passes the initial input into a bottleneck layer called latent space, which has a lower dimensionality, and the decoder network retrieves information from the latent space by recreating the original input. These primitive models are typically designed and optimized for data compression and denoising. The model architecture can be seen in Figure 3.6.

The encoder function $g(.)$ with parameter ϕ maps the input to $z = g_{\phi}(x)$, and the decoder function $f(.)$ with parameter θ reverts back to the original input as $x' =$

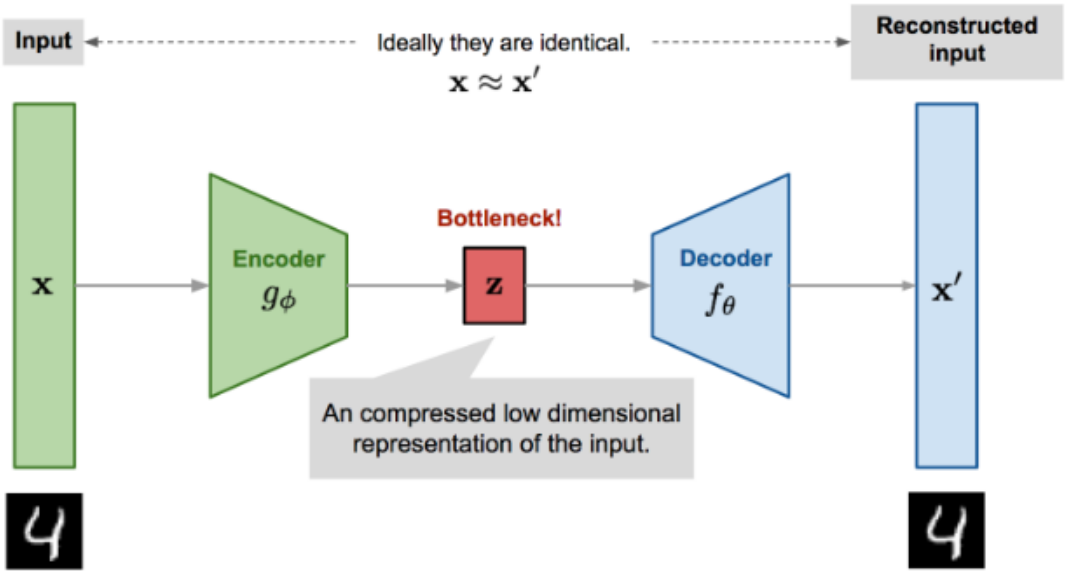

Figure 3.6 : Illustration of the autoencoder (AE) architecture [5].

$f_\theta(g_\phi(x))$. The model learns by minimizing the reconstruction error $\mathscr{L}_{AE}$. Different loss functions can be implemented between the initial input and the reconstructed version, such as cross-entropy or MSE loss, to learn (θ, ϕ) together (Eq. 3.8).

$$\mathscr{L}_{AE}(\theta, \phi) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)} - f_\theta(g_\phi(\mathbf{x}^{(i)})))^2 \tag{3.8}$$

The variational autoencoder (VAE) [46] is an extension of autoencoders designed for generative image sampling. In a variational autoencoder, the initial data is sampled from a parameterized prior distribution $p_\theta(\mathbf{z})$ instead of a fixed compressed space. Encoder-decoder blocks are trained together so that the model minimizes a joint loss, reconstruction error $\mathscr{L}_{MSE}$, and Kullback-Leibler divergence between the parametric posterior $q_\phi(\mathbf{z}|\mathbf{x})$ and the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$. The loss function is expressed as in Eq. 3.9.

$$\begin{gathered}\mathscr{L}_{VAE} = \mathscr{L}_{MSE} + \mathscr{L}_{KL} \\ \mathscr{L}_{VAE}(\theta, \phi) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)} - f_\theta(g_\phi(\mathbf{x}^{(i)})))^2 + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x}))\end{gathered} \tag{3.9}$$

A re-parameterization trick is applied while sampling the latent vector to make the stochastic sampling process deterministic and backpropagate the gradient, making the model trainable. The most common choice for parametric posteriors is a multivariate Gaussian distribution. In that case, VAE learns from the mean and variance of the

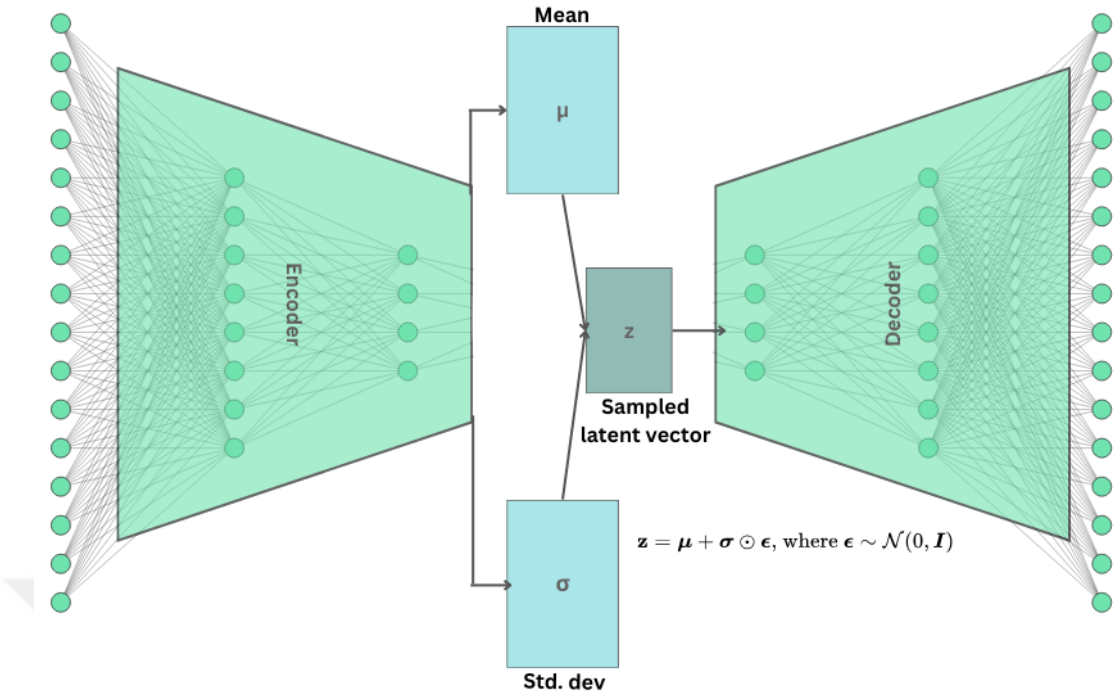

Figure 3.7 : Illustration of variational autoencoder (VAE) architecture.

distribution and a random noise $\boldsymbol{\varepsilon}$ sampled from the Gaussian. The re-parameterization trick is expressed in Eq. 3.10.

$$\begin{aligned} &\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\boldsymbol{I}) \\ &\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon}, \text{ where } \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \boldsymbol{I}) \quad \text{; Reparameterization trick.} \end{aligned} \tag{3.10}$$

Our baseline *MetaboliteVAE* model includes multiple fully connected linear layers, which are then followed by the *LeakyRELU* activation function with a negative slope of 0.2, which is formulated as $z = max(0,a) + 0.2 * min(0,a)$. The size of the encoder layers is reduced by dividing by two, while the decoder layers are expanded by multiplying by two. The latent dimension size is tuned within the given bounds to minimize training loss. A hyperparameter called kld-weight is set to 0.01 to adjust to the weight of the loss between $\mathscr{L}_{MSE}$ and $\mathscr{L}_{KL}$. Adam optimizer is used with a learning rate of 0.001. The model architectures are implemented using PyTorch 1.13.1 [55] with CUDA version 11.7 on Python 3.8.12. Tensor operations are performed on both CPU and GPU.

The personal computer was equipped with a GTX 1080Ti graphics card, which had 11 GB of video random access memory (VRAM), an Intel(R) Core(TM) i7-8700K central processing unit (CPU) running at a speed of 5.0 gigahertz, along with 16 gigabytes of random access memory (RAM).

4. EXPERIMENTAL RESULTS

In this chapter, we empirically evaluate the different aspects of the proposed imputation methods. We initially focus on identifying the optimal configuration of various parameters. Then, we assess the performance of the imputation methods.

4.1 Performance Metrics

As a performance metric, R^2 (R-squared) score, also known as the coefficient of determination, is utilized to calculate the robustness of simulated missingness. It is originally a regression metric, which serves as a statistical indicator of the degree to which the regression line accurately represents the observed data. The range of values that R^2 can take is in the interval $[-\infty, 1]$. A value of $R^2 = 1.0$ indicates a perfect fit, while a value of $R^2 < 0$ indicates that the model does not make sense for the given data. R^2 is computed as shown in Eq. 4.1. To compare the only imputed values across datasets, the $Sim - R^2$ score is calculated based on the masked values applied prior to the cross-validation operation. $Sim - R^2$ is computed as shown in Eq. 4.2.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum(y_i - y_{pred})^2}{\sum(y_i - y_{avg})^2} \tag{4.1}$$

$$Sim - R^2 = 1 - \frac{SS_{res(mask)}}{SS_{tot(mask)}} = 1 - \frac{\sum(y_{i(mask)} - y_{pred(mask)})^2}{\sum(y_{i(mask)} - y_{avg(mask)})^2} \tag{4.2}$$

4.2 Comparison of Different Initial Imputation Methods

Our database includes multiple metabolomics studies with various missingness rates. These datasets need to be imputed to have a ground truth value and evaluate our performance after reconstructing novel data. Therefore, we employed multiple imputation techniques to fill the original missing values in the datasets. For this experiment, we chose datasets that were originally complete and introduced artificial

random gaps in them, ranging from a minimum of 5% missingness to a maximum of 50% missingness. This experiment employs the following imputation approaches:

- *Mean Imputation (mean)*: Missing values in each metabolite column are filled with the mean of the non-empty values for that metabolite.
- *Median Imputation (median)*: Missing values in each metabolite column are filled with the median of the non-empty values for that metabolite.
- *Half-minimum (HM)*: Missing values in each metabolite column are filled with half of the non-missing minimum value for that metabolite.
- *Random Imputation (random)*: Missing values in each metabolite column are filled with a random floating number between the minimum and maximum of tha metabolite value.
- *K-nearest-neighbor (KNN) imputation* [11]: Missing values in each metabolite column are filled with the weighted average of a predetermined number of nearest non-missing variables for the specific metabolite. The origin of this imputation approach comes from microarray gene expression data. Neighbor distances were calculated using Euclidean distance, where closer neighbors of a missing metabolite will exert more effect than neighbors farther away. The Euclidean distance e between two points (x_1, y_1) and (x_2, y_2) can be calculated using equation 4.3. We applied the Python package scikit-learn *KNNImputer* for the imputation approach with a neighborhood size of 5.

$$e = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{4.3}$$

- *Random Forest (RF) imputation* [12]: Missing values in each metabolite column are filled with the prediction of a random forest model. More specifically, first, missing values are filled with mean/mode values of the corresponding metabolite. Then, for each metabolite containing missing values, an RF regressor was trained on other features, and missing values were predicted using the trained model. The iteration continued until a convergence value was reached. We applied the Python

package scikit-learn *IterativeImputer* with *RandomForestRegressor* estimator for the imputation approach.

An initial experiment was conducted using the dataset ST000284-CRC (Colorectal Cancer Detection) from Metabolomics Workbench, which has 224 patients, 118 females and 116 males, 103 metabolites, a total of 23,072 metabolite values, and three groups: *CRC*, *Healthy*, and *Polyp*. A total of 10 unique seeds were used at every missingness percentage rate, ranging from 5% to 50% with increments of 5%. The results are shown in Figure 4.1 and Figure 4.2. While *RF* demonstrated superior performance, it was comparable to *knn*. However, *RF*'s performance suffered from its complexity due to the training of numerous regressors for the imputation. In Figure 4.1, the overall distortion caused by imputed values from the optimal R^2 value of 1 can be interpreted as the robustness of the effectiveness method. R^2 values tend to decrease as the amount of missing data increases. $Sim - R^2$ values in Figure 4.2 oscillated between certain thresholds. Performance drawbacks of *random* and *hm* imputation are more drastic in that case.

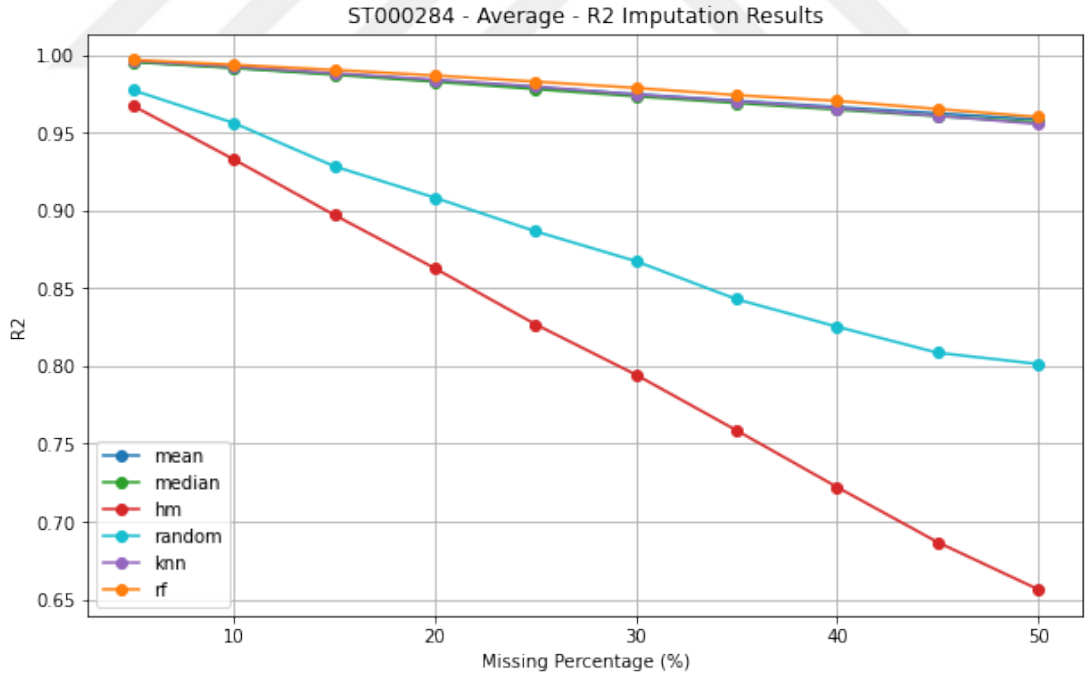

Figure 4.1 : R^2 scores for different imputation strategies on the ST000284-CRC (Colorectal Cancer Detection) dataset. 10 unique seeds were used, with varying missingness percentages. The performance drops as the amount of missing data increases.

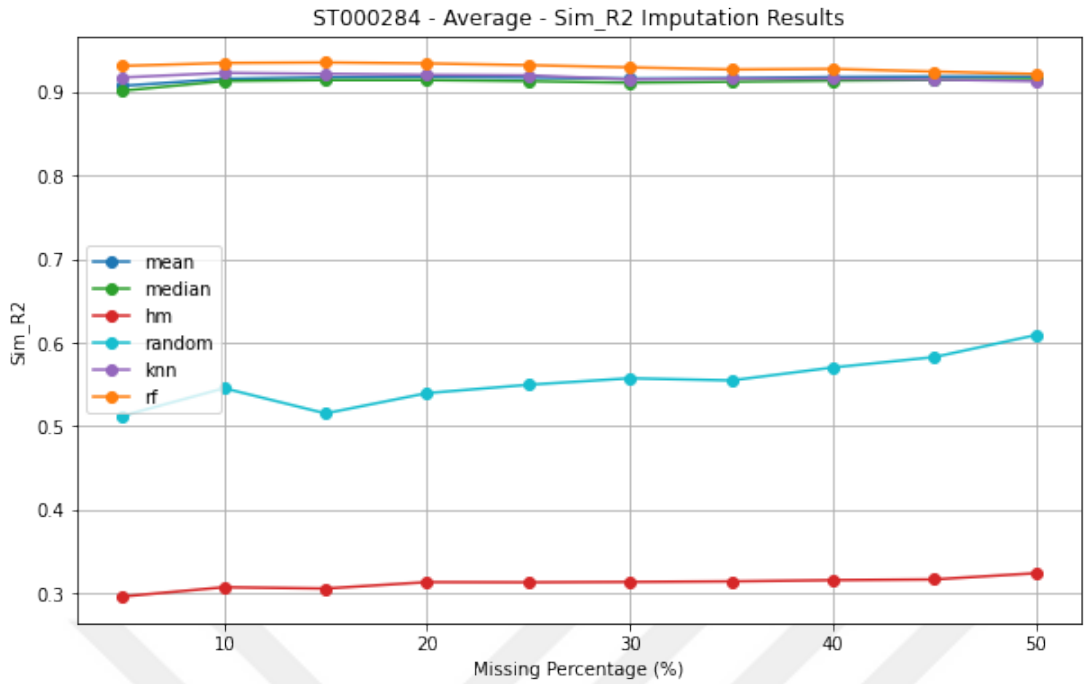

Figure 4.2 : Sim-R^2 scores of simulated missingness for different imputation strategies on the ST000284-CRC (Colorectal Cancer Detection) dataset. 10 unique seeds were used, with varying missing percentages. The random forest (*rf*) algorithm achieved the highest performance, with an average value of 0.9372.

To further evaluate the robustness of the imputation approaches, we selected representative non-empty datasets of different sizes (small, medium, and large) and repeated the same experiment. The results are shown in Figures 4.3 and 4.4. Table 4.1 shows the mean Sim-R^2 values over all missingness percentage rates in the studies. As seen in Figure 4.4 when the dataset size increases, nearly all algorithms' performance drops drastically for the Metabolomics Workbench. However, for MetaboLights, the performance patterns do not change significantly as the dataset size increases because the metabolite concentrations are pre-scaled small values which are close to each other. Total R^2 scores tend to decrease as the missingness percentage of the datasets increases because of the distortion of inaccurately filled-in missing values, as seen in Figure 4.3. The complexity drawback of *RF* was more evident in larger datasets due to the training of multiple regressors for the study. For this reason, *knn* imputation is chosen as the standard missing filling method in our proposed merging approaches for imputing the already existing missing values.

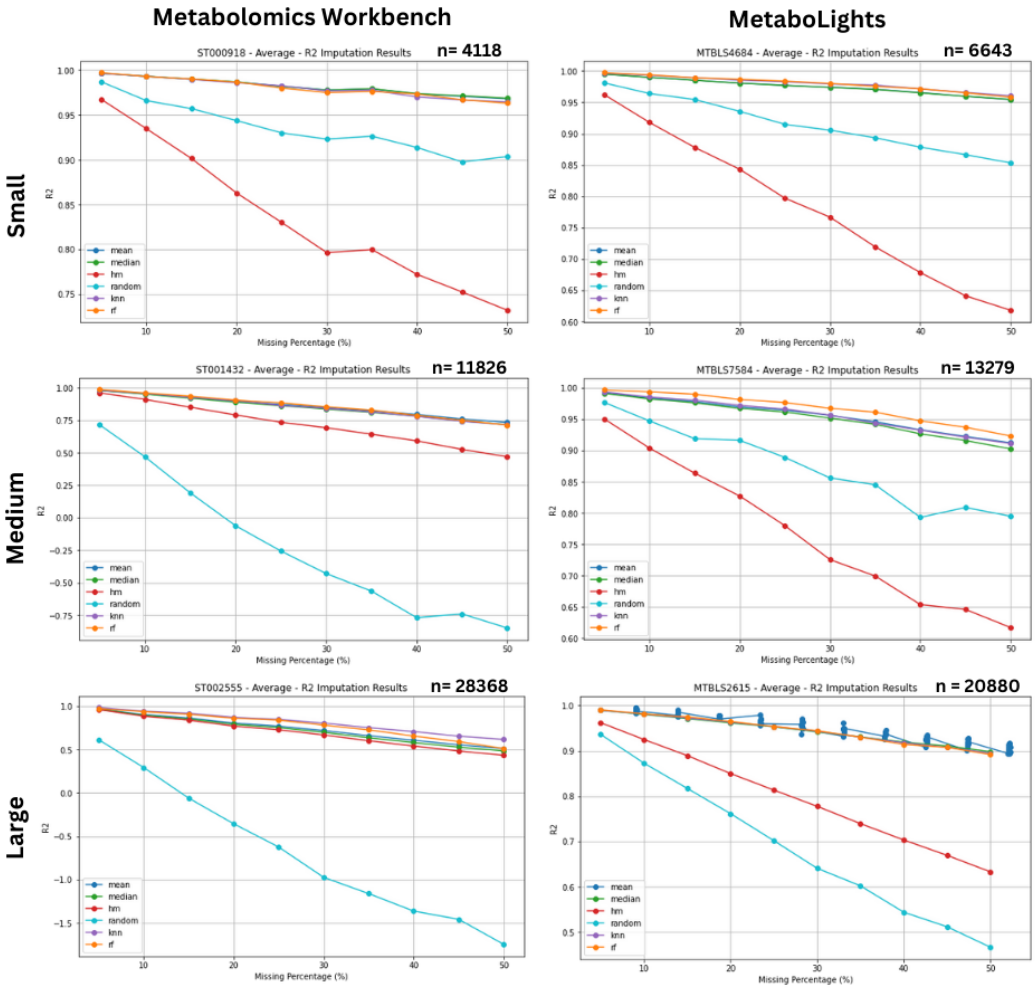

Figure 4.3 : R^2 scores of simulated missingness for different imputation strategies on six datasets of different sizes, denoted by n. 10 unique seeds were used, with varying missing percentages. The Metabolomics Workbench datasets are on the left, and the MetaboLights datasets are on the right.

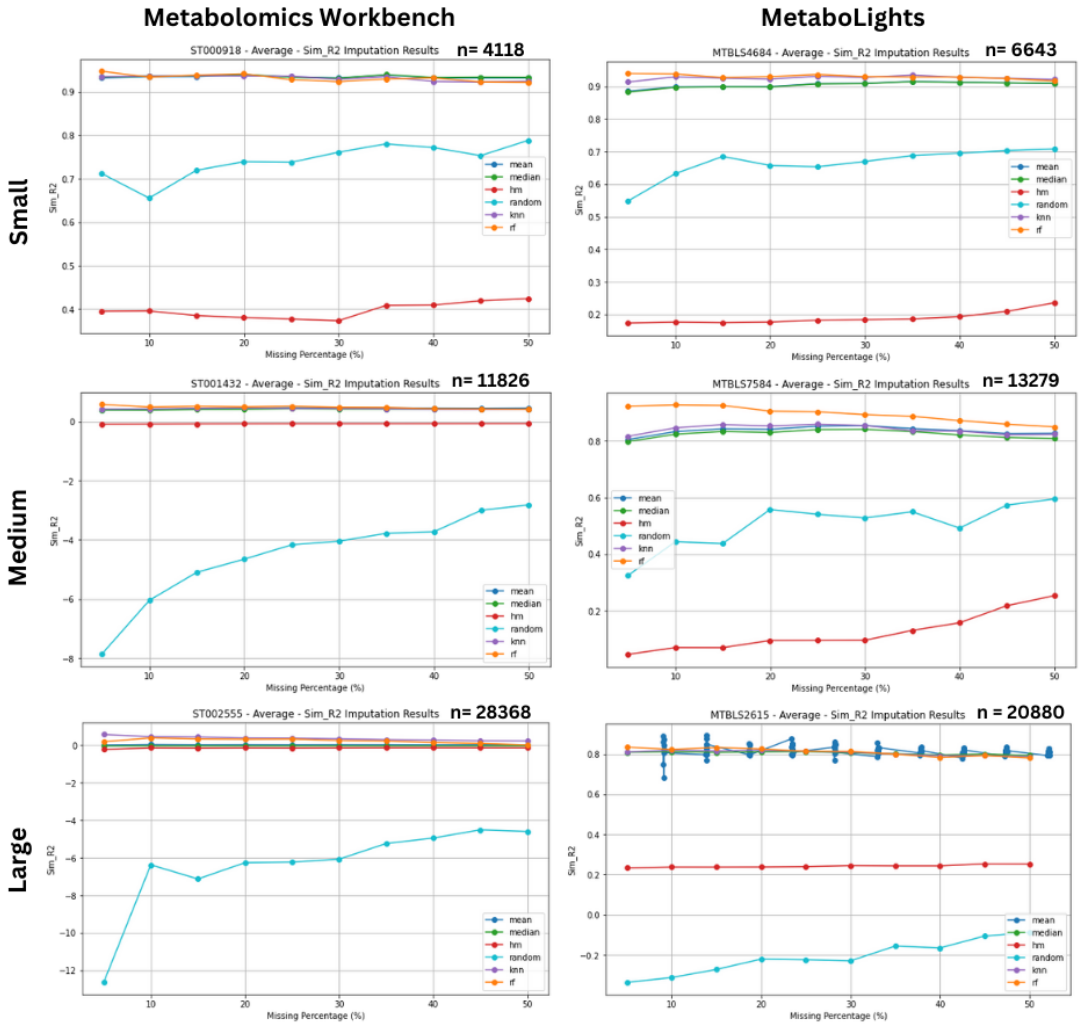

Figure 4.4 : Sim-R^2 scores of simulated missingness for different imputation strategies on six datasets of different sizes, denoted by n. 10 unique seeds were used, with varying missing percentages. The Metabolomics Workbench datasets are on the left, and the MetaboLights datasets are on the right.

Table 4.1 : Average Sim-R^2 scores for each imputation technique on datasets.

Dataset	Imputation Technique					
	Mean	Median	HM	Random	KNN	RF
ST000284 ($n = 23072$)	0.9255	0.9219	0.3827	0.5940	0.9266	**0.9372**
ST000918 ($n = 4118$)	0.9378	**0.9385**	0.4267	0.7659	0.9322	0.9330
ST001432 ($n = 11826$)	0.4793	0.4446	-0.0162	-3.6209	0.4517	**0.5127**
ST002555 ($n = 28368$)	0.1380	0.0782	-0.0140	-5.2625	**0.4313**	0.3366
MTBLS2615 ($n = 6643$)	**0.8319**	0.8169	0.3162	-0.0461	0.8204	0.8290
MTBLS7584 ($n = 13279$)	0.8351	0.8208	0.1189	0.5097	0.8443	**0.8934**
MTBLS4684 ($n = 20880$)	0.9159	0.9151	0.2541	0.6911	0.9315	**0.9323**
Average	0.7233	0.7051	0.2097	-0.9098	0.7625	**0.7677**

4.3 Comparison of Different Pretreatment Schemes

In order for the *MetaboliteVAE* model to learn effectively, the data needs to be normalized, scaled, or transformed after the imputation of initial missing values. Otherwise, the larger raw values in the dataset could hinder the model's ability to backpropagate and optimize the weights and biases of the layers present in the encoder and decoder parts of the VAE.

To this end, twelve different data pre-treatment methods (min-max normalization, centering, unit variance, Pareto, range, vast, level scaling, and log, Box-Cox, Yeo-Johnson transformation) (see Section 2.3) are compared with that of a baseline VAE model trained on the previously mentioned study ST000284. A random missingness rate between 10% and 20% to mimic the missingness in the real clinical datasets is considered, and 10-fold cross-validation is applied for evaluation. The baseline VAE model, which is called *MetaboliteVAE*, is discussed in Section 3.4. Due to the range of the data, not all pretreatment methods can create feasible models. The experiment is repeated on the six datasets mentioned above to compare the robustness of pretreatment methods. While the Yeo-Johnson transformation outperformed the other approaches, the drawback of increased complexity was more apparent due to the optimization of the λ parameter. For this reason, the log_{10} transformation was chosen as the standard pre-treatment method for our proposed merging approaches. The average Sim$-R^2$ results for 10-folds are shown in Table 4.2.

Table 4.2 : Average Sim-R^2 scores of 10-folds *MetaboliteVAE* reconstructions for each data pretreatment technique.

Method	Dataset						
	ST284	ST918	ST1432	ST2555	M2615	M7584	M4684
Min - Max	0.4379	0.1905	0.2117	0.3273	0.2770	0.1565	0.2706
Centering	-	-0.2927	0.1403	0.5771	-0.1854	0.0296	-0.0029
Standard	0.1318	0.1588	0.0440	0.6877	0.0496	-0.4090	0.3996
Pareto	-	-0.1513	0.0721	0.6856	0.0881	0.2856	0.2834
Range	0.1655	0.1270	0.0406	0.6619	0.0253	-0.4532	0.3627
Vast	0.1805	0.1697	0.0899	0.6669	0.0773	-0.5243	0.3254
Level	-	-0.0225	-16.3541	0.6279	-0.0094	-0.0385	0.4073
$\log_e$	0.8613	0.9184	0.5747	0.7491	-	0.8623	0.8916
$\log_2$	0.8576	0.9152	0.5637	**0.7603**	-	0.8611	0.8836
$\log_{10}$	**0.8854**	0.9257	0.6045	0.7566	-	**0.8627**	**0.8989**
Box-Cox	-	0.9011	0.6301	0.7243	-	0.8527	0.8769
Yeo-Johnson	0.7460	**0.9468**	**0.6524**	0.6172	**0.9568**	0.8511	0.8782

4.4 Comparison of Different Dataset Merging Approaches

We next evaluate our proposed dataset merging strategies for missing metabolite value imputations.

The first approach, *Union-based Merging*, aims to join all available datasets in public databases to create one universal large model. This approach represents the state-of-the-art in the metabolomics data imputation space [19]. However, under the *Union-based Merging* scheme, *MetaboliteVAE* fails to learn and optimize parameters. Hence, we conclude that the Union-based Merging strategy is not suitable for real-life, diverse datasets.

Based on this problem, to form a comparable baseline, we made an attempt to make union-based merging work by introducing an improved version of it. We call the improved version *Best Effort Union-based Merging*. It selects random seed datasets from our database as a starting point and joins them based on the Jaccard similarity score until the merged dataset does not lead to a trainable model (i.e., the loss value is not undefined or NaN). The number of most similar studies that are merged in this approach is a parameter, represented as n_{sim}. Different n_{sim} values are selected, such as 16, 12, 10, and 8, and we determine that 8 is the most optimal number to create trainable models under the best-effort union-based merging approach

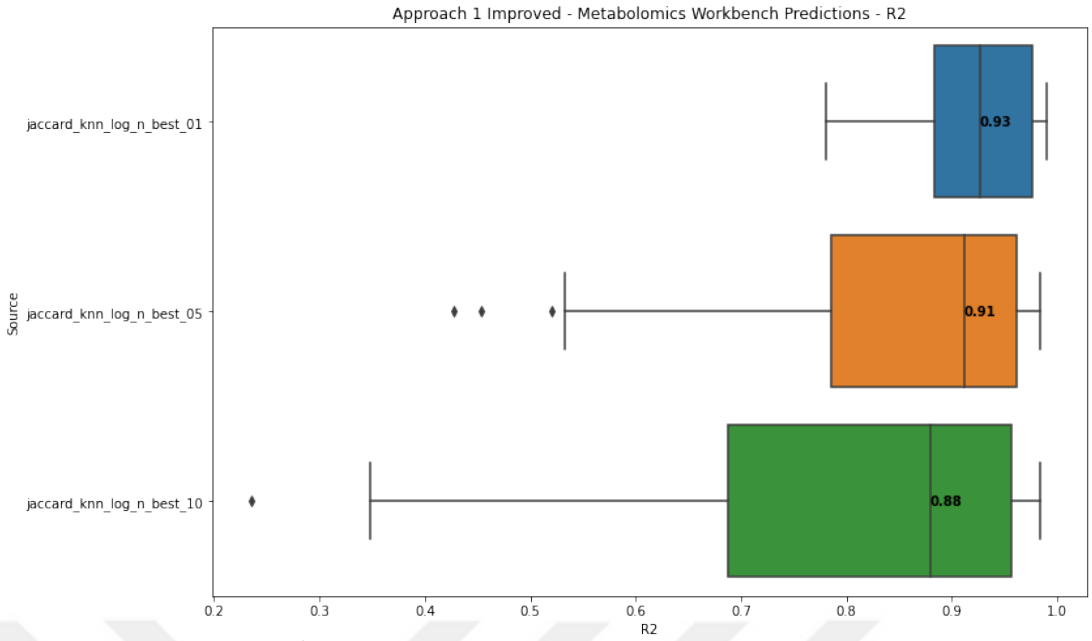

Figure 4.5 : Average R^2 scores of *Best-effort Union-based Merging* on Metabolomics Workbench studies.

before trained models lead to undefined loss values. A random selection of 10 seed datasets was made from the Metabolomics Workbench database, and each of them was combined with their 8 most similar datasets. Their models were trained using a 10-fold cross-validation setup. In cross-validation, datasets are first shuffled, then divided into 90% training and 10% test setup, repeated by the number of folds. Trained models were saved to be used in prediction. For 437 studies in the Metabolomics Workbench database, the test data that was reserved in the cross-validation split was used as unseen data. These test dataset splits are fully imputed with the *KNNImputer*. Then, artificial missing values are created at random missingness rates between 10%-20%, and predictions are made using different numbers of the most similar models (parametrized as n_{best}). We experiment with different n_{best} values of 1, 5, and 10. The results are shown in Figure 4.5 and Figure 4.6. For cases where we have predictions by multiple models (i.e., $n_{best} > 1$), weighted averages based on Jaccard similarity were used. Because a limited amount of data was used during model training, the models under this merging strategy were unable to provide predictions for all of the unseen Metabolomics Workbench data. More specifically, the best-effort union-based approach was able to come up with missing value predictions for 63 studies only.

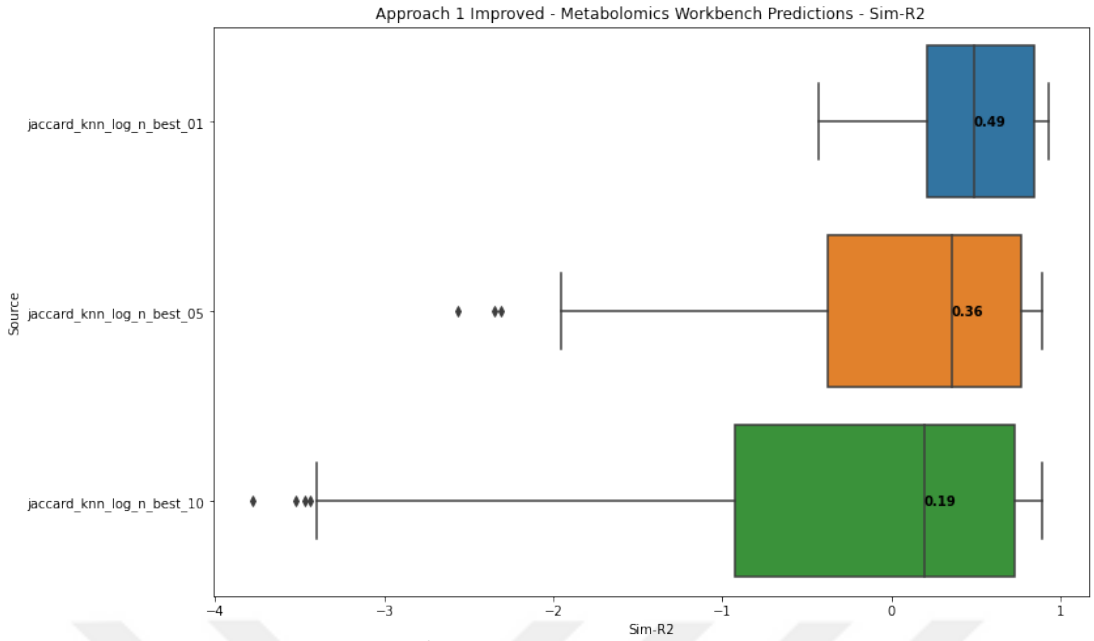

Figure 4.6 : Average *Sim-*R^2 scores of *Best-effort Union-based Merging* on Metabolomics Workbench studies.

For the second approach, *Iterative Similarity-based Merging*, for each of 437 studies in our Metabolomics Workbench database, an optimal merge set is created with the proposed algorithm. After duplicates are eliminated, we obtain 264 unique Jaccard-based merge sets. For each merge set, studies inside the set are artificially merged, and a model was trained using a 10-fold cross-validation setup on the merged dataset as explained above.

After saving our models in a repository, we tested them with the test data from each study that we split with cross-validation. Finally, the artificially created missing values are imputed with our newly proposed *IterativeSimilarityImputer*. This imputer was created to fit the dataset with the best models possible, which are the models that have the highest Jaccard similarity score among the datasets. The test data of 437 studies are imputed with *IterativeSimilarityImputer* with n_{best} as 1, 5, and 10. The results are presented in Figure 4.7 and Figure 4.8 .

The third dataset merging approach, *Model-guided Agglomerative Merging*, merges datasets two by two until a single model is produced. When we look at the training losses of each individual model up until the final trained joint models, we observe that the loss value of the joint model becomes NaN (not a number) after joining 64 models,

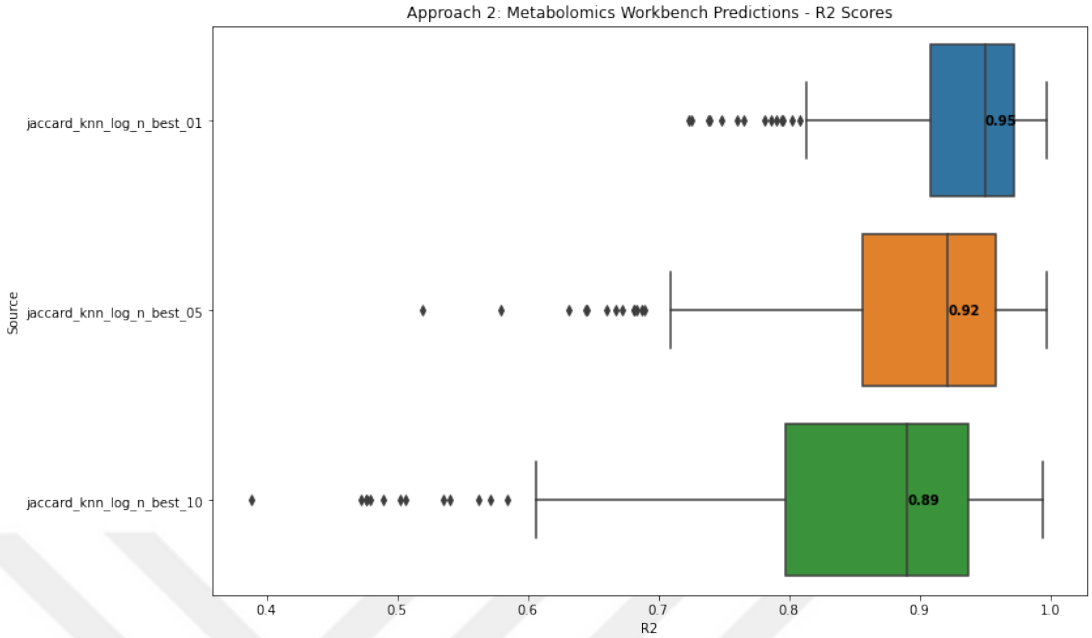

Figure 4.7 : Average R^2 scores of *Iterative Similarity-based Merging* on Metabolomics Workbench studies.

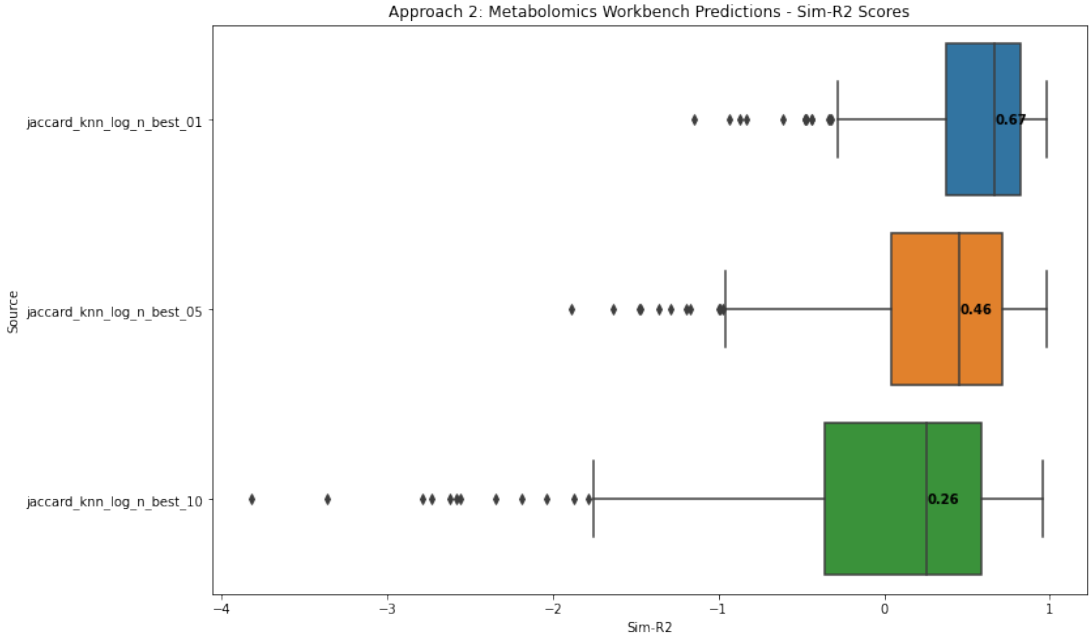

Figure 4.8 : Average *Sim-*R^2 scores of *Iterative Similarity-based Merging* on Metabolomics Workbench studies.

similar to the scenario of *Union-based Merging*. As a result, the approach's original prediction scheme as explained in the methods section was slightly revised to tackle this issue. That is, instead of going all the way through the root node, the joining process is early-stopped just before loss values become undefined, i.e., NaN. With this revision, the performance of the model-guided agglomerative merging on unseen Metabalomics Workbench data is presented in Figure 4.9. Since the predictions are the weighted averages of multiple models, the total R^2 was a bit lower than single model predictions. Nevertheless, the most robust way to impute missing values among all the presented approaches is this approach which has an average *Sim*-R^2 of 0.72. Finally, to compare different approaches within the same chart, we present the performance of the best configuration for each studied approach in Figure 4.10.

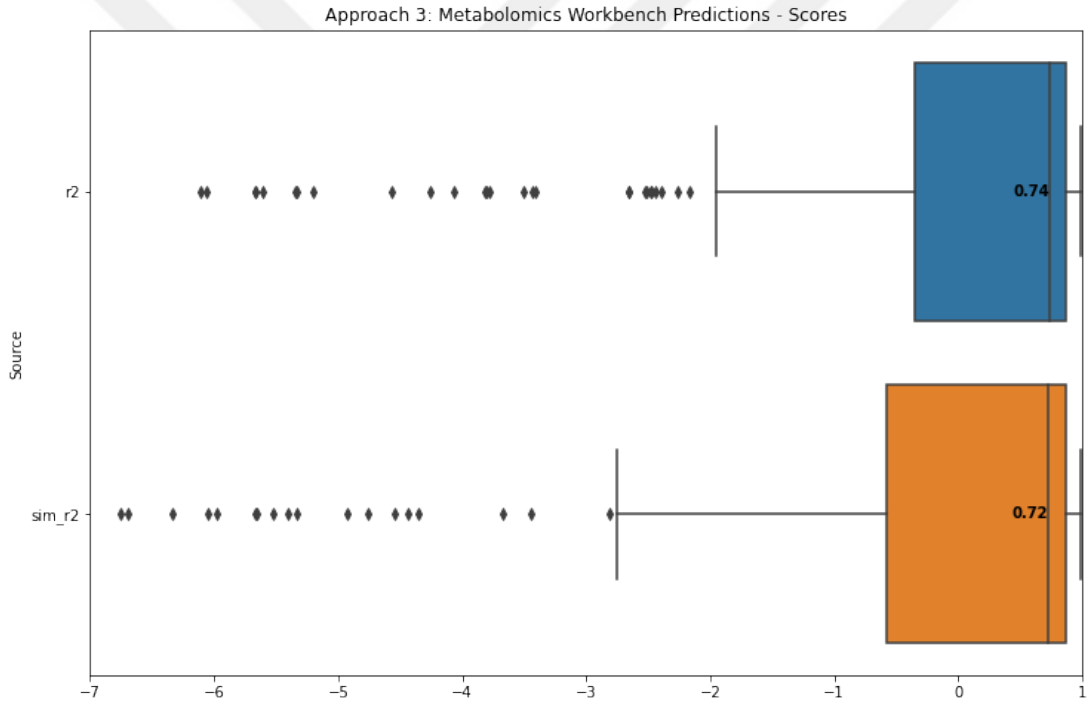

Figure 4.9 : Average scores of *Model-guided Agglomerative Merging* on Metabolomics Workbench studies.

4.5 Comparison of Different Similarity Metrics

The Jaccard similarity score, which we employed in our merging methods, solely assessed the similarity between datasets, or between models and datasets based on the metabolite names. Since datasets may have been generated with different measurement techniques and devices, metabolite concentrations may have different units. To

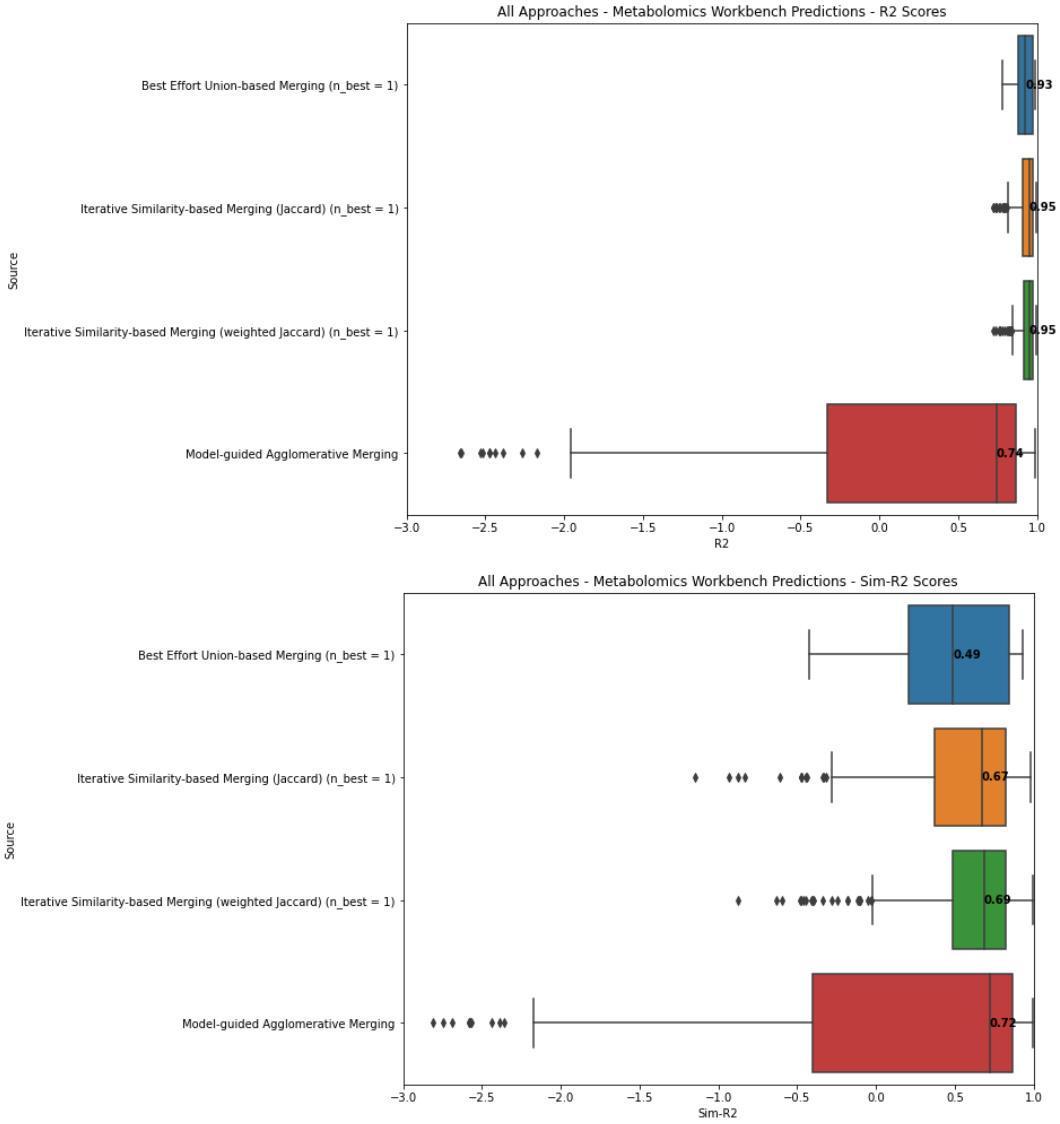

Figure 4.10 : Performance scores of the baseline union-based merging and the proposed two approaches with best configurations on unseen Metabolomics Workbench data. R^2 at the top and *Sim*-R^2 at the bottom.

consider the range of each individual metabolite and merge datasets in a more effective way, a weighted Jaccard similarity scoring scheme was employed, as explained in Section 3.3.2. For Iterative Similarity-based Merging, each of the 437 studies in our Metabolomics Workbench database was used to create new optimal merge sets with a weighted Jaccard score. After duplicates are eliminated, we obtain 363 unique weighted Jaccard-based merge sets. The same cross-validation setup was used to train all merge sets. Also, the weighted score was used for predictions. The average 10-fold prediction results of unseen Metabolomics Workbench data are presented in Figure 4.11 with R^2 scores at the top and *Sim-*R^2 scores at the bottom. The increase in performance for R^2 scores was negligible. However, upon examining the only missing values, namely the *Sim-*R^2 scores of n_{best} with 5 and 10 models, it is evident that the performance has improved. As a result, the weighted Jaccard score enabled us to build more stable and concise models for our model pool, increasing the multi-model prediction power.

4.6 Comparison of Training and Prediction Times

In this thesis, we presented various dataset merging approaches to efficiently create robust models that accurately impute missing values in metabolomics research. The training and prediction times were recorded to evaluate the performance. The training durations for one fold of proposed merging approaches on Metabolomics Workbench data are presented in Table 4.3 and Figure 4.12. *Union-based Merging* could not learn and capture any patterns when faced with NaN losses during its initial training phase. The time requirements for training *Model-guided Agglomerative Merging* were excessive since the size of the models after the first layer expanded.

Table 4.3 : One-fold training times for different approaches on the Metabolomics Workbench studies.

Approach	Training Time
Union-based Merging	No training
Best Effort Union-based Merging (Jaccard)	28.5 min
Iterative Similarity-based Merging (Jaccard)	43.7 min
Iterative Similarity-based Merging (weighted Jaccard)	48.2 min
Model-guided Agglomerative Merging (until 128 joint)	517.5 min

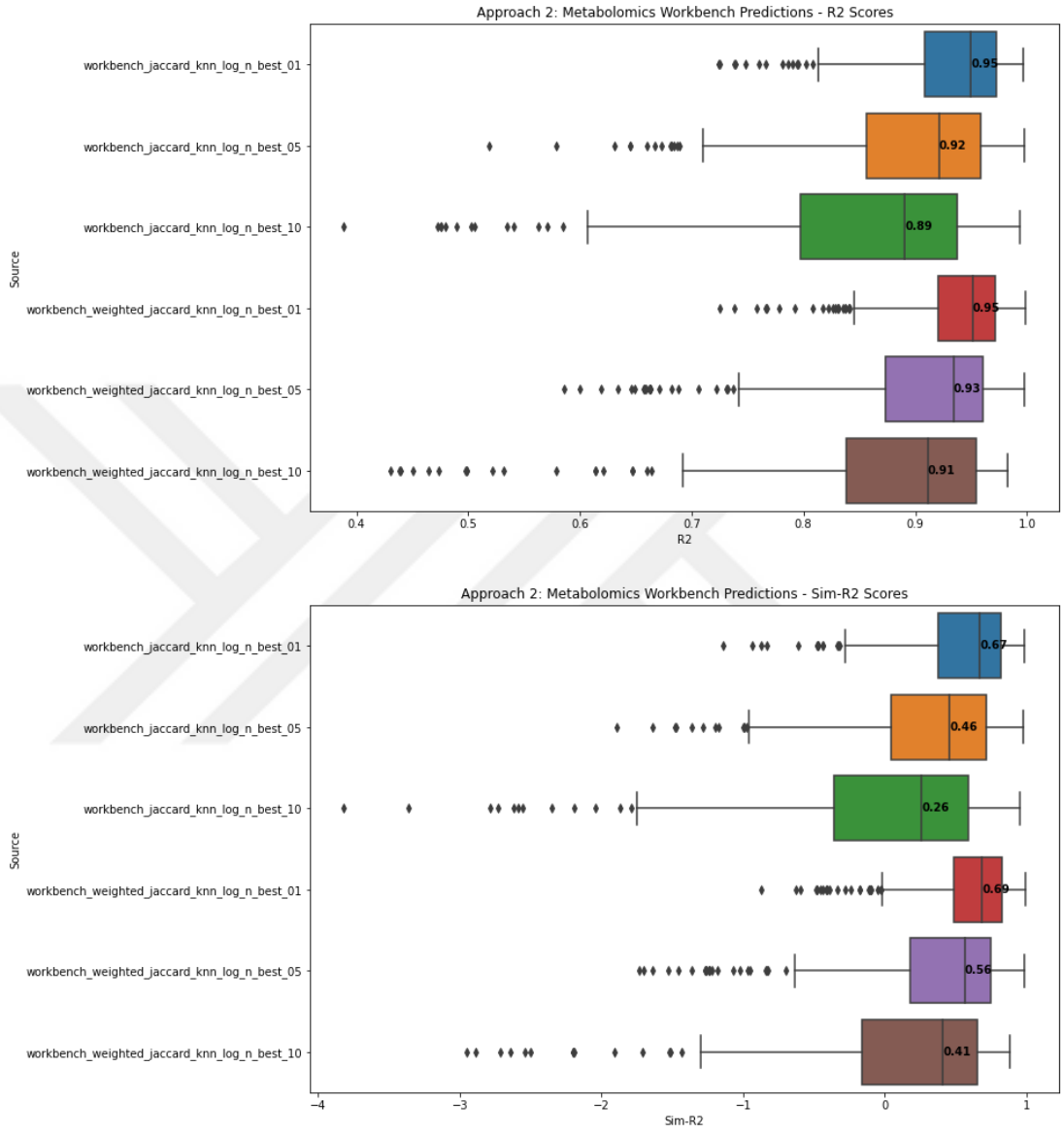

Figure 4.11 : Average test scores of *Iterative Similarity-based Merging* with different similarity metrics on Metabolomic Workbench studies. R^2 at the top and, *Sim-*R^2 at the bottom.

Table 4.4 : Prediction times of one missing metabolite for different approaches trained on Metabolomics Workbench.

Approach	Prediction Time
Union-based Merging	No prediction
Best Effort Union-based Merging (Jaccard) ($n_{best} = 1$)	3.18×10^{-7} sec
Best Effort Union-based Merging (Jaccard) ($n_{best} = 5$)	1.28×10^{-6} sec
Best Effort Union-based Merging (Jaccard) ($n_{best} = 10$)	3.18×10^{-6} sec
Iterative Similarity-based Merging (Jaccard) ($n_{best} = 1$)	2.23×10^{-7} sec
Iterative Similarity-based Merging (Jaccard) ($n_{best} = 5$)	1.21×10^{-6} sec
Iterative Similarity-based Merging (Jaccard) ($n_{best} = 10$)	2.53×10^{-6} sec
Iterative Similarity-based Merging (weighted Jaccard) ($n_{best} = 1$)	2.00×10^{-7} sec
Iterative Similarity-based Merging (weighted Jaccard) ($n_{best} = 5$)	1.16×10^{-6} sec
Iterative Similarity-based Merging (weighted Jaccard) ($n_{best} = 10$)	2.53×10^{-6} sec
Model-guided Agglomerative Merging (until 128 joint)	7.85×10^{-7} sec

The prediction times of different approaches were measured based on the average single metabolite prediction time. The results are presented in Table 4.4 and Figure 4.13. Although configurations with the first similar model, $n_{best} = 1$, provide a fast prediction, their imputation performance suffers when compared to *Model-guided Agglomerative Merging*. For bigger datasets, *Iterative Similarity-based Merging* with $n_{best} = 1$ configurations are fast and robust ways to impute missing values, by compromising a small amount of prediction power. However for small and medium sized datasets *Model-guided Agglomerative Merging* is a promising approach.

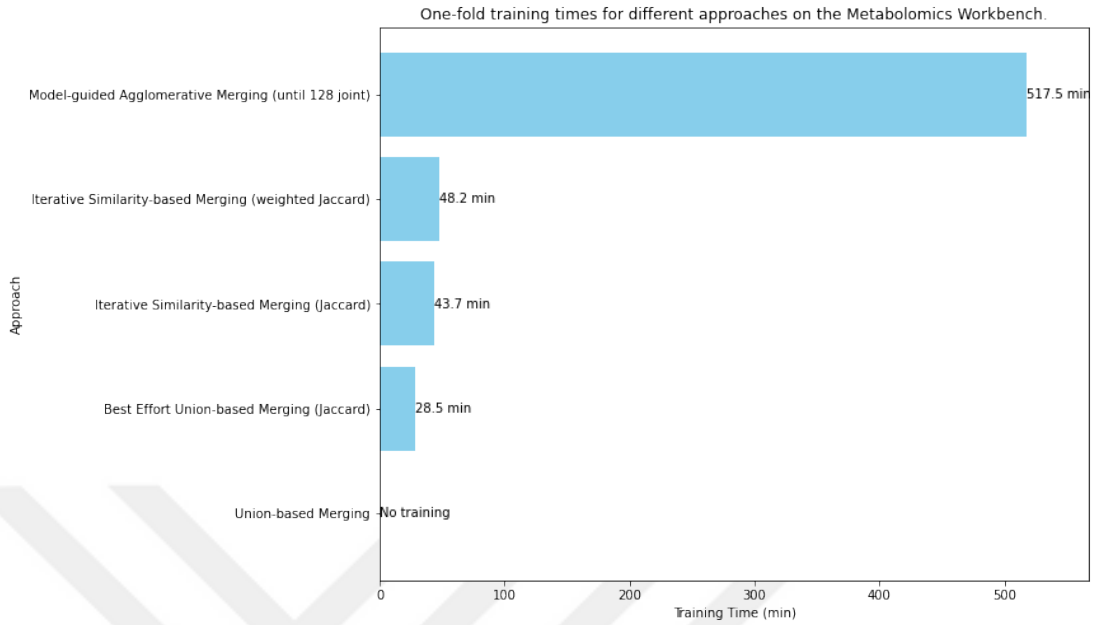

Figure 4.12 : One-fold training times for different approaches on the Metabolomics Workbench studies.

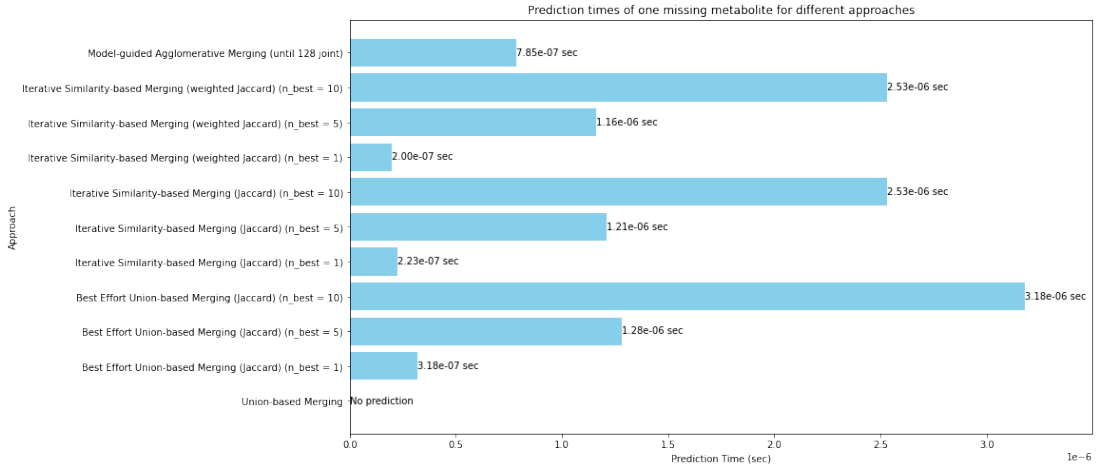

Figure 4.13 : Prediction times of one missing metabolite for different approaches trained on Metabolomics Workbench.

5. CONCLUSION AND FUTURE WORK

In this thesis, we aim to build a robust deep-learning framework based on generative models to accurately impute missing values in metabolomics datasets. To this end, we propose three dataset merging approaches to take advantage of diverse datasets in public data repositories.

We collected and parsed a large number of publicly available metabolomics datasets from two different major metabolomics data sources, namely, Metabolomics Workbench and MetaboLights. A metabolite name mapping procedure was conducted using the RefMet database to unify the metabolites in different datasets. Dealing with varying data scales was one of the primary challenges, as studies were conducted in various laboratories worldwide and measured using different devices. A generalized pre-treatment pipeline was developed to minimize inter-dataset differences. The original study data may include missing values due to various external reasons, such as the device's detection limit or a misalignment of the peak signals. *KNNImputer* was employed to effectively fill the original missing values to create baseline non-empty datasets to train our models. During the data pre-treatment stage, log transformation with different bases and Yeo-Johnson transformation helped the *MetaboliteVAE* model capture patterns in the data effectively.

To the best of our knowledge, there is no data source- and dataset-independent missing value imputation method in the literature for metabolomics data imputation. This study serves as a pioneer in implementing different merging approaches for data pooling in the metabolomics domain. Some publications use uniform, comprehensive datasets, such as Gomari et al. [19]. We expanded our research based on their starting point and faced many challenges. This was represented by the *Union-based Merging* in our work. Models created with *Union-based Merging* approach failed to learn due to the diversity of the metabolite set in the training data. The improved version of the first approach, *Iterative Similarity-based Merging* could create models with high predictive

power that cover a limited number of metabolites. *Model-guided Agglomerative Merging* increases the robustness of the models by predicting at each merging set. The cross-validation on a smaller subset of the datasets showed promising results.

In future research, it is possible to categorize the studies in the database according to their units, allowing for the creation of smaller sub-databases. This would facilitate the merging of datasets and enhance the development of more reliable models. Although some of the pre-treatment methods are robust to outliers, others ignore their occurrence. A further outline elimination can be done on each metabolite after creating single dataset files for each study.

As another future work, this work may be extended with more data sources. In addition, a graph convolutional model architecture may be developed to take advantage of the metabolic network connections. One may use a biological network, such as Recon3D, to incorporate the relationships among metabolites as additional features of a VAE since the metabolites are connected with reactions and pathways in the human biological system. With this modification, the generic generative model architecture can be transformed into a graph-based convolutional architecture, known as GCNN-VAE.

REFERENCES

[1] **Proteomics, C.** (2024). *Q Exactive Plus Hybrid Quadrupole Orbitrap Mass Spectrometer*, https://www.creative-proteomics.com.

[2] **ten Haaft, R.** (2023). *Bronkhorst*, https://www.bronkhorst.com.

[3] **Bruker** (2024). *Avance NMR Spectrometer | Bruker*, https://www.bruker.com/en/products-and-solutions/mr/nmr/avance-nmr-spectrometer.html.

[4] **Rankin, N.**, **Preiss, D.**, **Welsh, P.**, **Burgess, K.**, **Nelson, S.**, **Lawlor, D. and Sattar, N.** (2014). The emergence of proton nuclear magnetic resonance metabolomics in the cardiovascular arena as viewed from a clinical perspective, *Atherosclerosis*, *237*, 287–300.

[5] **Weng, L.** (2018). *From Autoencoder to Beta-VAE*, https://lilianweng.github.io/posts/2018-08-12-vae/.

[6] **Rinschen, M.M.**, **Ivanisevic, J.**, **Giera, M. and Siuzdak, G.** (2019). Identification of bioactive metabolites using activity metabolomics, *Nature Reviews Molecular Cell Biology*, *20*(6), 353–367, https://doi.org/10.1038/s41580-019-0108-4.

[7] **Fuhrer, T. and Zamboni, N.** (2015). High-throughput discovery metabolomics, *Current Opinion in Biotechnology*, *31*, 73–78, https://www.sciencedirect.com/science/article/pii/S0958166914001487, analytical Biotechnology.

[8] **Sun, J. and Xia, Y.** (2024). Pretreating and normalizing metabolomics data for statistical analysis, *Genes & Diseases*, *11*(3), 100979, https://www.sciencedirect.com/science/article/pii/S2352304223002246.

[9] **Wei, R.**, **Wang, J.**, **Su, M.**, **Jia, E.**, **Chen, S.**, **Chen, T. and Ni, Y.** (2018). Missing Value Imputation Approach for Mass Spectrometry-based Metabolomics Data, *Scientific Reports*, *8*(1), 663, https://doi.org/10.1038/s41598-017-19120-0.

[10] **Do, K.T.**, **Wahl, S.**, **Raffler, J.**, **Molnos, S.**, **Laimighofer, M.**, **Adamski, J.**, **Suhre, K.**, **Strauch, K.**, **Peters, A.**, **Gieger, C.**, **Langenberg, C.**, **Stewart, I.D.**, **Theis, F.J.**, **Grallert, H.**, **Kastenmüller, G. and Krumsiek, J.** (2018). Characterization of missing values in untargeted

MS-based metabolomics data and evaluation of missing data handling strategies, *Metabolomics*, *14*(10), 128, https://doi.org/10.1007/s11306-018-1420-2.

[11] **Troyanskaya, O.**, **Cantor, M.**, **Sherlock, G.**, **Brown, P.**, **Hastie, T.**, **Tibshirani, R.**, **Botstein, D. and Altman, R.B.** (2001). Missing value estimation methods for DNA microarrays , *Bioinformatics*, *17*(6), 520–525, https://doi.org/10.1093/bioinformatics/17.6.520, https://academic.oup.com/bioinformatics/article-pdf/17/6/520/48837104/bioinformatics_17_6_520.pdf.

[12] **Stekhoven, D.J. and Bühlmann, P.** (2011). MissForest—non-parametric missing value imputation for mixed-type data, *Bioinformatics*, *28*(1), 112–118, https://doi.org/10.1093/bioinformatics/btr597, https://academic.oup.com/bioinformatics/article-pdf/28/1/112/50568519/bioinformatics_28_1_112.pdf.

[13] **van Buuren, S. and Oudshoorn, K.C.** (1999). *Flexible Multivariate Imputation by MICE*, TNO Prevention and Health.

[14] **Zheng, S. and Charoenphakdee, N.** (2023). *Diffusion models for missing value imputation in tabular data*, 2210.17128.

[15] **Gondara, L. and Wang, K.** (2018). *MIDA: Multiple Imputation using Denoising Autoencoders*, 1705.02737.

[16] **Yoon, J.**, **Jordon, J. and van der Schaar, M.** (2018). *GAIN: Missing Data Imputation using Generative Adversarial Nets*, 1806.02920.

[17] **Dekermanjian, J.P.**, **Shaddox, E.**, **Nandy, D.**, **Ghosh, D. and Kechris, K.** (2022). Mechanism-aware imputation: a two-step approach in handling missing values in metabolomics, *BMC Bioinformatics*, *23*(1), 179, https://doi.org/10.1186/s12859-022-04659-1.

[18] **Jin, Z.**, **Kang, J. and Yu, T.** (2017). Missing value imputation for LC-MS metabolomics data by incorporating metabolic network and adduct ion relations, *Bioinformatics*, *34*(9), 1555–1561, https://doi.org/10.1093/bioinformatics/btx816, https://academic.oup.com/bioinformatics/article-pdf/34/9/1555/48915313/bioinformatics_34_9_1555.pdf.

[19] **Gomari, D.P.**, **Schweickart, A.**, **Cerchietti, L.**, **Paietta, E.**, **Fernandez, H.**, **Al-Amin, H.**, **Suhre, K. and Krumsiek, J.** (2022). Variational autoencoders learn transferrable representations of metabolomics data, *Communications Biology*, *5*(1), 645, https://doi.org/10.1038/s42003-022-03579-3.

[20] **Zhao, C.**, **Su, K.J.**, **Wu, C.**, **Cao, X.**, **Sha, Q.**, **Li, W.**, **Luo, Z.**, **Qin, T.**, **Qiu, C.**, **Zhao, L.J.**, **Liu, A.**, **Jiang, L.**, **Zhang, X.**, **Shen, H.**, **Zhou, W. and**

Deng, H.W. (2023). Multi-view variational autoencoder for missing value imputation in untargeted metabolomics, *ArXiv*.

[21] **Sud, M.**, **Fahy, E.**, **Cotter, D.**, **Azam, K.**, **Vadivelu, I.**, **Burant, C.**, **Edison, A.**, **Fiehn, O.**, **Higashi, R.**, **Nair, K.S.**, **Sumner, S. and Subramaniam, S.** (2015). Metabolomics Workbench: An international repository for metabolomics data and metadata, metabolite standards, protocols, tutorials and training, and analysis tools, *Nucleic Acids Res*, *44*(D1), D463–70.

[22] **Kale, N.S.**, **Haug, K.**, **Conesa, P.**, **Jayseelan, K.**, **Moreno, P.**, **Rocca-Serra, P.**, **Nainala, V.C.**, **Spicer, R.A.**, **Williams, M.**, **Li, X.**, **Salek, R.M.**, **Griffin, J.L. and Steinbeck, C.** (2016). MetaboLights: An Open-Access Database Repository for Metabolomics Data, *Curr Protoc Bioinformatics*, *53*, 14.13.1–14.13.18.

[23] **Manzoni, C.**, **Kia, D.A.**, **Vandrovcova, J.**, **Hardy, J.**, **Wood, N.W.**, **Lewis, P.A. and Ferrari, R.** (2016). Genome, transcriptome and proteome: the rise of omics data and their integration in biomedical sciences, *Briefings in Bioinformatics*, *19*(2), 286–302, https://doi.org/10.1093/bib/bbw114, https://academic.oup.com/bib/article-pdf/19/2/286/25524262/bbw114.pdf.

[24] **Suhre, K.** (2014). Metabolic profiling in diabetes, *J Endocrinol*, *221*(3), R75–85.

[25] **Kaddurah-Daouk, R.**, **Kristal, B.S. and Weinshilboum, R.M.** (2008). Metabolomics: A Global Biochemical Approach to Drug Response and Disease, *Annual Review of Pharmacology and Toxicology*, *48*(Volume 48, 2008), 653–683, https://www.annualreviews.org/content/journals/10.1146/annurev.pharmtox.48.113006.094715.

[26] **Costa dos Santos, G.**, **Renovato-Martins, M. and de Brito, N.M.** (2021). The remodel of the "central dogma": a metabolomics interaction perspective, *Metabolomics*, *17*(5), 48, https://doi.org/10.1007/s11306-021-01800-8.

[27] **Gonzalez-Covarrubias, V.**, **Martínez-Martínez, E. and del Bosque-Plata, L.** (2022). The Potential of Metabolomics in Biomedical Applications, *Metabolites*, *12*(2), https://www.mdpi.com/2218-1989/12/2/194.

[28] **Alonso, A.**, **Marsal, S. and Julià, A.** (2015). Analytical Methods in Untargeted Metabolomics: State of the Art in 2015, *Frontiers in Bioengineering and Biotechnology*, *3*, https://www.frontiersin.org/articles/10.3389/fbioe.2015.00023.

[29] **Bothwell, J.H.F. and Griffin, J.L.** (2011). An introduction to biological nuclear magnetic resonance spectroscopy, *Biological Reviews*, *86*(2), 493–510, https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-185X.2010.00157.x.

[30] **Reo, N.V.** (2002). NMR-BASED METABOLOMICS, *Drug and Chemical Toxicology*, *25*(4), 375–382, https://doi.org/10.1081/DCT-120014789, pMID: 12378948, https://doi.org/10.1081/DCT-120014789.

[31] **Patti, G.J.**, **Yanes, O. and Siuzdak, G.** (2012). Metabolomics: the apogee of the omics trilogy, *Nature Reviews Molecular Cell Biology*, *13*(4), 263–269, https://doi.org/10.1038/nrm3314.

[32] **van den Berg, R.A.**, **Hoefsloot, H.C.**, **Westerhuis, J.A.**, **Smilde, A.K. and van der Werf, M.J.** (2006). Centering, scaling, and transformations: improving the biological information content of metabolomics data, *BMC Genomics*, *7*(1), 142, https://doi.org/10.1186/1471-2164-7-142.

[33] **Mahalanobis, P.** (2018). Reprint of: Mahalanobis, P.C. (1936) "On the Generalised Distance in Statistics.", *Sankhya A*, *80*(1), 1–7, https://doi.org/10.1007/s13171-019-00164-5.

[34] **Riu, J. and Bro, R.** (2003). Jack-Knife Technique for Outlier Detection and Estimation of Standard Errors in PARAFAC models, *Chemometrics and Intelligent Laboratory Systems*, *65*, 35–49.

[35] **Xu, H.**, **Zhang, L.**, **Li, P. and Zhu, F.** (2022). Outlier detection algorithm based on k-nearest neighbors-local outlier factor, *Journal of Algorithms & Computational Technology*, *16*, 17483026221078111, https://doi.org/10.1177/17483026221078111, https://doi.org/10.1177/17483026221078111.

[36] **Dieterle, F.**, **Ross, A.**, **Schlotterbeck, G. and Senn, H.** (2006). Probabilistic Quotient Normalization as Robust Method to Account for Dilution of Complex Biological Mixtures. Application in 1H NMR Metabonomics, *Analytical Chemistry*, *78*(13), 4281–4290, https://doi.org/10.1021/ac051632c, pMID: 16808434.

[37] **Bolstad, B.M.**, **Irizarry, R.A.**, **Astrand, M. and Speed, T.P.** (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias, *Bioinformatics*, *19*(2), 185–193, https://doi.org/10.1093/bioinformatics/19.2.185.

[38] **Bro, R. and Smilde, A.K.** (2003). Centering and scaling in component analysis, *Journal of Chemometrics*, *17*(1), 16–33, https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/cem.773, https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/pdf/10.1002/cem.773.

[39] **Smilde, A.K.**, **van der Werf, M.J.**, **Bijlsma, S.**, **van der Werff-van der Vat, B.J.C. and Jellema, R.H.** (2005). Fusion of Mass Spectrometry-Based

Metabolomics Data, *Analytical Chemistry*, *77*(20), 6729–6736, https://doi.org/10.1021/ac051080y.

[40] **Jackson, E.A.**, **Rabbette, P.S.**, **Dezateux, C.**, **Hatch, D.J. and Stocks, J.** (1991). The effect of triclofos sodium sedation on respiratory rate, oxygen saturation, and heart rate in infants and young children, *Pediatric Pulmonology*, *10*(1), 40–45, https://onlinelibrary.wiley.com/doi/abs/10.1002/ppul.1950100109, https://onlinelibrary.wiley.com/doi/pdf/10.1002/ppul.1950100109.

[41] **Keun, H.C.**, **Ebbels, T.M.**, **Antti, H.**, **Bollard, M.E.**, **Beckonert, O.**, **Holmes, E.**, **Lindon, J.C. and Nicholson, J.K.** (2003). Improved analysis of multivariate data by variable stability scaling: application to NMR-based metabolic profiling, *Analytica Chimica Acta*, *490*(1), 265–276, https://www.sciencedirect.com/science/article/pii/S0003267003000941, papers presented at the 8th International Conference on Chemometrics and Analytical Chemistry.

[42] **Feng, C.**, **Wang, H.**, **Lu, N.**, **Chen, T.**, **He, H.**, **Lu, Y. and Tu, X.M.** (2014). Log-transformation and its implications for data analysis, *Shanghai Arch. Psychiatry*, *26*(2), 105–109, https://doi.org/10.3969/j.issn.1002-0829.2014.02.009.

[43] **Tukey, J.W.** (1957). On the Comparative Anatomy of Transformations, *The Annals of Mathematical Statistics*, *28*(3), 602 – 632, https://doi.org/10.1214/aoms/1177706875.

[44] **Box, G.E.P. and Cox, D.R.** (1964). An Analysis of Transformations, *Journal of the Royal Statistical Society: Series B (Methodological)*, *26*(2), 211–243, https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1964.tb00553.x, https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.2517-6161.1964.tb00553.x.

[45] **Yeo, I. and Johnson, R.A.** (2000). A new family of power transformations to improve normality or symmetry, *Biometrika*, *87*(4), 954–959, https://doi.org/10.1093/biomet/87.4.954, https://academic.oup.com/biomet/article-pdf/87/4/954/633221/870954.pdf.

[46] **Kingma, D.P. and Welling, M.** (2014). *Auto-Encoding Variational Bayes*, 1312.6114.

[47] **Moayyeri, A.**, **Hammond, C.J.**, **Hart, D.J. and Spector, T.D.** (2012). The UK Adult Twin Registry (TwinsUK Resource), *Twin Res Hum Genet*, *16*(1), 144–149.

[48] **Maćkiewicz, A. and Ratajczak, W.** (1993). Principal components analysis (PCA), *Computers and Geosciences*, *19*(3), 303–342, https://www.sciencedirect.com/science/article/pii/009830049390090R.

[49] **Schölkopf, B.**, **Smola, A. and Müller, K.R.** (1997). Kernel principal component analysis, ***W. Gerstner, A. Germond, M. Hasler and J.D. Nicoud**, editors, Artificial Neural Networks — ICANN'97*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp.583–588.

[50] **Yang, T.L.**, **Shen, H.**, **Liu, A.**, **Dong, S.S.**, **Zhang, L.**, **Deng, F.Y.**, **Zhao, Q. and Deng, H.W.** (2020). A road map for understanding molecular and genetic determinants of osteoporosis, *Nature Reviews Endocrinology*, *16*(2), 91–103, https://doi.org/10.1038/s41574-019-0282-7.

[51] **Fahy, E. and Subramaniam, S.** (2020). RefMet: a reference nomenclature for metabolomics, *Nature Methods*, *17*(12), 1173–1174, https://doi.org/10.1038/s41592-020-01009-y.

[52] **Kingma, D.P. and Ba, J.** (2017). *Adam: A Method for Stochastic Optimization*, 1412.6980.

[53] **Tieleman, T. and Hinton, G.** (2014). RMSprop gradient optimization, *http://www.cs. toronto. edu/tijmen/csc321/slides/lecture_slides_lec6. pdf*.

[54] **Hinton, G.E. and Salakhutdinov, R.R.** (2006). Reducing the dimensionality of data with neural networks, *Science*, *313*(5786), 504–507.

[55] **Paszke, A.**, **Gross, S.**, **Massa, F.**, **Lerer, A.**, **Bradbury, J.**, **Chanan, G.**, **Killeen, T.**, **Lin, Z.**, **Gimelshein, N.**, **Antiga, L.**, **Desmaison, A.**, **Kopf, A.**, **Yang, E.**, **DeVito, Z.**, **Raison, M.**, **Tejani, A.**, **Chilamkurthy, S.**, **Steiner, B.**, **Fang, L.**, **Bai, J. and Chintala, S.**, (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library, Advances in Neural Information Processing Systems 32, volume 32, Curran Associates, Inc., pp.8024–8035, https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

CURRICULUM VITAE

Name SURNAME: Sadi ÇELİK

EDUCATION:

- **M.Sc.:** 2024, Istanbul Technical University, Faculty of Computer and Informatics, Department of Computer Engineering
- **B.Sc.:** 2021, Sabanci University, Faculty of Engineering and Natural Sciences, Computer Science and Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2023 | TEKNOFEST | Entrepreneurship Competition | Environment, Energy and Climate Technologies Acceleration Category | Finalist
- 2022 | TEKNOFEST | Artificial Intelligence in Healthcare, Bioinformatics Analysis Development | University and Above Level | 3rd Place Award
- 2021 | Sabanci University | Dean's High Honor List
- 2020 | EnerjiSA Energy | Data Scientist Intern
- 2018 | Sabanci University | Dean's Honor List
- 2017 | Marmara High School | First Graduate Degree

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Çelik S.**, Can B., Çakmak A. (2024). A Multi-Model Strategy to Predict Missing Values in Human Metabolomics Datasets. *International Graduate Research Symposium - IGRS24*, May 8-10, 2024 İstanbul, Turkey.