A DISSERTATION
SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN COMPUTER SCIENCE AND ENGINEERING

Structured and Unstructured Data Analysis for Smart Society

by

Dao Ngoc Hong

June 2024

The thesis titled

Structured and Unstructured Data Analysis for Smart Society

by

Dao Ngoc Hong

is reviewed and approved by:

Chief referee	
Professor	Date Jun. 21, 2024
PAIK Incheon	
Professor	Date June 21, 2024
ZHAO Qiangfu	
Professor	Date June, 21, 2024
MARKOV Konstantin	
Associate Professor	Date June, 21, 2024
RAGE Uday Kiran	

THE UNIVERSITY OF AIZU

March 2024

Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence

BERT Bidirectional Encoder Representations from Transformers

CBOW Continuous Bag of Words

CLIP Contrastive Language-Image Pre-training

CT computerized tomography

DL Deep Learning

ECLAT Equivalent Class Transformation

EHRs Electronic Health Records

FPM Frequent Pattern Mining

ICU Intensive Care Unit

ITL-tree Interval Transaction-ids List tree

KDD Knowledge Discovery in Databases

LMM Large Multimodal Model

maxPer maximum periodicity

MIMIC-III Medical Information Mart for Intensive Care-III

minSup minimum support

MRI magnetic resonance imaging

MSE Mean Squared Error

MTKPP Mining Top-K Periodic-frequent pattern

NLP Natural Language Processing

NLTK Natural language toolkit

PFPM Periodic-frequent Pattern Mining

SPFPM Stable Periodic-Frequent Pattern Mining

SPP-ECLAT Stable Periodic frequent Pattern-Equivalent Class Transformation

SPP-growth Stable Periodic frequent Pattern-growth

SPP-tree stable periodic-frequent tree

SPPs Stable Periodic-frequent Patterns

TL Transfer Learning

Acknowledgment

I am deeply grateful for the guidance and support of numerous individuals whose invaluable assistance made this thesis possible.

I owe a great debt of gratitude to my advisor, Prof. PAIK Incheon, whose unwavering support and expertise have been crucial in completing my PhD work. His dedicated guidance has not only helped me navigate academic challenges in computer science but also provided encouragement during moments of doubt, motivating me to continue my research and achieve this work.

I also want to express my sincere thanks to my co-advisor, Prof. RAGE Uday Kiran, for teaching and guiding me in various aspects, from pattern mining to teamwork dynamics in his lab. His guidance, encouragement, and feedback were instrumental throughout my research.

I extend my gratitude to the members of my thesis committee, Prof. ZHAO Qiangfu and Prof. MARKOV Konstantin, for their valuable comments and insights, which significantly contributed to improving the quality of my research.

I am thankful to all the members of the Intelligent Analytics Laboratory at UoA for creating a stimulating academic environment. Special thanks to Tuyen Nguyen and Cherubin, whose friendship, energy, talent, and support inspired me greatly in my research and achievements.

I want to thank my family for their unwavering support. I'm especially grateful to my wonderful mothers for their encouragement and good health, which kept me going throughout my research path. A big thank you to my beloved children, Khai and Hana, for being such a strong source of motivation. I appreciate their understanding when I was busy and couldn't take good care for them. And most importantly, I want to thank my husband for being my great teacher and constant companion on this journey called life.

Abstract

The emergence of big data, sourced from diverse channels ranging from government records to sensor networks, offers unprecedented opportunities for optimizing urban operations and enhancing citizen well-being within Smart Cities. By utilizing sensors and connected devices to collect and analyze data, Smart Cities optimize operations, manage resources, and improve the quality of life for residents. As urban environments continue to generate vast volumes of data, analyzing the data can help yield descriptive and predictive models crucial for developing data-driven Smart City applications. The evolution of Deep Learning (DL), championed by pioneers such as Geoffrey Hinton, highlights the transition of data from raw information to insightful knowledge, driving growth and innovation.

In the context of Smart Cities, big data systems efficiently store, process, and mine data to enhance various city services and facilitate decision-making. Understanding the different data types—structured, unstructured, and semi-structured—is essential for selecting appropriate mining techniques, as it directly impacts the accuracy, effectiveness, and quality of insights. Proper selection and processing of data types enable the development of precise algorithms, driving innovation and improving Artificial Intelligence (AI) applications to enhance smart societies.

This thesis explores four models for structured and unstructured data mining in the context of smart societies. The first model aims to discover periodic patterns within structured data, offering actionable insights across transportation, marketing, customer services, and air pollution control domains. The second model employs transfer learning techniques for medical image classification, addressing data scarcity in healthcare. Subsequent models delve into multimodal analysis, integrating textual and image data for improved diagnostic accuracy in medical image analysis. Leveraging advanced AI techniques, including prompt engineering, these models enhance computational efficiency and deepen our understanding of clinical data. Lastly, the fourth model makes use of a self-supervised learning approach for mining medical text data, enhancing the quality of healthcare services within smart societies. The research in this thesis is organized as follows.

Chapter 2 focuses on developing an algorithm to uncover significant patterns in various datasets within smart cities, such as transportation, pollution, and consumer behavior. This algorithm aims to identify recurring behaviors within itemsets, providing valuable insights for decision-making processes across different domains. The algorithm is designed for structured temporal databases, ensuring versatility in data analysis. The primary objective of this chapter is to streamline the process of extracting meaningful periodic patterns efficiently while minimizing computational resources and time consumption. By introducing a novel algorithm for frequent pattern mining, our research addresses the fundamental challenge of effectively mining periodic data in structured databases, optimizing cost-effectiveness. Experimental results on six structured

datasets demonstrate that the proposed algorithm is more computationally efficient and scalable than state-of-the-art algorithms.

In Chapter 3, our research centers on data mining within healthcare datasets, mainly focusing on medical image datasets. Integrating healthcare data into Smart Cities is essential for driving citizen well-being and sustainable development by optimizing healthcare delivery in evolving urban ecosystems. The COVID-19 pandemic has emphasized the critical importance of leveraging data-driven solutions to address emerging health threats. In Smart Cities, medical image classification is crucial as the demand for efficient and accurate diagnostic tools increases with urban expansion and aging populations. Medical imaging, including X-rays, MRIs, and CT scans, is pivotal in the early detection, diagnosis, and treatment planning of various medical conditions.

Given these considerations, our research focuses on developing deep-learning models for medical image classification. This chapter presents two models for medical image classification:

1. A transfer learning model that classifies diseases from medical images. This model employs transfer learning techniques using the large multimodal pre-trained model PubMedCLIP on multiple medical image datasets covering various body regions.

2. A multimodal transfer learning model that incorporates medical text prompts alongside medical images. Our experiments show that integrating textual and image features allows this model to outperform state-of-the-art models, even with limited training data.

In Chapter 4, we delve into the analysis of semi-structured healthcare data, particularly within Smart Cities. Electronic Health Records (EHRs) are diverse information repositories, including free-text clinical notes, drug information, and health indices. Despite the richness of insights within these records, extracting meaningful information presents challenges, especially regarding domain specificity. Our research introduces a novel self-supervised method for improved feature extraction. By leveraging tags such as outcomes, diagnosis codes, and categories in EHR data, our deep learning model adapts to the specific characteristics of each dataset. Experimental results show that when a sufficient number of tags are provided, the performance of similarity matching significantly improves.

Chapter 1

Introduction

1.1 Overview

1.1.1 Big Data Analytics and Smart Society

The world is witnessing rapid urbanization, with the urban population projected to reach 4,774 million by 2025, constituting 58.3% of the global population. As highlighted in the 'World Cities Report 2022', this growth is expected to continue at a rate of 0.46% annually from 2020 to 2025 and 0.4% from 2030 to 2035 [2]. As cities expand, their urban environments become increasingly crowded, leading to significant transformations in economic and social landscapes. This rapid urbanization brings about both modernization and new challenges in city management. Issues such as increasing traffic congestion, resource planning on a large scale, air pollution, and delayed healthcare services have emerged as critical concerns. Concurrently, cities are generating vast amounts of data through their dynamic environments. Technological advancements have enabled the collection of massive urban datasets containing valuable insights into city dynamics, offering opportunities for improved management and urban policy development. In recent years, growing research has focused on developing Smart City services and applications that enhance livability and efficiency [3–5]. Data analytics and machine learning play crucial roles in this endeavor, offering algorithms and models for data association, classification, and analysis. These tools enable the extraction of valuable insights for citizens and decision-makers alike. In this thesis, we explore using data analysis to design and develop data-driven smart city services, aiming to address the complex challenges of urbanization and enhance the quality of life in cities.

1.1.2 Data Mining and Knowledge Discovery

Knowledge Discovery in Databases (KDD) is a complex process to uncover valuable insights within existing data. Fayyad et al.'s research [1] provides a comprehensive explanation of the KDD process, shedding light on various approaches within the multidisciplinary field of Knowledge Discovery. This research enhances our understanding of the different methodologies employed in KDD and how they complement each other. In the evolving landscape of the smart society, characterized by the proliferation of IoT devices and the advent of Industry 4.0, the volume and diversity of digital data have surged exponentially. This surge necessitates efficient and effective data processing to distill potentially valuable insights and knowledge, facilitating informed decision-

making. The relentless innovation in information technology has triggered an explosion in the generation, accessibility, and storage of structured and unstructured data sourced from diverse platforms such as corporate databases, online transaction processing systems (OLTP), web platforms, social networks, and the IoT ecosystem.

Consequently, there has been a concurrent evolution in algorithms and technologies for Big Data analytics [6, 7], which is imperative for dealing with the continuous proliferation and distribution of data across networked computing nodes. Despite these advancements, only a fraction of stored data undergoes comprehensive analysis using modern technologies and methodologies. Hence, a pressing need arises to identify suitable data mining methodologies for effectively harnessing specific datasets. Knowledge discovery in databases (KDD) or data mining entails identifying previously unknown or hidden structured and unstructured data. This data is then interpreted to derive practical insights and knowledge, forming a robust knowledge base conducive to sound decision-making. Various KDD tools exhibit distinct strengths and weaknesses, necessitating a methodological approach that aligns with the nature of the data and the specific objectives. Extracting meaningful insights is contingent upon understanding the data's characteristics, often requiring interpretation of identified patterns or dependencies. The heterogeneity and unstructured nature of generated data pose challenges to conventional knowledge discovery approaches, which conventionally handle structured data from singular sources. In the subsequent subsection, we delve into the distinctions between structured and unstructured data, which are pivotal to understanding the data mining landscape in the context of our thesis.

1.1.3 Structured Data and Unstructured Data

Structured Data

Structured data refers to information organized in a specific format, typically rows and columns, facilitating processing and analysis by computer systems. This data type adheres to a clear structure defined by a schema or data model. Examples include numerical data, dates, and strings in relational databases like SQL. Structured data is efficiently indexed and queried, making it ideal for applications ranging from business intelligence to data analytics [8, 9]. The well-defined organization of structured data enhances accessibility and manageability. It simplifies data storage, retrieval, and analysis, catering to users with varying technical expertise. Stable and reliable analytics workflows are possible due to the standardized nature of structured data, enabling businesses to derive insights and make informed decisions effectively. However, structured data only represents about 20% of enterprise data, offering a limited view of business functions. Relying solely on structured data overlooks potential insights that could be derived from unstructured data. Therefore, it is crucial to acknowledge the limitations of structured data and explore the benefits of integrating unstructured data into the data analysis strategy.

Unstructured Data

Unstructured data encompasses information without a predefined data model or schema. This qualitative data includes various formats such as text, video, audio, images, and social media posts. Unlike structured data, which is easily searchable and

analyzable in databases, unstructured data presents challenges in processing and research due to its lack of organization. However, unstructured data offers inherent advantages that can unlock new possibilities across various disciplines [10,11]. It captures real-world nuances and complexities often absent in structured datasets. For instance, analyzing human sentiment, behavior, and interactions in their natural forms becomes possible through unstructured data, which includes text, audio, video, and images. Deep learning engineers can explore diverse data sources without being constrained by predefined formats, contributing to innovation by providing deep learning models with a broader and deeper understanding of the world. In a smart society, with the increasing affordability of data storage and processing technologies, handling vast amounts of unstructured data is becoming more feasible, paving the way for innovative applications and insights.

This thesis explores various data analysis techniques to extract insights from societal data. Specifically, for structured data analysis, the focus is on developing an algorithm to identify stable periodic-frequent patterns in temporal databases. This algorithm applies to real-world smart society datasets like transportation, air pollution monitoring, and market basket analysis. This research model for structured data will be presented in Chapter 2 of this thesis.

In unstructured data, attention is directed towards unimodal and multimodal models tailored for analyzing medical image data. By proposing novel unimodal and multimodal deep learning models, insightful feature information can be extracted from medical images and text for classification tasks. These models and their applications will be discussed in Chapter 3 of this thesis.

Additionally, a self-supervised learning approach is proposed for analyzing medical textual data from the semi-structured data of EHRs. This approach aims to extract similarity features from patient records to facilitate the identification of patient similarities, thereby aiding in the diagnostic process. This research will be presented in Chapter 4 of this thesis.

1.2 Definition

This section outlines the fundamental concepts and terminology used in this study, including frequent pattern mining, transfer learning, multimodal learning, and similarity matching.

1.2.1 Frequent Pattern Mining

Frequent Pattern Mining (FPM) [12] also known as Association Rule Mining, is an analytical process used to discover frequent patterns, associations, or causal relationships within datasets found in various types of databases, such as relational or transactional databases. Given a set of transactions, the goal of this process is to identify rules that can predict the occurrence of specific items or item sets based on the presence of other items within the transactions. To better understand FPM, some key concepts will be introduced as follows:

- **Transaction:** Consider a set $X = \{x_1, x_2, \ldots, x_m\}$ consisting of m items, and let $T = \{t_1, t_2, \ldots, t_n\}$ be a collection of n subsets of these items, known as transactions. Each transaction in T represents a subset of items from X.

- **Frequent itemset:** Given a set of items $X = \{x_1, x_2, \ldots, x_m\}$ and a set of transaction $T = \{t_1, t_2, \ldots, t_n\}$, a subset of X, denoted as S, is called a frequent itemset if it appears in a proportion of transactions in T that exceeds a predefined threshold, known as $Support$.

- **Support:** The support of an itemset Y, represented as support(Y), refers to the count of transactions in T that include the itemset Y.

- **minSup:** Refers to the minimum transaction that a pattern must cover. This is the threshold defined by user.

Table 1.1: Transactions database

Tid	Items
T1	bread, butter, jam, milk
T2	beer, salmon
T3	bread, milk, butter
T4	bread, milk, salmon
T5	coke, salmon

For example, consider the following transaction database containing five transactions depicted in Table1.1. Given a $minSup$ of three transactions, frequent itemsets are "bread, milk", "bread", "milk", "salmon".

1.2.2 Transfer Learning

In traditional machine learning, the training data and testing data mostly have the same data distribution. Whereas, Transfer Learning (TL) is a machine learning approach that enhances a learner's performance in a new domain by leveraging knowledge which was learned in a related source domain. It addresses situations where there is a scarcity of target training data, which may be due to data rarity, high collection and labeling costs, or data inaccessibility. In transfer learning, information from a source domain with ample data is used to enhance the performance of the learner in the target domain, even when the feature spaces and data distributions between the two domains differ. This approach is particularly beneficial as big data repositories become more prevalent, allowing for the utilization of existing datasets related to, but not identical to, the target domain of interest.

A domain D is composed of two key components: a feature space X and a probability distribution over the features $P(X)$. In simpler terms, $D = \{X, P(X)\}$, X represents a set of instances, denoted as $X = \{x \mid x_i \in X,\ i = 1, \ldots, n\}$.

Given a task T, it consists of a label space Y and a predictive function $f(\cdot)$, represented as $Y = \{Y, f(\cdot)\}$. The predictive function $f(\cdot)$ is not explicitly defined but is intended to be derived from the sample data.

Given these definitions, in some machine learning models, a source domain data D_S is defined as $\{(x_{S1}, y_{S1}), ..., (x_{Sn}, y_{Sn})\}$, where x_{Si} represents the ith example in D_S; y_{Si} is its corresponding class label. Similarly, target domain data D_T is denoted as $\{(x_{T1}, y_{T1}), ..., (x_{Tn}, y_{Tn})\}$. Source tasks are denoted as T_S and target tasks are denoted as T_T, with corresponding predictive functions $f_S(\cdot)$ and $f_T(\cdot)$.

Transfer learning aims to improve the function $f_T(\cdot)$ by leveraging information from the source domain D_S and task T_S, where $D_S \neq D_T$ or $T_S \neq T_T$. This process can involve single or multiple source domains. It can be seen that, if $D_S \neq D_T$, it means that the probability distributions $P(X)$ or the feature spaces X of the source and target domains are different.

Heterogeneous transfer learning arises when the feature spaces X are different ($X_S \neq X_T$), while homogeneous transfer learning applies when the feature spaces are the same ($X_S = X_T$).

1.2.3 Multimodal Learning

The term "Multimodal learning", in the context of ML, is a type of DL that uses a combination of various modalities of data commonly encountered in real-world applications. An example of multimodal data is the medical image and its relevant medical text note, where medical images are characterized by pixel intensities and annotation tags, while medical text notes are typically represented as feature vectors [13].

Integration of these varied data types may improve the accuracy and reliability of predictive models, as different data types capture various aspects of a patient's health status. Through the utilization of multiple modalities, multimodal models afford a more comprehensive and holistic comprehension of patient health, thereby facilitating more informed clinical decision-making and ultimately improving patient outcomes.

For instance, integrating radiology scans with medical records has led to significant advancements in tasks related to image understanding, including tumor segmentation in brain scans [14] and analysis of skin images [15]. Similarly, the fusion of medical images and medical records has demonstrated promise, as evidenced by predictive analyses utilizing radiology images alongside clinical records [16, 17]. Notably, these multimodal approaches have surpassed traditional machine learning models, highlighting their potential in precision medicine.

However, the development and optimization of multimodal models for healthcare applications present challenges owing to the heterogeneity and complexity of electronic health record (EHR) data. Essential steps such as data preprocessing, feature selection, and model optimization are pivotal yet demanding. Moreover, ensuring interpretability is imperative, particularly in healthcare contexts, as clinicians necessitate insights into the rationale behind model predictions to make well-informed decisions.

1.2.4 Similarity Matching

Healthcare data generates a vast amount of information across different modalities. Consequently, big data tools, such as patient matching systems, are essential for facilitating analytics. These tools help reduce costs and improve the efficiency of the healthcare system.

Patient similarity analytics involves investigating the distance between patients based on various components of their data. Clustering patients by identifying similarities in their characteristics facilitate efficient computational analyses. These characteristics include information about diseases, hospitalizations, medical imaging, and other clinical data that evaluate medical evidence related to human behavior. One example demonstrating the utility of patient similarity analytics is in the fields of diabetes [18, 19] and cancer [20] research. In these studies, patient similarity metrics are determined using

a Euclidean vector representation. Predictor variables, including laboratory test results and vital signs, establish a similarity metric across multiple patients. The calculation of this metric can be facilitated using a dot product, often referred to as "cosine similarity." The patient similarity metric is defined as follows:

$$\text{PSM}(P_1, P_2) = \frac{P_1 \cdot P_2}{\|P_1\| \cdot \|P_2\|} = \frac{\sum_{i=1}^{n} P_{1i} \times P_{2i}}{\sqrt{\sum_{i=1}^{n} P_{1i}^2} \times \sqrt{\sum_{i=1}^{n} P_{2i}^2}}. \tag{1.1}$$

where P_{1i} and P_{2i} denote the predictor variable vectors for two distinct patients. Also, the dot product $(\cdot)$ calculates the cosine of the angle between the vectors, while $\|\cdot\|$ denotes the Euclidean vector magnitude. Because the metric of patient similarity relies on the cosine of an angle, it is normalized to fall within the range of -1 and 1. For example, two predictor variable vectors that point in precisely opposite directions would have a 180-degree angle between them, resulting in a similarity score of -1 between the patients. Conversely, vectors that overlap perfectly would form an angle of 0 degrees between them, leading to a patient similarity score of 1.

1.3 Scope and Motivation of The Study

This study is motivated by the imperative to extract insights from big data to enhance smart city management and facilitate ease and comfort in urban living. The era of big data presents a challenge in efficiently extracting valuable knowledge from abundant structured data, which necessitates the development of expressive query languages and optimization techniques. Our research in pattern mining focuses on discovering meaningful insights from structured data while optimizing computational resources for large-scale datasets.

In addition, data integration challenges, particularly in medical health data, hinder the effectiveness of machine learning (ML) or deep learning (DL) models. Our focus is on analyzing unstructured medical data using advanced models to develop domain-adapted transfer learning and multimodal learning. These models leverage medical images and related textual information to improve accuracy and performance.

The presence of semi-structured data, such as electronic health records (EHRs), further motivates our research. We aim to extract useful information for diagnosis by leveraging both structured and unstructured data components.

Figure 1.1 illustrates the different types of datasets used in this study. Fig.1.1(a) shows structured data, which includes tabular data where information is organized in a predefined format, making it easier to analyze and query. Fig.1.1(b) depicts unstructured medical image data, comprising medical images that are not organized in a predefined manner, thereby posing challenges for analysis and interpretation. Fig.1.1(c) presents semi-structured data from Electronic Health Records (EHRs), which includes both structured table records and unstructured medical text notes, combining organized and free-form information. These examples highlight the diversity of data types we work with, each posing unique challenges and opportunities for analysis.

Fig.1.2shows the placement of our research in the Knowledge Discovery Database (KDD) framework, highlighting how our work integrates into this process. Our re-

index	ROW_ID	SUBJECT_ID	HADM_ID	ICUSTAY_ID	STARTDATE	ENDDATE	DRUG_TYPE	DRUG	DRUG_NAME_POE	DRUG_NAME_GENERIC	FORMULARY_DRUG_CD	GSN	NDC	PROD_STRENGTH	DOSE_VAL_RX	DOSE_UNIT_RX	FORM_VAL_DISP	FORM_UNIT_DISP	ROUTE
0	2214776	6	107064	NaN	2175-06-11 00:00:00	2175-06-12 00:00:00	MAIN	Tacrolimus	Tacrolimus	Tacrolimus	TACR1	021796	469307711.0	1mg Capsule	2	mg	2	CAP	PO
1	2214775	6	107064	NaN	2175-06-11 00:00:00	2175-06-12 00:00:00	MAIN	Warfarin	Warfarin	Warfarin	WARF5	008562	56017273.0	5mg Tablet	5	mg	1	TAB	PO
2	2215524	6	107064	NaN	2175-06-11 00:00:00	2175-06-12 00:00:00	MAIN	Heparin Sodium	NaN	NaN	HEPAPREMIX	008522	338059302.0	25,000 unit Premix Bag	25,000	UNIT	1	BAG	IV
3	2216265	6	107064	NaN	2175-06-11 00:00:00	2175-06-12 00:00:00	BASE	D5W	NaN	NaN	HEPBASE	NaN	0.0	HEPARIN BASE	250	ml	250	ml	IV
4	2214773	6	107064	NaN	2175-06-11 00:00:00	2175-06-12 00:00:00	MAIN	Furosemide	Furosemide	Furosemide	FURO20	008208	54839735.0	20mg Tablet	20	mg	1	TAB	PO
5	2214774	6	107064	NaN	2175-06-11 00:00:00	2175-06-12 00:00:00	MAIN	Warfarin	Warfarin	Warfarin	WARF5	014116	56019375.0	Check with MD for Dose	1	dose	1	dose	PO
6	2215523	6	107064	NaN	2175-06-12 00:00:00	2175-06-12 00:00:00	MAIN	Heparin Sodium	NaN	NaN	HEPAPREMIX	008522	338059302.0	25,000 unit Premix Bag	25,000	UNIT	1	BAG	IV
7	2216266	6	107064	NaN	2175-06-12 00:00:00	2175-06-13 00:00:00	BASE	D5W	NaN	NaN	HEPBASE	NaN	0.0	HEPARIN BASE	250	ml	250	ml	IV
8	2215526	6	107064	NaN	2175-06-12 00:00:00	2175-06-13 00:00:00	MAIN	Heparin Sodium	NaN	NaN	HEPAPREMIX	008522	338059302.0	25,000 unit Premix Bag	25,000	UNIT	1	BAG	IV
9	2214778	6	107064	NaN	2175-06-12 00:00:00	2175-06-13 00:00:00	MAIN	Warfarin	Warfarin	Warfarin	WARF2	008561	56017075.0	2mg Tab	2	mg	1	TAB	PO
10	2216267	6	107064	NaN	2175-06-12 00:00:00	2175-06-13 00:00:00	BASE	D5W	NaN	NaN	HEPBASE	NaN	0.0	HEPARIN BASE	250	ml	250	ml	IV
11	2214780	6	107064	NaN	2175-06-12 00:00:00	2175-06-13 00:00:00	MAIN	Tacrolimus	Tacrolimus	Tacrolimus	TACR5	021797	469065773.0	5mg Capsule	5	mg	1	CAP	PO
12	2214779	6	107064	NaN	2175-06-12 00:00:00	2175-06-13 00:00:00	MAIN	Tacrolimus	Tacrolimus	Tacrolimus	TACR1	021796	469307711.0	1mg Capsule	2	mg	2	CAP	PO
13	2214781	6	107064	NaN	2175-06-12 00:00:00	2175-06-13 00:00:00	MAIN	Mycophenolate Mofetil	Mycophenolate Mofetil	Mycophenolate Mofetil	MYCO500T	032599	402600?.0	500mg Tablet	1500	mg	3	TAB	PO
14	2215486	6	107064	NaN	2175-06-13 00:00:00	2175-06-13 00:00:00	MAIN	Tacrolimus	Tacrolimus	Tacrolimus	TACR5	021797	469065773.0	5mg Capsule	5	mg	1	CAP	PO

(a) Structured Dataset

(b) Unstructured Medical Images

index	row_id	subject_id	hadm_id	chartdate	charttime	storetime	category	description	cgid	iserror	text
0	738407	20409	NaN	2119-01-04 00:00:00	2119-01-04 12:59:00	NaN	Radiology	ABDOMEN U.S. (COMPLETE STUDY)	NaN	NaN	[**2119-1-4**] 12:59 PM ABDOMEN U.S. (COMPLETE STUDY) Clip # [**Clip Number (Radiology) 20829**] Reason: HEPATITIS ,ELAVATED LFT,S ,DISTENDED ABDOMEN ,EVAL FOR ASCITES ,LIVER MASS _____ [**Hospital 3**] MEDICAL CONDITION: 47 year old man with HEPATITIS REASON FOR THIS EXAMINATION: RULE OUT ASCITES IF ASITES PRESENT TAP _____ FINAL REPORT INDICATION: Hepatitis ?ascites. FINDINGS: The liver is heterogeneous without focal abnormalities. There is no intrahepatic ductal dilatation. The portal vein is patent. There is no ascites. The spleen is enlarged and measures 14 cm. The visualized pancreas is unremarkable. Left kidney measures 13.9 cm in long axis. The right kidney measures 11.9 cm in long axis. There is no hydronephrosis, stones, or masses. IMPRESSION: 1) No evidence of ascites. 2) Heterogeneous liver.
1	738408	20409	NaN	2119-01-09 00:00:00	2119-01-09 13:06:00	NaN	Radiology	MR LIVER WITH CONTRAST	NaN	NaN	[**2119-1-9**] 1:05 PM MR LIVER WITH CONTRAST; MR RECONSTRUCTION IMAGING Clip # [**Clip Number (Radiology) 99985**] Reason: increase afp Contrast: MAGNEVIST Amt: 20 _____ [**Hospital 2**] MEDICAL CONDITION: 47 year old man with increased afp r.o hepatoma or liver lesion REASON FOR THIS EXAMINATION: increase afp _____ FINAL REPORT MR OF LIVER: CLINICAL HISTORY: Increased AFP. Chronic liver disease. TECHNIQUE: Multiple images including dynamic VIBE sequence, and T1 and T2 weighted images obtained. FINDINGS: The liver has a fine granular micronodular pattern, with extensive collaterals noted in the mesentery, and recanalized paraumbilical vein. The spleen is enlarged measuring up to 19 cm in craniocaudal dimension. No enhancing masses are identified in arterial phase, but there are areas showing increased T1 signal, with decreased signal on T2 weighted imaging and some drop on in and out-of-phase imaging consistent with multiple regenerative nodules. The adrenals, pancreas, are normal. Imaged portions of kidneys are unremarkable except for tiny simple cyst seen in lower portion of the right kidney. IMPRESSION: Changes consistent with chronic liver disease with portal hypertension. No MR features to suggest hepatoma.
2	738409	20409	NaN	2119-01-16 00:00:00	2119-01-16 21:24:00	NaN	Radiology	CHEST (PORTABLE AP)	NaN	NaN	[**2119-1-16**] 9:24 PM CHEST (PORTABLE AP) Clip # [**Clip Number (Radiology) 20830**] Reason: verify right IJ (triple lumen) line placement. _____ [**Hospital 3**] MEDICAL CONDITION: 47 year old man with Hep. C/cirrhosis, encephalopathy, and ARF. REASON FOR THIS EXAMINATION: verify right IJ (triple lumen) line placement. _____ FINAL REPORT CHEST, SINGLE AP FILM. The CV line is in distal SVC. NG tube extends below diaphragm. No pneumothorax. The lungs are grossly clear.
3	738410	20409	NaN	2119-01-18 00:00:00	2119-01-18 13:24:00	NaN	Radiology	CT ABD W&W/O C	NaN	NaN	[**2119-1-18**] 1:24 PM CT ABD W&W/O C; CT PELVIS W/CONTRAST Clip # [**Clip Number (Radiology) 99986**] CT 150CC NONIONIC CONTRAST Reason: EEVAL FOR LIVER LAC Field of view: 40 Contrast: OPTIRAY Amt: _____ [**Hospital 2**] MEDICAL CONDITION: 47 year old man s/p mva [**1-12**]. Now w upper abd pain. r/o liver lac. REASON FOR THIS EXAMINATION: r/o liver lac _____ FINAL REPORT INDICATION: Abdominal pain. Assess for liver laceration. PRIORS: Outside prior from [**Hospital6 52283**] dated [**2119-1-14**]. TECHNIQUE: unenhanced ct images of the abdomen were obtained followed by post contrast images of the abdomen and pelvis. CT ABDOMEN WITH AND WITHOUT CONTRAST: The liver shows evidence of cirrhosis unchanged from the prior without evident laceration. However since the prior exam, there has been interval development of perihepatic fluid extending into the paracolic gutters bilaterally and into the pelvis. This may represent ascites vs hemorrhage. Recanalization of the umbilical vein is seen consistent with cirrhosis; extensive varices and splenomegaly are present and unchanged. The adrenals, kidneys, pancreas, gallbladder are unremarkable. Diffuse bowel wall thickening is seen involving the small bowel, colon and perhaps the proximal transverse colon. The SMA however is patent from the origin at least to the first 2 cm. The distal SMA is not well assessed. The lung bases show no mass or nodules although bilateral pleural effusions are present. CT PELVIS WITH CONTRAST: The pelvic bowel loops are unremarkable. Again noted is the pelvic free fluid. IMPRESSION: Free fluid within the abdomen which may represent blood vs ascites, bowel wall thickening involving the small bowel and ascending colon suggesting SMA territory compromise but the proximal SMA appears patent in this limited evaluation of the SMA. This finding my be related to patient's cirrhosis and portal hypertension.
4	738411	20409	NaN	2119-01-18 00:00:00	2119-01-18 15:45:00	NaN	Radiology	PARACENTESIS DIAG. OR THERAPEUTIC	NaN	NaN	[**2119-1-18**] 3:45 PM PARACENTESIS DIAG. OR THERAPEUTIC; US ABD LIMIT, SINGLE ORGAN Clip # [**Clip Number (Radiology) 20831**] GUIDANCE FOR [**Female First Name (un) **]/ABD/PARA CENTESIS US Reason: ? CAUSE OF ASCITES, BLOOD, INFECTION IN MAN S/P TRAUMA _____ [**Hospital 3**] MEDICAL CONDITION: 47 year old man with cirrhosis, presenting with liver failure following traumatic hemothorax. REASON FOR THIS EXAMINATION: increased abdominal fluid on follow up CT - ? ascites / SBP / blood from trauma _____ FINAL REPORT HISTORY: Cirrhosis-exclude spontaneous bacterial peritonitis. FINDINGS: Informed consent obtained. Small amount of ascites identified abuting the liver in the right upper quadrant. Patient was prepped and draped in a sterile fashion. Anesthesia achevied using 1% Xylocaine. A 20 gauge needle was advanced into the fluid under ultrasound guidance, and approximately 25 ml of straw colored fluid aspirated and sent for analysis as per primary team. The patient tolerated the procedure without immediate complications. Dr. [**Last Name (STitle) 4070**] present for procedure. IMPRESSION: Successful ultrasound guided paracentesis.
5	738412	20409	NaN	2119-01-19 00:00:00	2119-01-19 12:52:00	NaN	Radiology	CHEST (PA & LAT)	NaN	NaN	[**2119-1-19**] 12:52 PM CHEST (PA & LAT); STERNUM Clip # [**Clip Number (Radiology) 100241**] Reason: hep c cirrhosis, s/p mva with R hemothorax, hep c cirrhosis, _____ [**Hospital 2**] MEDICAL CONDITION: 47 year old man with R hemothorax REASON FOR THIS EXAMINATION: hep c cirrhosis, s/p mva with R hemothorax _____ FINAL REPORT INDICATION: Status post MVA with right hemothorax. CHEST, PA AND LATERAL AND LATERAL STERNAL VIEW: Right IJ line tip at the distal SVC. Low lung volumes are present. There is no evidence of failure. Note is made of a sternal fracture with the distal component displaced approximately the width of the sternum anteriorly to the superior component. There are small bilateral pleural effusions. IMPRESSION: 1. Sternal fracture, as described above. 2. Small bilateral pleural effusions.
											[**2119-1-20**] 1:12 PM US ABD LIMIT, SINGLE ORGAN Clip # [**Clip Number (Radiology) 20924**] Reason: hep B cirrhosis, new onset ascites, with weight gain 5 kgs in one day, pressure on sternum. Please evaluate for large _____ [**Hospital 3**] MEDICAL CONDITION: 47 year old man with see above REASON FOR THIS EXAMINATION: hep B cirrhosis, new onset ascites, with weight gain 5 kgs in one day, pressure on sternum. Please evaluate for large

(c) Semi-structured EHR Data

Figure 1.1: Illustration of the different types of dataset have been used this study

search is driven by the need to address challenges in mining insights from various data modalities, including structured, unstructured, and semi-structured data. The overarching goal of this dissertation is to propose optimization methods for mining insightful patterns and predictive models. To achieve this goal, we focus on the development and utilization of the following key components:

- **Pattern Mining Algorithm (SPP-ECLAT)**: The proposed SPP-ECLAT algorithm is designed to discover stable periodic-frequent patterns within large columnar temporal databases. By efficiently extracting these patterns, the algorithm identifies recurring behaviors and trends in smart city datasets, such as transportation, air pollution monitoring, and market basket analysis. Its ability to operate efficiently in real-world scenarios enhances the decision-making process for smart city services while optimizing computational resources and time.

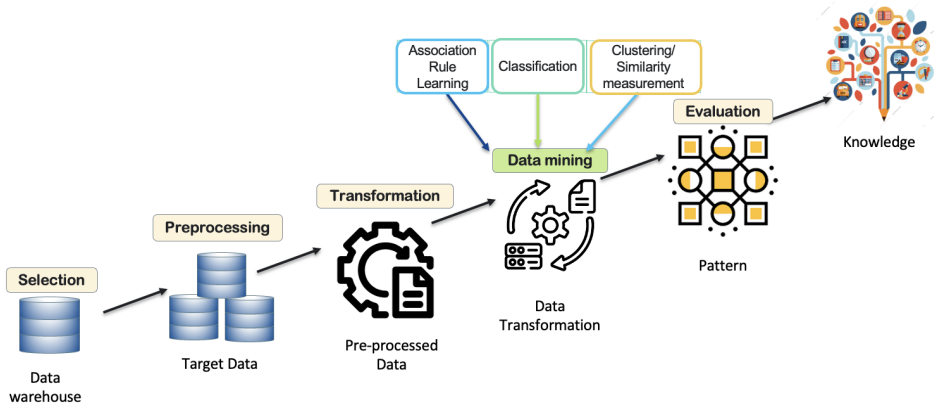


Figure 1.2: Our Research Context in the Knowledge Discovery Database (KDD) Process ([1])

- **Transfer Learning Method**: Utilizing a pre-trained Large Multimodal Model (LMM), our transfer learning method addresses the challenge of limited training data in medical image analysis. The pre-trained LMM, trained on large medical image datasets, enhances classification accuracy across multiple medical modalities. This approach improves diagnostic capabilities and reduces the need for extensive data labeling and collection, making it particularly valuable in resource-constrained healthcare settings.

- **Multimodal Model Learning**: The proposed multimodal model integrates medical images and textual data to enhance medical diagnosis performance. By leveraging the complementary information from these modalities, the model achieves higher accuracy and robustness in predicting disease outcomes. This integration enables a holistic understanding of patient conditions, leading to more informed clinical decision-making and personalized treatment strategies.

- **Self-supervised Learning Method for Patient Similarity**: Our self-supervised learning method employs self-supervised learning with tags to improve the performance of patient similarity matching. By exploring feature embedding and fusion techniques, this method refines the understanding of patient similarity and enhances the accuracy of clinical decision support systems. It enables healthcare professionals to identify relevant patient cohorts more effectively, facilitating personalized treatment plans and improving patient outcomes.

In conclusion, we aim to tackle the complexities of mining insights from diverse data modalities in smart city management domains. By developing novel algorithms and leveraging advanced machine learning techniques, our research seeks to enhance the efficiency, accuracy, and effectiveness of data analysis processes. Through the proposed optimization methods, we aspire to contribute to the advancement of smart city management and healthcare systems, ultimately improving the quality of life for urban residents and patients alike.

1.4 Dissertation Outline

The dissertation outlined herein comprises several key components, as depicted in Figure 1.3. It delineates the structure across five chapters:

- **Chapter 1:** Provides an overview of the research, introducing the background on big data analytics and smart cities. It defines key concepts related to the dissertation's title, spanning four chapters, and outlines the research scope. Additionally, it discusses the challenges and opportunities of data analysis techniques in structured, unstructured, and semi-structured data types within the big data era for smart cities, along with insights on future work.

- **Chapter 2:** Presents a study focused on developing algorithms for mining significant patterns, specifically stable periodic-frequent patterns, applicable to various structured datasets. This chapter offers valuable insights into decision-making processes across different domains. Understanding existing solutions that provide optimal performance and memory reduction in pattern mining from big structured data is essential.

- **Chapter 3:** Concentrates on data analysis within healthcare datasets, particularly employing a deep learning approach for unstructured healthcare data. Two models are proposed in this chapter. Firstly, a transfer learning model utilizing a pre-trained model for medical image classification tasks is discussed, showcasing its ability to perform well across multiple medical image modalities and addressing limitations in training data. Secondly, a multimodal transfer learning model is introduced, incorporating medical images and related text inputs. The study also explores the efficacy of prompt techniques to guide the model for improved learning, addressing challenges posed by limited training data.

- **Chapter 4:** Introduces a method for patient similarity using semi-structured data, specifically Electronic Health Record (EHR) data in healthcare. Different embedding models and a self-supervised learning method are employed for better feature extraction. By leveraging various tags such as outcomes, diagnosis codes, and categories in EHR data, the deep learning model adapts to the specific characteristics of the given dataset.

- **Chapter 5:** Concludes the dissertation by providing an overall evaluation across three data types (structured, unstructured, and semi-structured data), highlighting the contributions and limitations of the methods presented. It also offers exciting insights into potential future research directions, hinting at the transformative impact our work could have in the near future.

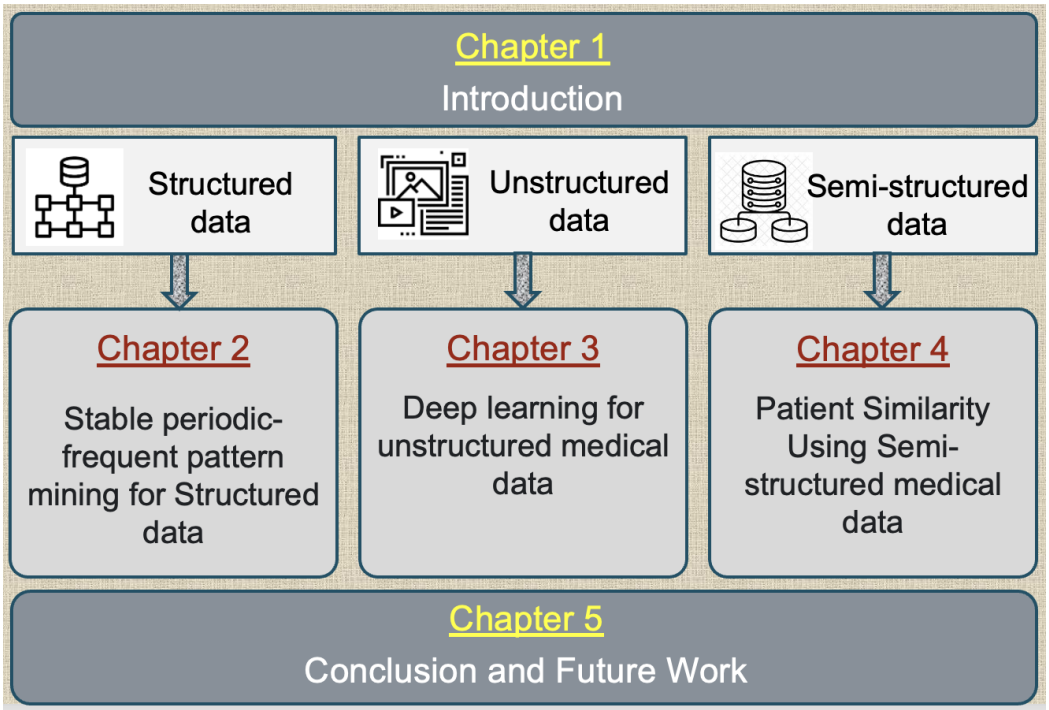


Figure 1.3: Illustration outlining the structure of the dissertation

Chapter 2

Pattern Mining with Structured Data

2.1 Introduction

Database systems play a crucial role in storing the big data generated by real-world applications. Depending on the layout used for storing the data, one can broadly classify the databases into two types: *row databases* and *columnar databases*[1]. Row databases are primarily based on ACID[2] properties and organize the data as records by keeping the data associated with a record next to each other in a storage device. The popular row databases include MySQL [21] and Postgres [22]. In contrast, columnar databases are based on BASE [3] properties and organize data into fields and store all of the data associated with a field next to each other in a storage device. The popular columnar databases include BigQuery [23], HBase [24], and Snowflake [25]. Both row and columnar databases have their respective advantages and disadvantages. Henceforth, no universally accepted best data layout exists for any given application. Selecting the right database layout is subjective to user and/or application requirements.

Extracting meaningful information from the data is a crucial task of data mining. Frequent pattern mining (FPM) [12, 26–31] is a renowned data mining technique that aims to discover all frequently occurring patterns in the data. Numerous algorithms have been presented in the literature to discover frequent patterns effectively. One can broadly classify these approaches into the following three types: (i) candidate-generate-and-test algorithms (e.g., Apriori [27] and Partitioning [32]), (ii) pattern-growth algorithms (e.g., FP-Growth [33], HMine [34], and [35]), and (iii) vertical format algorithms (e.g. ECLAT [36] and CHARM [37]). The candidate-generate-and-test and pattern-growth approaches can find frequent patterns only in row databases, while vertical format approaches can find frequent patterns in both row and columnar databases. Henceforth, researchers are putting forth efforts to develop efficient vertical format algorithms. This study also develops an efficient vertical format algorithm to discover a class of frequent patterns, called stable periodic-frequent patterns, in columnar temporal databases.

A temporal database represents a temporally ordered set of transactions. Crucial information that can empower the domain experts to gain competitive advantage lies hidden in this data. Tanbeer et al. [38] described the model of periodic-frequent pattern to discover regularities in a temporal database. This model involves discovering all pat-

[1]Columnar and row databases are referred as vertical and horizontal databases, respectively

[2]ACID is an acronym for Atomicity, Consistency, Isolation, and Duration

[3]BASE is an acronym for Basically Available, Soft state, and Eventually consistent

terns in a database that satisfy the user-specified *minimum support (minSup)* and *maximum periodicity (maxPer)* constraints. The $minSup$ controls the minimum number of transactions in which a pattern must appear in the database. The $maxPer$ controls the maximum time interval within which an pattern must reappear. A classic application of periodic-frequent pattern mining is market-basket analytics. It involves identifying the patterns that the customers regularly purchase in a supermarket. An example of a periodic-frequent pattern is as follows:

$$\{Cheese, Wine\}\ [support = 20\%, periodicity = 5\ hour].$$

The above pattern provides information that 20% of the customers have purchased the products 'Cheese' and 'Wine' at least once every 5 hours. Therefore, supermarket managers may find this information beneficial for inventory management and product placement.

In the literature, the periodic-frequent pattern model was extended to discover fuzzy periodic-frequent pattern [39], rare periodic-frequent pattern [40], partial periodic pattern [41, 42], and high utility periodic-frequent pattern [43]. However, the successful real-world adoption of this model has been affected by the following obstacle: "*Since $maxPer$ controls the maximum inter-arrival time of an pattern in a database, the basic model of periodic-frequent pattern considers any pattern uninteresting if anyone of its inter-arrival time is more than the user-specified $maxPer$ value [44, 45]. In other words, the strict restriction that* ***all periods*** *of a pattern must be within the user-specified $maxPer$ constraint often prunes all of those interesting patterns that have exhibited stable (or partial) periodic behavior in a database.*"

When confronted with this problem in real-world applications, researchers introduced the model of stable periodic-frequent pattern [46] to find all of those interesting pattern that have exhibited stable periodic behavior in columnar database. This model provides a function to find interesting patterns that have a stable periodic behavior. A pattern-growth algorithm, called Stable Periodic-frequent Pattern-growth (SPP-growth), was described to find stable-periodic patterns in temporal databases. Unfortunately, this algorithm can discover the interesting patterns in row (temporal) databases only. Therefore, whenever we give a columnar temporal database as an input to the SPP-growth algorithm, it has to be converted into a row temporal database to get interesting patterns. As a result, the above algorithms will take longer to run and use more memory because of this conversion overhead. With this motivation, this study proposes a generic algorithm to find stable periodic-frequent pattern in both row and columnar temporal databases effectively. To the best of our knowledge, this is the first algorithm that focuses on finding stable periodic-frequent pattern in columnar temporal database. It should be noted that existing algorithms find the patterns only in row databases.

Discovering stable periodic-frequent pattern in columnar databases is significant and challenging because of some reasons as follows:

1. The importance of discovering frequent pattern in columnar databases was first discussed in the work of Zaki et al. [47], where the depth-first-search algorithm, named Equivalent Class Transformation (ECLAT), was proposed to extract frequent pattern in a columnar database. However, the ECLAT algorithm cannot be directly applied to find stable periodic-frequent pattern in a columnar temporal database. The reason is ECLAT algorithm completely disregards the temporal occurrence information of an pattern in the data.

2. Reducing search space (itemset lattice) is a challenging task in pattern mining. The process of recursively mining the constructed tree increases the memory and runtime requirements of the SPP-growth algorithm.

3. One can transform a columnar temporal database into a row database and then apply those available algorithms to extract stable periodic-frequent pattern. However, we should avoid such a transformation process due to its high computational cost.

Against this backdrop, we have extended the functionality of ECLAT [47] to mine stable periodic-frequent patterns by introducing a new algorithm called Stable Periodic frequent Pattern-Equivalent Class Transformation (SPP-ECLAT) in columnar temporal database. This study extends the related work by extensively understanding the current literature, presenting the complexity analysis of our algorithm, and performing in-depth experiments studying the memory, runtime, and scalability of the mining algorithms. In this thesis, we show that SPP-ECLAT outperformsStable Periodic frequent Pattern-growth (SPP-growth) [46] on both synthetic and real-world databases by a very large margin.

The key contributions of this study are summarized as follows:

1. An efficient and novel SPP-ECLAT algorithm is proposed to ensure that the discovered Stable Periodic-frequent Patterns (SPPs) not only satisfy the user-specified *minimum support* and *maximum periodicity* thresholds but are stable patterns based on the user-specified *maximum lability* threshold in any big columnar temporal databases.

2. In SPP-ECLAT, the observed $Lability$ information is stored in a unique, compact list-based data structure called SPP-List. The newly introduced *maximum lability* measure considers the periodic behavior of an pattern as stable when the lability value is low. On the other hand, if the value is high, it means the patterns are unstable. So stable pattern can be found using this measure, given a limit on the maximum lability.

3. On six synthetic and real-world databases, we compare the performance of the proposed SPP-ECLAT algorithm against that of the current state-of-the-art SPP-Growth algorithm. This indicates that the SPP-ECLAT algorithm outperforms the SPP-Growth algorithm with respect to runtime requirements and memory consumption. Furthermore, the scalability of the SPP-ECLAT algorithm is also shown to demonstrate the efficacy and productivity of the proposed algorithm on big columnar databases relative to those of the state-of-the-art SPP-Growth algorithm.

2.2 Related Work

In this section, we will review the previous work related to frequent pattern mining, periodic-frequent pattern mining, and stable periodic-frequent pattern mining.

2.2.1 Frequent Pattern Mining

Argawal et al. [26] introduced frequent pattern mining to find interesting relationships among different data items. An algorithm, called Apriori [12], was also introduced to discover all frequent patterns in a row (transactional) database. This algorithm works in a breadth-first manner that uses frequent k-itemsets to form candidate $(k+1)$-itemsets, from which frequent $(k+1)$-itemsets are obtained. Many extensions of Apriori have been proposed in the literature [32] [48]. Essentially, they have the same general structure, with some additional techniques to optimize certain steps within the algorithm. Though Apriori can find all wanted frequent pattern, it has to scan the database several times to generate a complete set of pattern. Thus, it is a very time-consuming process. Beside Apriori algorithm, Argawal et al. [49] proposed two other algorithms called AprioriTid and AprioriHybrid. The AprioriTid algorithm reduces the processing time of the support counting procedure by replacing every transaction in the database with a set of candidate itemsets that appears in that transaction. This is done repeatedly at every iteration k. It is demonstrated in [27] that although AprioriTid is much faster in the later iterations, it performs slower than Apriori in early iterations. Therefore, the AprioriHybrid algorithm has been proposed [49], which combines Apriori and AprioriTid. Basically, the hybrid algorithm uses Apriori for the initial iterations and then switches to AprioriTid. Even though the AprioriTID algorithm have utilized a vertical database representation, this algorithm is based on the breadth-first search technique.

The first algorithm to generate all frequent patterns in a depth-first search manner, called Eclat, is proposed by Zaki [50]. Eclat is a vertical database layout algorithm. This algorithm utilizes the TID-list data structure for the mining task. Eclat applies the depth-first approach to find frequent pattern and scan the database only two times. In the first round, it scans the entire database to find all frequent items. In the second round, the TID-list of the frequent items is generated. The Eclat algorithm uses common $(k-1)$-prefixes to organize frequent k-itemsets into disjoint equivalence classes. Then the candidate $(k+1)$ itemsets can be found by joining two frequent k-itemsets from the same classes. The main advantage of utilizing TID-list is that, only by intersecting the TID-lists of the two subsets, the support of a candidate pattern is simply computed. A simple check on the received TID-list can tell whether the new pattern is frequent.

The frequent pattern-growth (FP-growth) algorithm proposed by Han et al. [28] is a tree-based algorithm to discover frequent pattern in a database. This algorithm uses the divide-and-conquer method. In this algorithm, frequent pattern are mined from the fp-tree, and there is no need for a candidate frequent pattern. In the first step, a list of frequent patterns is generated and sorted in their descending support order. This list is represented as a node structure, containing the item name, support count, and a pointer to a node in the tree that has the same prefix. These nodes then are used to create an fp-tree. The paths from the root to leaf nodes are arranged in the decreasing order of their support. Frequent pattern are extracted from the fp-tree starting from the leaf nodes. To mine frequent pattern(s) each prefix path subtree is processed recursively. The only differences between Eclat and FP-growth are the process to count the support of every candidate pattern and how they represent the database. In fact, it is difficult to say which algorithm performs better. Over two decades, many other FPM algorithms have been proposed, mainly by extending the Apriori, Eclat, and FP-growth algorithms to find a frequent pattern. However, frequent pattern mining algorithms are inapplicable to identify pattern that appear in a temporal database regularly.

Besides, many studies focus on finding new kinds of pattern and rules present in a large amount of data. This is especially important with the emergence of Big data. Over nearly 30 past years, various pattern have been identified, namely sequential and time-series pattern [51] [52] [53], high utility pattern [54] [55] [56], structural pattern [57] [58], temporal (periodic) pattern [59] [60] [61] [62].

2.2.2 Periodic-frequent Pattern Mining

The target of periodic frequent pattern mining is to identify how regularly the pattern occur in a temporal database. In Tanbeer et al. [38], the problem of mining periodic frequent pattern was first introduced and correspondingly a model called PF-growth was proposed to tackle this problem. Compared to the classic FPM which only employs the $minSup$ constraint, Periodic-frequent Pattern Mining (PFPM) includes one more parameter called $maxPer$. This algorithm performs in two steps. First, it represents the database by a periodic-frequent tree (PF-tree), and items in a PF-tree are arranged in the descending item support order. Second, the algorithm mines the PF-tree by using FP-growth mining technique to find all periodic-frequent pattern.

Amphawan et al. [63] proposed an efficient algorithm called Mining Top-K Periodic-frequent pattern (MTKPP), which is based on a depth-first search and a vertical database representation. This algorithm mines periodic-frequent pattern without using the $minSup$ constraint and provides a list-based data structure called the top-K list to maintain the set of k regular pattern with the highest support. MTKPP algorithm uses this top-K list during the mining process to generate candidate pattern. Uday et al. [64] introduced an efficient model that extended multiple $minSup$'s and multiple $maxPer$'s to discover periodic-frequent pattern consisting of both frequent and rare items. This model used two different constraints to identify useful pattern, namely minimum item support and maximum item periodicity. Each pattern satisfies different $minSup$ and $maxPer$ values based on the available items in the pattern. That study also proposed a pattern-growth algorithm using a novel and efficient tree-based data structure, named a multi-constraint periodic-frequent tree, to find the complete set of frequent and rare items.

Amphawan et al. [63] proposed a novel technique called approximate periodicity to reduce the calculation time requirements of mining periodic-frequent pattern. This algorithm splits the transactional timeline into intervals with different $maxPer$ values. The interval information is stored only when there exists a pattern in that interval. The authors also proposed a tree structure, called Interval Transaction-ids List tree (ITL-tree). The goal of this technique is to maintain the occurrence information in a highly compact manner by using interval transaction-ids list instead of tid-list. Then the approximate periodicity of each pattern can be found. To generate all periodic-frequent patterns, a pattern growth mining technique is also used by a bottom-up traversal of the ITL-tree based on $minSup$ and $maxPer$ constraints. An interesting novel measure was proposed by Uday et al. [65] to extract periodic-frequent pattern in a transactinonal database, which is called periodic-ratio. The authors defined that some pattern which appear almost periodically in the database can be considered interesting pattern. Therefore, a periodic interestingness of a frequent pattern is calculated as the ratio of its periodic occurrences in a database. A pattern can be defined as a potential pattern if its support is greater than $minSup$ and its periodic interestingness is greater than the user-specified minimum periodic-ratio. Then an extended periodic-frequent tree was built based on these potential pattern. Also a pattern-growth algorithm was proposed

to find the complete set of periodic-frequent pattern. Ravi et al. [66] have described an algorithm named PF- ECLAT, to efficiently discover periodic-frequent patterns in a columnar temporal databases. Some other variations of the above models were also proposed [67] [68] [69] to find periodic-frequent pattern. However, all these algorithms have a drawback that, if only one of the periods of a pattern exceeds $maxPer$, this pattern is discarded. Kiran et al. [59] proposed a model called partial PFP mining that relaxes the maximum periodicity constraint by considering that a pattern X is (partial) periodic if its periodic-frequency is no less than a user-specified threshold. However, this algorithm cannot be applied to find stable-periodic-frequent pattern. The reason is that it measures the periodicity of a pattern by counting the number of times where the periods of a pattern are less than $maxPer$ without considering how much these periods deviate from $maxPer$.

To overcome the drawback of periodic frequent pattern mining, Philippe et a. [46] proposed a model to find stable periodic-frequent pattern in a transactional database.

2.2.3 Stable Periodic-frequent Pattern Mining

Philippe et al. [46] introduced a concept called $lability$, which is the cumulative sum of the difference between each period length and $maxPer$ constraint. A novel parameter, called *maximum lability* ($maxLa$), was also used to assess the stability of a periodic behavior of a pattern in a database. An algorithm named SPP-growth to mine stable periodic-frequent pattern was presented with two steps. First, the database is represented as a stable periodic-frequent tree (SPP-tree), and then the algorithm mines the SPP-tree to find all stable periodic-frequent pattern. Ruimeng et al. [70] discussed a model to find stable periodic-frequent pattern in an uncertain database. The authors proposed a Stable Periodic-Frequent Pattern Mining (SPFPM) algorithm on an uncertain database by considering both the frequency and periodicity of pattern. That is, an pattern X in an uncertain transaction database is considered a stable periodic frequent pattern if the support count and stability value of pattern X meet the minimum support threshold ($minSup$) and the stability threshold ($maxLa$). Phillipe et al. [71] proposed a model using the concept of top-K mining to generate stable periodic-frequent pattern. This study introduced an algorithm that, rather than using a $minSup$ threshold, the user can directly specify parameter k, where k represents the number of pattern that the user wants to find. The output of the algorithm is the top-K most frequent pattern that have a stable periodic behavior. To the best of our knowledge, up to now, there have been only three above references related to the study of finding a Stable Periodic-Frequent pattern in row databases.

Because all of the above algorithms find the patterns in row databases, whenever we give a columnar temporal database as an input to the above algorithms, it has to be converted into a row temporal database to get interesting patterns. As a result, the above algorithms will take longer to run and use more memory because of this conversion overhead. In contrast, the algorithm proposed in our work is different in that it deals specifically with columnar databases.

Table 2.1: Row database

ts	**items**
1	*b,c,d,e*
2	*a,b,c*
3	*a,b,c,d*
4	*e,f*
5	*a,c,d*
6	*a,b,c*

ts	**items**
7	*d,e,f*
8	*b,c,d*
9	*a,b,c,d*
10	*a,b,c*
11	*a,c,e*
12	*b,c,d*

Table 2.2: Columnar database

	items					
ts	*a*	*b*	*c*	*d*	*e*	*f*
1	0	1	1	1	1	0
2	1	1	1	0	0	0
3	1	1	1	1	0	0
4	0	0	0	0	1	1
5	1	0	1	1	0	0
6	1	1	1	0	0	0

	items					
ts	*a*	*b*	*c*	*d*	*e*	*f*
7	0	0	0	1	1	1
8	0	1	1	1	0	0
9	1	1	1	1	0	0
10	1	1	1	0	0	0
11	1	0	1	0	1	0
12	0	1	1	1	0	0

Table 2.3: Item's dictionary with their timestamp list

item	**TS-list**
a	2,3,5,6,9,10,11
b	1,2,3,6,8,9,10,12
c	1,2,3,5,6,8,9,10,11,12
d	1,3,5,7,8,9,12
e	1,4,7,11
f	4,7

2.3 Model of SPP

Let $O = \{o_1, o_2, \cdots, o_n\},\ n \geq 1,$ be a set of objects (or items). Let $Y \subseteq O$ be an patterns (or a pattern). Let $t_a = (ts,\ X)$, $a \geq 1$, be a transaction, where $ts \in \mathbb{R}^+$ represents the timestamp and X is an patterns. Let $TDB = \{t_1, \cdots,\ t_d\},\ d \geq 1$ be a temporal database representing an ordered set of transactions such that $t_a.ts \leq t_b.ts$, where $1 \leq a < b \leq d$. Let $TS^Y = \{ts_i^Y, \cdots, ts_j^Y\}, i,\ j \in [1, d]$, denote a set of timestamps containing Y in TDB.

Example 1. Assume that we have a set of items $I = \{a, b, c, d, e, f\}$. Table 2.1 shows a row temporal database constituting of these items. Without loss of generality, this database can be viewed as a columnar temporal database as shown in Table 2.2. In Table 2.3, we show the dictionary storing the items and their temporal occurrence information in the database. The set of items 'b' and 'c', i.e., $\{b,\ c\}$ is a patterns. This patterns will be represented as 'bc' for brevity. This patterns is denoted as 2-patterns because it contains two items. The occurrences of patterns 'bc' are at the timestamps of $1,\ 2,\ 3,\ 6,\ 8,\ 9,\ 10$, and 12. Therefore, we have a list of timestamps containing 'bc', i.e., $TS^{bc} = \{1,\ 2,\ 3,\ 6,\ 8,\ 9,\ 10,\ 12\}$.

Definition 1. (**The** $support$ **of** Y**.**) The **support** of Y, denoted as $sup(Y)$, represents the number of transactions containing Y in TDB. That is, $sup(Y) = |TS^Y|$.

Example 2. The $support$ of 'bc', i.e., $sup(bc) = |TS^{bc}| = 8$.

Definition 2. (**Frequent patterns** Y**.**) The patterns Y is a **frequent patterns** if $sup(Y) \geq minSup$, where $minSup$ is a ***minimum support*** value specified by user.

Example 3. If $minSup = 5$, then bc is said to be a frequent patterns because $sup(bc) \geq minSup$.

Definition 3. ($Periodicity$ **of** Y**.**) Let ts_m^Y and ts_n^X, $j \leq m < n \leq k$, denote two consecutive timestamps in TS^Y. The time difference between ts_n^Y and ts_m^Y is given by a ***period*** of Y, denoted by p_z^Y. That is, $p_z^Y = ts_n^Y - ts_m^Y$. Denoted $P^Y = (p_1^Y,\ p_2^Y, \cdots,\ p_n^Y)$ the set of all $periods$ for patterns Y. The ***periodicity*** of Y, denoted by $per(Y) = maximum(p_1^Y,\ p_2^Y, \cdots,\ p_n^Y)$.

Example 4. All periods of the patterns 'bc' are : $p_1^{bc} = 1\ (= 1 - ts_{initial})$, $p_2^{bc} = 1\ (= 2 - 1)$, $p_3^{bc} = 1\ (= 3 - 2)$, $p_4^{bc} = 3\ (= 6 - 3)$, $p_5^{bc} = 2\ (= 8 - 6)$, $p_6^{bc} = 1\ (= 9 - 8)$, $p_7^{bc} = 1\ (= 10 - 9)$, $p_8^{bc} = 2\ (= 12 - 10)$, and $p_9^{bc} = 0\ (= ts_{final} - 12)$, where first transaction time stamp is denoted by $ts_{initial} = 0$ and the last transaction's time stamp is denoted by, $ts_{final} = |TDB| = 12$. The ***periodicity*** of bc, i.e., $per(bc) = maximum(1,\ 1,\ 1,\ 3,\ 2,\ 1,\ 1,\ 2,\ 0) = 3$.

Definition 4. (**Periodic-frequent patterns** Y**.**) The frequent patterns Y be considered as **periodic-frequent patterns** if $per(Y) \leq maxPer$, here $maxPer$ is ***maximum periodicity*** value which is specified by user.

Example 5. Let the user-specified $maxPer = 3$, in this case the frequent patterns 'bc' is called as a periodic-frequent patterns as $per(bc) \leq maxPer$.

Definition 5. (**$Lability$ of an patterns**). Let ts_{i+1}^Y and ts_i^Y, $i \in [0, sup(Y)]$, be two consecutive time stamps where Y occurs in TDB. We call i-th $lability$ of Y denoted by $la(Y, i) = max(0, la(Y, i-1) + p_i^Y - maxPer)$, where $la(Y, -1) = 0$. For simplicity, the following short form is used

$$la(Y, i) = max(0, la(Y, i-1) + ts_{i+1}^Y - ts_i^Y - maxPer)$$

The following is a list of periods which represent the $lability$ of an patterns Y: $la(Y) = \{la(Y, 0), la(Y, 1), \cdots, la(Y, sup(Y))\}$, and $|la(Y)| = |per(Y)| = sup(Y) + 1$.

Example 6. Consider an item a. If $maxPer$=2, then the $lability$ of a are: $la(a, 0) = max(0, la(p, -1) + p_0^p - maxPer) = max(0, 0 + 2 - 2) = 0$, $la(a, 1) = max(0, 0 + 1 - 2) = 0$, $la(a, 2) = max(0, 0 + 2 - 2) = 0$, $la(a, 3) = max(0, 0 + 1 - 2) = 0$, $la(a, 4) = max(0, 0 + 3 - 2) = 1$, and $la(a, 5) = max(0, 1 + 1 - 2) = 0$. Therefore, the sequence of labilities of a in the database, i.e., $la(a) = \{0, 0, 0, 0, 1, 0\}$.

Based on Definition 5, the periodic patterns can be considered as stable ($lability$ is zero) if all its periods are less than or equal to $maxPer$. The $lability$ of a period of a patterns will increase when a period of a patterns larger than $maxPer$, and these exceeding values are accumulated using the measure of $lability$. The value of $lability$ will be reduced when periods of a patterns no more than $maxPer$. Therefore, according

to the periodic characteristic of a patterns, its *lability* will vary over time, and each value exceeding $maxPer$ is accumulated. A periodic behavior is considered stable when $lability$ value is low while a high value means an unstable one. So stable patterns can be found using this measure given a limit on the *maximum lability*.

Definition 6. (Stable periodic-frequent patterns). For a patterns Y, denote la(Y) the set of all i-th $lability$. The stability of the patterns is defined by $maxLa$(Y) = max(la(Y)). patterns Y is a SPP if $sup(Y) \geq minSup$ and $maxla(Y) \leq maxLa$.

Example 7. Given the above example, if the user specified $minSup$=4, $maxPer$=2, and $maxLa$ = 1, the complete set of SPPs are a: (7,1), b: (8,1), c: (10,0), d: (7,1), bc: (8,1),bca: (5,1), cd: (6,1), ca: (7,1), where each SPP Y is annotated with Y: (*sup(Y), maxLa(Y)*).

Be noted that if $maxLa$ = 0, SPPs are the traditional PFPs. Therefore, the PFPs is a special case of SPPs.

Definition 7. (Problem definition). Given a temporal database (TDB) with *minimum support* ($minSup$), *maximum periodicity* ($maxPer$), and *maximum lability* ($maxLa$) constraints, our objective is to discover the complete set of stable periodic-frequent patterns having $support$ higher or equal to $minSup$ and $lability$ lower or equal to $maxLa$ constraints.

2.4 The Proposed Algorithm: SPP-ECLAT

The patterns lattice represents the search space of stable periodic-frequent patterns mining. The size of this lattice is $2^n - 1$, where n represents the total number of items in a database. Using the *downward closure property* (see Property 1) and the depth-first search technique, the proposed SPP-ECLAT searches this huge lattice and finds the complete set of SPPs. Briefly, the SPP-ECLAT algorithm involves the following two steps: (i) find the stable periodic items (or 1-patternss) from a database (Section 2.4) and (ii) discover the complete set of stable periodic k-patternss, $k > 1$, by recursively mining the previously generated stable periodic patternss (Section 2.4). We now explain each of these steps in detail.

Property 1. If A is a stable periodic-frequent patterns, then $\forall A \subset B$ and $A \neq \emptyset$, A is also a stable periodic-frequent patterns.

Mining 1-stable periodic-frequent patterns

The proposed algorithm can find stable periodic-frequent patternss in both row and columnar databases. The proposed algorithm achieves this ability by transforming a row and columnar database into a unified data structure constituting of candidate items and transaction identifiers. This data structure is called SPP-list.

Denote $SPP\text{-}list = (Y, TS\text{-}list(Y))$ a dictionary with the temporal occurrence information of a patterns in a TDB; TS_l is a temporary variable of list type to store the *timestamp* of the final occurrence of a patterns; la and ML are temporary variable of list type to store the *lability* and the *Maximum Lability* of a patterns; $last$ is a term for the final timestamp; $support$ is a temporary varibale of list type to store the *support* of

a patterns. This part focuses on discovering 1-patterns by SPP-list. The detailed steps are shown in Algorithm 1, which works on a row database shown in Table 2.1. Let $minSup = 5$ and $maxPer = 2$ and $maxLa = 1$.

The 1-patterns are first generated by reading the whole database transactions at once. Then, the row database is converted to the columnar database. After reading the 1^{st} transaction, "$1 : b, c, d, e$", with $ts_{cur} = 1$ inserts the items $b,\ c,\ d$ and e, in the SPP-list. We have the timestamps of these items is $1\ (= ts_{cur})$. Similarly, ML and TS_l contents were updated to 0 and 1, respectively (lines 7 and 8 in Algorithm 1). Fig. 2.1(a) shows the generated SPP-list from the 1^{st} transaction. After reading the 2^{nd} one, "$2 : a, b, c$", with $ts_{cur} = 2$ inserts the new items p into the SPP-list by adding $2\ (= ts_{cur})$ in their TS-list. At the same instant, the ML and TS_l contents were updated to 0 and 2, respectively. Besides $2\ (= ts_{cur})$ was added to the TS-list of existing items q with ML and TS_l contents were updated to 0 and 2, respectively (lines 10 and 13 in Algorithm 1). The SPP-list which is generated after reading the 2^{nd} one is shown in Fig. 2.1(b). After reading the 3^{rd} one, "$3 : a, b, c, d$", updates the TS-list, ML and TS_l values of $a,\ b,\ c$, and d in the SPP-list. Fig. 2.1(c) shows the SPP-list which is generated after reading the $3^{b,c,d,a}$ one. After reading the 4^{th} one, "$4 : e, f$" with $ts_{cur} = 4$, inserts the new items e and f into the SPP-list by adding $4\ (= ts_{cur})$ in their TS-list. Simultaneously, the ML and TS_l values as 2 and 4. Fig. 2.1(d) shows the SPP-list which is generated after reading the 4^{th}. We repeat the whole process for the remaining transactions. Fig. 2.1(e) depicts the final SPP-list which is generated after scanning the whole database. The patterns e and f are pruned (using the Property 1) from the SPP-list as its $support$ value is no more than the $minSup$ value and ML value is greater than $maxLa$ (lines 15 to 20 in Algorithm 1). The complete list of patterns available in the SPP-list are considered as 1-stable periodic-frequent patterns. Those patterns are sorted in descending order in terms of their $support$ values. Fig. 2.1(f) shows the final SPP-list.

Finding all interesting patterns from SPP-List

The detailed procedure for finding stable periodic-frequent patterns is shown in Algorithm 2. Given the newly generated SPP-list, the procedure of this algorithm is carried out as follows. Initially we choose the patterns b, as this is the initial patterns in the SPP-list (line 2 in Algorithm 2). Fig. 2.2(a) shows a record of its $support$ and $lability$. Since b is a stable periodic-frequent patterns, we move to its child node bc . TS-list of bc is generated by performing intersection of TS-lists of b and c, i.e., $TS^{bc} = TS^{b} \cap TS^{c}$ (lines 2 and 3 in Algorithm 2). This $support$ and $lability$ of bc are recorded, as shown in Fig. 2.2(b). We check whether bc is a stable periodic-frequent patterns or unstable periodic frequent patterns (line 4 in Algorithm 2). Since bc is stable periodic-frequent patterns we move it to its child node bcd. Next, TS-list will be generated by performing the intersection of TS-lists of bc and d, i.e., $TS^{bcd} = TS^{bc} \cap TS^{d}$. Fig. 2.2(c)shows a record of $support$ and $lability$ of bcd. Then bcd is identified as an unstable periodic-frequent patterns because a $lability$ of bcd is greater than $maxLa$, the patterns bcd will be remove from the stable periodic-frequent patterns list as shown in Fig. 2.2(c). We repeat the process to find all stable periodic-frequent patterns for remaining nodes in the tree. Fig. 2.2(d) shows the final list of generated stable periodic-frequent patterns. Since we can reduces the search space and the computational cost effectively our proposed approach is efficient.

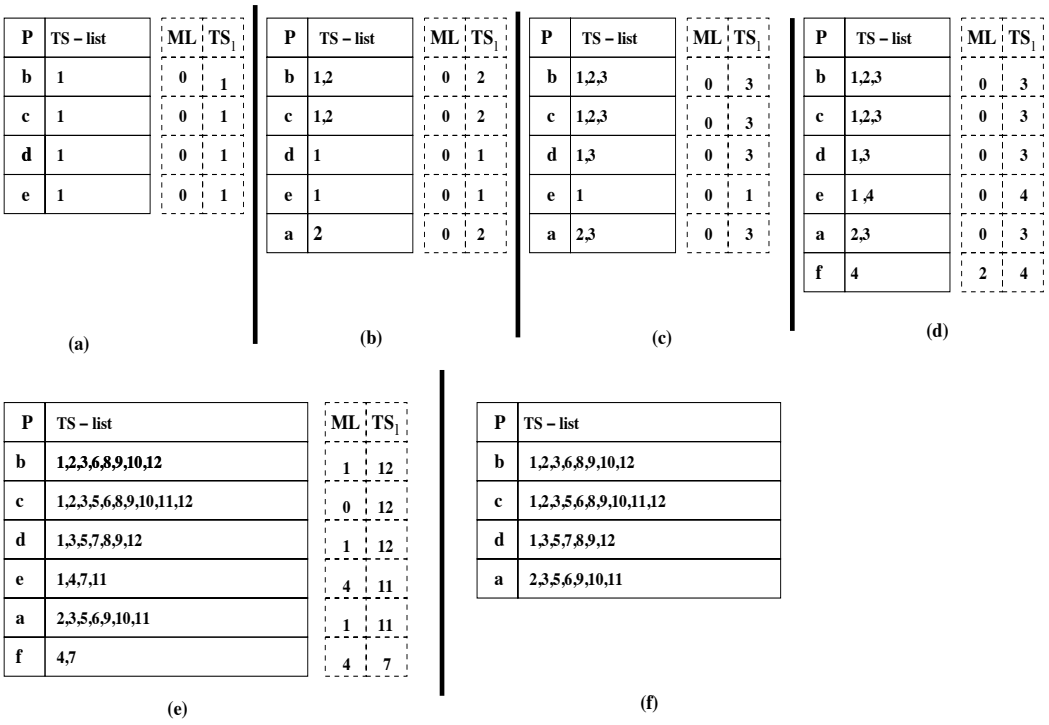


Figure 2.1: SPP-list generation process. (a) content of the list after reading the 1^{st} transaction, (b) after reading the 2^{nd} one, (c) after reading the 3^{rd} one, (d) after reading the 4^{th} one, (e) Final content after reading the whole database, and (f) The complete list of 1-stable periodic-frequent patterns

2.5 Experiments

This section evaluates the performance of the SPP-ECLAT against the state-of-the-art SPP-growth [46] algorithm. Through experiment results, we will show that the SPP-ECLAT algorithm is more efficient in memory consumption and runtime than SPP-growth. The scalability of the SPP-ECLAT algorithm is also shown to demonstrate the superior efficacy and productivity over the SPP-Growth algorithm on big columnar temporal databases. The implementations of these two algorithms are available at PAMI [72]. Note that the metric for runtime is seconds and memory is bytes throughout the experimentation.

2.5.1 Experimental Setup

The algorithms, SPP-growth and SPP-ECLAT, were developed in Python 3.7 and executed on a Gigabyte R282-z94 rack server machine containing two AMD EPIC 7542 CPUs and 600 GB RAM. The operating system of this machine is Ubuntu Server OS 20.04. The experiments have been conducted on various real-world databases including T10I4D100K, Retail, T20I6D100K, BMS-WebView-1, BMS-WebView-2 and Mushrooms. The characteristics of these databases are shown in the Table 3.5. The **T10I4D100K** and **T20I6D100K** synthetic databases are generated according to the properties of market basket data. The procedure of constructing these databases is described in [73]. These spare databases have been widely employed to evaluate var-

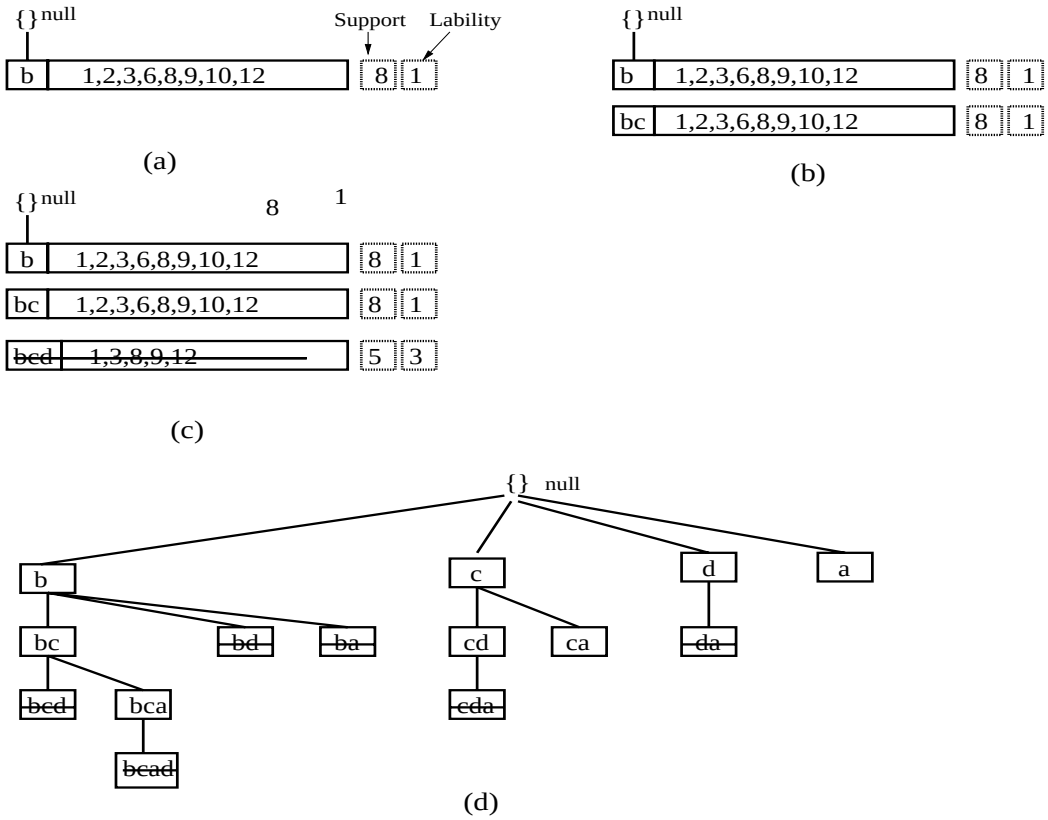


Figure 2.2: The complete process of discovering stable periodic-frequent patterns using SPP-ECLAT algorithm

ious pattern-mining algorithms in the literature. The **BMS-WebView-1** and **BMS-WebView-2** are a real-world sparse databases containing clickstream data from e-commerce sites. Each transaction is a viewing session consisting of all the viewed product detail pages where each product detail view is an item. These databases contain very long transactions and they were used in KDD CUP 2000 competition [74] . The **Retail** is a real-world sparse database consisting of basket databases in a retail supermarket store. The Retail database is provided by Brijs et al. [75]. The **Mushrooms** is a real-world dense database containing different species of gilled mushrooms prepared from the UCI mushrooms dataset. All of the above databases have been downloaded from SPMF repository [76].

Table 2.4: Statistics of the databases

S.No	Database	Type	Nature	Sparsity	Transaction Length	Database Size
1	T10I4D100K	Synthetic	Sparse	0.988	1	100,000
2	Retail	Real	Sparse	0.999	2	88,162
3	T20I6D100K	Synthetic	Sparse	0.978	2	199,844
4	BMS-WebView-1	Real	Sparse	0.995	1	59,602
5	BMS-WebView-2	Real	Sparse	0.999	2	77,512
6	Mushroom	Real	Dense	0.993	23	8,124

Algorithm 1 StablePeriodicFrequentItems(Temporal database (TDB), minimum support ($minSup$), maximum periodicity ($maxPer$), maximum Lability($maxLa$):

Definition: $SPP\text{-}list = (Y, TS\text{-}list(Y))$ is a dictionary with the temporal occurrence information of a patterns in a TDB; TS_l is a temporary variable of list type to store the *timestamp* of the final occurrence of a patterns; la and ML are temporary variable of list type to store the *lability* and the *Maximum Lability* of a patterns; $last$ is a term for the final timestamp; $support$ is a temporary varibale of list type to store the *support* of a patterns.
Initate $ts_{cur} = 0$
for each transaction $t_{cur} \in TDB$ **do**
    Set $ts_{cur} = t_{cur}.ts$;
    for each item $j \in t_{cur}.Y$ **do**
        if j does not exit in SPP-list **then**
            SPP-list is updated by inserting j and corresponding timestamp value
            $la[j] = max(0, ts_{cur} - maxPer)$. Set $ML[j] = la[j]$
        else
            Add j's timestamp in the SPP-list.
            $la[j] = max(0, la[j] + ts_{cur} - TS_l[j] - maxPer)$
            $ML[j] = max(la[j], ML[j])$
            Update $TS_l[j] = ts_{cur}$.
        end if
    end for
    $last = ts_{cur}$
end for
for each item j in SPP-list **do**
    $la[j] = max(0, la[j] + last - TS_l[j] - maxPer)$
    $ML[j] = max(la[j], ML[j])$
    $s[j] = length(TS\text{-}list[j])$
    if $s[j] < minSup$ and $ML[j] > maxla$ **then**
        Prune j from SPP-list
    end if
end for
After the pruning the final list of patterns available in the SPP-list is sorted in ascending order or descending order of the corresponding patterns's $support$. Initiate pi as Null. Call SPP-ECLAT(SPP-List, pi).

Algorithm 2 SPP-ECLAT(SPP-List, pi)

for each item j in SPP-List **do**
    Set $Y = j \cup pi$ and $TS^Y = TS^j \cap TS^{pi}$;
    Calculate $support$ and $lability$ of X;
    if $sup(TS^Y) \geq minSup$ and $la(TS^Y) \leq maxla$ **then**
        Add j to pi and Y is considered as stable periodic-frequent patterns;
        $SPP\text{-}ECLAT$(SPP-list[j+1:], pi);
    end if
end for

2.5.2 Experiment-1: Varying $minSup$ and $maxLa$

In this experiment, we study the impact of $minSup$ and $maxLa$ constraints on the number of patterns generated, the runtime requirements of SPP-Growth and SPP-ECLAT algorithms, and the memory consumed by SPP-Growth and SPP-ECLAT algorithms. Please note that the $maxPer$ constraint has been fixed at a particular value for each database throughout this experimentation.

First, we have shown the number of SPPs generated by SPP-Growth and SPP-ECLAT algorithms in Figure 2.3 by varying the value of $maxLa$. In detail, Figure 2.3(a) to Figure 2.3(c), shows the number of SPPs generated by both the algorithms in the T10I4D100K database, respectively, for different $minSup$ and $maxPer$ values. From Figure 2.3(d) to Figure 2.3(f), shows the number of SPPs generated by both the algorithms in the Retail database, respectively, for different $minSup$ and $maxPer$ values. From Figure 2.3(g) to Figure 2.3(i), shows the number of SPPs generated by both the algorithms in the T20I6D100K database, respectively, for different $minSup$ and $maxPer$ values. From Figure 2.3(j) to Figure 2.3(l), shows the number of SPPs generated by both the algorithms in the BMS-WebView1 database, respectively, for different

$minSup$ and $maxPer$ values. From Figure 2.3(m) to Figure 2.3(o), shows the number of SPPs generated by both the algorithms in the BM-WebView-2 database, respectively, for different $minSup$ and $maxPer$ values. From Figure 2.3(p) to Figure 2.3(r), shows the number of SPPs generated by both the algorithms in the Mushrooms database, respectively, for different $minSup$ and $maxPer$ values. It is to be noted that both algorithms will generate an equal number of patterns. Therefore, both the curves were overlapped throughout the figures. The following two observations can be drawn from these figures: (i) The $minSup$ constraint has negative effect on the generation of SPPs. That is, increase in $minSup$ decreases the number of SPPs, and vice-versa. It is because many patterns fail to satisfy the increased $minSup$ value. (ii) The $maxLa$ constraint has positive effect on the generation of SPPs in the T10I4D100k, T20I6D100k, Retai, and BMS-WebView1 sparse database. That is, increase in $maxLa$ increases the number of SPPs, and vice-versa. It is because higher $maxLa$ values facilitate the patterns to have their inter-arrival times further away from the user-specified $maxPer$ value. (iii) In the dense Mushrooms database, the $maxLa$ constraint does not affect the generation of SPPs. It is because, in a dense database, the mined periodic-frequent patterns are the ones that appear regularly in the database, i.e. having a stable behavior.

Next, we have shown the runtime requirements of SPP-Growth and SPP-ECLAT algorithms in Figure 2.4 by varying the value of $maxLa$. In detail, Figure 2.4 (a) to Figure 2.4(c), shows the runtime requirements of both the algorithms in the T10I4D100K database, respectively, for different $minSup$ and $maxPer$ values. Figure 2.4 presents the performance comparison of the two algorithms in three cases, minSup = 0.5% and maxPer = 0.4% (Figure 2.4(a)), minSup = 0.8% and maxPer = 0.4% (Figure 2.4(b)), minSup = 1% and maxPer = 0.4% (Figure 2.4(c)). The results in these three cases all show that SPP-ECLAT is faster than SPP-Growth.

From Figure 2.4(d) to Figure 2.4(f), shows the runtime requirements of both the algorithms in the Retail database, respectively, for different $minSup$ and $maxPer$ values. Figure 2.4 presents the performance comparison of the two algorithms in three cases, minSup = 0.8% and maxPer = 2% (Figure 2.4(d)), minSup = 0.9% and maxPer = 2% (Figure 2.4(e)), minSup = 1% and maxPer = 2% (Figure 2.4(f)). The results in these three cases all show that, in general, SPP-ECLAT is faster than SPP-Growth.

From Figure 2.4(g) to Figure 2.4(i), shows the runtime requirements of both the algorithms in the T20I6D100K database, respectively, for different $minSup$ and $maxPer$ values. The runtime analysis is given in Figure 2.4 shows the performance of the two algorithms in three cases, minSup = 3% and maxPer = 4% (Figure 2.4(g)), minSup = 5% and maxPer = 4% (Figure 2.4(h)), minSup = 7% and maxPer = 4% (Figure 2.4(i)). The results in these three cases all show that SPP-ECLAT is faster than SPP-Growth.

From Figure 2.4(j) to Figure 2.4(l), shows the runtime requirements of both the algorithms in the BMS-WebView1 database, respectively, for different $minSup$ and $maxPer$ values. Figure 2.4 depicts the performance comparison of the two algorithms in three cases, minSup = 0.5% and maxPer = 5% (Figure 2.4(j)), minSup = 0.8% and maxPer = 5% (Figure 2.4(k)), minSup = 1% and maxPer = 5% (Figure 2.4(l)). The results in these three cases all show that SPP-ECLAT is faster than SPP-Growth.

From Figure 2.4(m) to Figure 2.4(o), shows the runtime requirements of both the algorithms in the BMS-WebView2 database, respectively, for different $minSup$ and $maxPer$ values. Figure 2.4 shows the performance comparison of the two algorithms in three cases, minSup = 0.06% and maxPer = 5% (Figure 2.4(m)), minSup = 0.08% and maxPer = 0.5% (Figure 2.4(o)), minSup = 0.1% and maxPer = 0.5% (Figure 2.4(l)). The

results in these three cases show that SPP-ECLAT is faster than SPP-Growth; however, the difference in runtime comparison between the two algorithms is small, it is only around 0.3 seconds on average .

From Figure 2.4(p) to Figure 2.4(r), shows the analysis for the runtime requirements of both the algorithms in the Mushrooms database, respectively, for different $minSup$ and $maxPer$ values. Figure 2.4 presents the performance comparison of the two algorithms in three cases, minSup = 6% and maxPer = 3% (Figure 2.4(m)), minSup = 7% and maxPer = 3% (Figure 2.4(n)), minSup = 8% and maxPer = 3% (Figure 2.4(o)). The results in these three cases all show that SPP-Growth requires more time than SPP-ECLAT.

It can be observed that the SPP-ECLAT runs faster than the SPP-Growth algorithm. The good performance of SPP-ECLAT is a result of the effectiveness of periodic calculation and pruning techniques. The following are some noteworthy findings that can be derived from this figure: (i) If we increase the $maxLa$ value, then subsequently, both algorithms' runtime requirements increase. The primary reason for this observation is that both the algorithms will discover many SPPs in any database if the $maxLa$ value continues to increase. (ii) SPP-ECLAT generates SPPs much faster than SPP-Growth under any given $maxLa$ in BMS-WebView-1, Retail, T10I4D100K, and T20I6D100K, and Mushrooms databases. More importantly, we can also observe that at high $maxLa$ values, SPP-ECLAT algorithm generates the SPPs much faster than SPP-Growth algorithm. The reason is that SPP-ECLAT uses the downward closure property and the depth-first search technique, so the SPPs are generated by simply performing intersection of SPP-list. The process is repeated to find all SPPs. (iii)With the BMS-WebView-2 dataset, which contains long transactions and many distinct items, the SPP-ECLAT algorithm takes more time than the SPP-Growth algorithm. It is because the SPP-ECLAT algorithm is based on the downward closure property and the depth-first search technique, so it does not require scanning the database each time; but to generate all SPPs, it has to perform the intersection of the SPP-list. So the long SPP-list requires more time to repeat the intersection process.

Finally, we have shown the memory consumption details of SPP-Growth and SPP-ECLAT algorithms in Figure 2.5 by varying the value of $maxLa$. In detail, Figure 2.5(a) to Figure 2.5(c), shows the memory consumption of both the algorithms in the T10I4D100K database, respectively, for different $minSup$ and $maxPer$ values. Figure 2.5(a) depicts the comparison of the two algorithms in three cases, minSup = 0.5% and maxPer = 0.4% (Figure 2.5(a)), minSup = 0.8% and maxPer = 0.4% (Figure 2.5(b)), minSup = 1% and maxPer = 0.4% (Figure 2.5(c)). The results all show that SPP-ECLAT consumes less memory than SPP-Growth in all cases. When maxLa is 0.1%, SPP-ECLAT consumes less memory than SPP-Growth by 26 MB on average. As maxLa is increased, the performance gap becomes bigger (up to 62 MB). From Figure 2.5(d) to Figure 2.5(f), shows the memory consumption of both the algorithms in the Retail database, respectively, for different $minSup$ and $maxPer$ values. Figure 2.5 depicts the performance of the two algorithms in three cases, minSup = 0.8% and maxPer = 2% (Figure 2.5(d)), minSup = 0.9% and maxPer = 2% (Figure 2.5(e)), minSup = 1% and maxPer = 2% (Figure 2.5(f)). The results all show that SPP-ECLAT performs consistently and consumes less memory than SPP-Growth by 145 MB on average.

From Figure 2.5(g) to Figure 2.5(i), shows the memory consumption of both the algorithms in the T20I6D100K database, respectively, for different $minSup$ and $maxPer$ values. Figure 2.5 presents the performance comparison of the two algorithms in three

cases, minSup = 3% and maxPer = 4% (Figure 2.5(g)), minSup = 3.5% and maxPer = 4% (Figure 2.5(h)), minSup = 4% and maxPer = 4% (Figure 2.5(i)). In these cases, SPP-ECLAT consumes less memory than SPP-Growth by around 130MB on average.

From Figure 2.5(j) to Figure 2.5(l), shows the memory consumption of both the algorithms in the BMS-WebView1 database, respectively, for different $minSup$ and $maxPer$ values. Figure 2.5 shows the performance comparison of the two algorithms in three cases, minSup = 0.5% and maxPer = 5% (Figure 2.5(j)), minSup = 0.8% and maxPer = 5% (Figure 2.5(k)), minSup = 1% and maxPer = 5% (Figure 2.5(l)). The gain of SPP-ECLAT in terms of memory here is about 4MB on average.

From Figure 2.5(m) to Figure 2.5(o), shows the memory consumption of both the algorithms in the BMS-WebView2 database, respectively, for different $minSup$ and $maxPer$ values. Figure 2.5 shows the performance comparison of the two algorithms in three cases, minSup = 0.6% and maxPer = 5% (Figure 2.5(m)), minSup = 0.8% and maxPer = 5% (Figure 2.5(n)), minSup = 1% and maxPer = 0.5% (Figure 2.5(o)). The results show that the memory consumption of SPP-ECLAT is around 18MB less than SPP-Growth.

From Figure 2.5(p) to Figure 2.5(r), shows the memory consumption of both the algorithms in the Mushroom database, respectively, for different $minSup$ and $maxPer$ values. Figure 2.5 presents the performance of the two algorithms in three cases, minSup = 6% and maxPer = 3% (Figure 2.5(m)), minSup = 7% and maxPer = 3% (Figure 2.5(n)), minSup = 8% and maxPer = 3% (Figure 2.5(o)). In these cases, SPP-ECLAT consumes less than 58MB on average.

It can be observed that the SPP-ECLAT consumes less memory than the SPP-Growth algorithm. The following are some noteworthy findings that can be derived from this figure: (i) If we increase the $maxLa$ value, then subsequently, both algorithms' memory consumption increase. The primary reason for this observation is that both the algorithms will discover many SPPs in any database if the $maxLa$ value continues to increase. (ii) SPP-ECLAT generates SPPs using a SPP-list structure, which helps reduce the search space on every database. (iii) With the BMS-WebView-2 dataset, the processing time of the proposed algorithm is comparable to the SPP-Growth algorithm; however, it is interesting to note that the SPP-ECLAT algorithm requires much less memory than the SPP-Growth algorithm.

2.5.3 Experiment-2: Varying $minSup$ and $maxPer$

In the previous experiment, we have evaluated the performance of the SPP-Growth and SPP-ECLAT algorithms by varying $minSup$ and $maxLa$ values. In this experiment, we study the impact of $minSup$ and $maxLa$ constraints on the number of patterns generated, the runtime requirements of SPP-Growth and SPP-ECLAT algorithms, and the memory consumed by SPP-Growth and SPP-ECLAT algorithms. Please note that the $maxLa$ constraint has been fixed at a particular value for each database throughout this experimentation. First, the number of SPPs generated by SPP-Growth and SPP-ECLAT algorithms is shown in Figure 2.6 by varying the value of $maxPer$. In detail, Figure 2.6(a) to Figure 2.6(c), shows the number of SPPs generated by the two algorithms in the T10I4D100K database, respectively, for different $minSup$ and $maxLa$ values. From Figure 2.6(d) to Figure 2.6(f), shows the number of SPPs generated by the two algorithms in the Retail database, respectively, for different $minSup$ and $maxLa$ values. From Figure 2.6(g) to Figure 2.6(i), shows the number of SPPs generated by

the two algorithms in the T20I6D100K database, respectively, for different $minSup$ and $maxLa$ values. From Figure 2.6(j) to Figure 2.6(l), shows the number of SPPs generated by the two algorithms in the BMS-WebView-1 database, respectively,for different $minSup$ and $maxLa$ values. From Figure 2.6(m) to Figure 2.6(o), shows the number of SPPs generated by the two algorithms in the BMS-WebView-2 database, respectively,for different $minSup$ and $maxLa$ values. From Figure 2.6(p) to Figure 2.6(r), shows the number of SPPs generated by the two algorithms in the Mushrooms database respectively,for different $minSup$ and $maxLa$ values. It is to be noted that both algorithms will generate an equal number of patterns. Therefore, both the curves were overlapped throughout the figures. The following two observations can be drawn from these figures: (i) The $maxPer$ constraint has positive effect on the generation of SPPs. That is, increase in $maxPer$ increases the number of SPPs, and vice-versa. It is because, if we increase the value of the $maxPer$, then most of the non-periodic patterns have become periodic with an increase in the maximum inter-arrival time duration. (ii) The $minSup$ constraint has negative effect on the generation of SPPs. That is, increase in $minSup$ decreases the number of SPPs, and vice-versa. It is because many patterns fail to satisfy the increased $minSup$ value.

Next, the runtime requirements of SPP-Growth and SPP-ECLAT algorithms is shown in Figure 2.7 by varying the value of $maxPer$. In detail, Figure 2.7(a) to Figure 2.7(c), shows the runtime requirements of the two algorithms in the T10I4D100K database, respectively, for different $minSup$ and $maxLa$ values. Figure 2.7 presents the performance comparison of the two algorithms in three cases, minSup = 0.5% and maxLa = 0.4% (Figure 2.7(a)), minSup = 0.8% and maxLa = 0.4% (Figure 2.7(b)), minSup = 1% and maxLa = 0.4% (Figure 2.4(c)). The results in these three cases all show that SPP-Growth requires more time than SPP-ECLAT. From Figure 2.7(d) to Figure 2.7(f), shows the runtime requirements of the two algorithms in the Retail database, respectively, for different $minSup$ and $maxLa$ values.Figure 2.7 presents the performance comparison of the two algorithms in three cases, minSup = 0.5% and maxLa = 2% (Figure 2.7(d)), minSup = 0.8% and maxLa = 2% (Figure 2.7(e)), minSup = 1% and maxLa = 2% (Figure 2.4(f)). The results in these three cases all show that SPP-Growth requires more time than SPP-ECLAT. From Figure 2.7(g) to Figure 2.7(i), shows the runtime requirements of the two algorithms in the T20I6D100K database, respectively, for different $minSup$ and $maxLa$ values.Figure 2.7 presents the performance comparison of the two algorithms in three cases, minSup = 3% and maxLa = 2% (Figure 2.7(g)), minSup = 3.5% and maxLa = 2% (Figure 2.7(h)), minSup = 4% and maxLa = 4% (Figure 2.4(i)). The results in these three cases all show that SPP-Growth requires more time than SPP-ECLAT.

From Figure 2.7(j) to Figure 2.7 (l), shows the runtime requirements of the two algorithms in the BMS-WebView-1 database, respectively,for different $minSup$ and $maxLa$ values. Figure 2.7 presents the performance comparison of the two algorithms in three cases, minSup = 0.5% and maxLa = 2% (Figure 2.7(j)), minSup = 0.8% and maxLa = 2% (Figure 2.7(k)), minSup = 1% and maxLa = 2% (Figure 2.4(l)). The results in these three cases all show that SPP-Growth requires more time than SPP-ECLAT.

From Figure 2.7(m) to Figure 2.7(o), shows the runtime requirements of the two algorithms in the BMS-WebView-2 database, respectively,for different $minSup$ and $maxLa$ values. Figure 2.7 presents the performance comparison of the two algorithms in three cases, minSup = 0.6% and maxLa = 2% (Figure 2.7(m)), minSup = 0.8% and maxLa = 2% (Figure 2.7(n)), minSup = 1% and maxLa = 2% (Figure 2.4(o)). The re-

sults in these three cases all show that SPP-Growth requires less time than SPP-ECLAT.

From Figure 2.7(q) to Figure 2.7(r), shows the runtime requirements of the two algorithms in the Mushrooms database respectively,for different $minSup$ and $maxLa$ values. Figure 2.7 presents the performance comparison of the two algorithms in three cases, minSup = 6% and maxLa = 3% (Figure 2.7(p)), minSup = 7% and maxLa = 3% (Figure 2.7(q)), minSup = 8% and maxLa = 3% (Figure 2.4(r)). The results in these three cases show that SPP-Growth requires more time than SPP-ECLAT.

It can be observed that the SPP-ECLAT runs faster than the SPP-Growth algorithm in most case. The good performance of SPP-ECLAT is a result of the effectiveness of periodic calculation and pruning techniques. The following are some noteworthy findings that can be derived from this figure: (i) If we increase the $maxPer$ value, then subsequently, both algorithms' runtime requirements increase. The primary reason for this observation is that both the algorithms will discover many SPPs in any database if the $maxPer$ value continues to increase. (ii) SPP-ECLAT generates SPPs much faster than SPP-Growth under any given $maxPer$ in BMS-WebView-1, Retail, T10I4D100K, T20I6D100K, and Mushrooms databases. More importantly, we can also observe that at high $maxPer$ values, SPP-ECLAT algorithm generates the SPPs much faster than SPP-Growth algorithm. The reason is SPP-ECLAT using the downward closure property and the depth-first search technique, so the SPPs are generated by simply peforming intersection of SPP-list. The process is repeated to find all SPPs. (iii) With the BMS-WebView-2 dataset, which contains long transactions and many distinct items, SPP-ECLAT algorithm takes more time than SPP-Growth algorithm. It is because SPP-ECLAT algorithm is based on the downward closure property and the depth-first search technique, so it doesn't require scanning the database each time, but generating all SPPs it has to perform the intersection of the SPP-list. So the long SPP-List requires more time to repeat the intersection process.

Finally, we have shown the memory consumption details of SPP-Growth and SPP-ECLAT algorithms in Figure 2.8 by varying the value of $maxPer$. In detail, Figure 2.8(a) to Figure 2.8(c), shows the memory consumption of both the algorithms in the T10I4D100K database, respectively, for different $minSup$ and $maxLa$ values. Figure 2.8 presents the performance of the two algorithms in three cases, minSup = 0.5% and maxLa = 0.4% (Figure 2.8(a)), minSup = 0.8% and maxLa = 0.4% (Figure 2.8(b)), minSup = 1% and maxLa = 0.4% (Figure 2.8(c)). The results all show that SPP-ECLAT consumes less memory than SPP-Growth. In detail, the average memory consumption difference between the two algorithms is around 120MB. The more maxLa value increase, the more significant the difference memory consumption between the two algorithms, from 80MB (when maxLa value is 0.1%) to 170MB (when maxLa value is 0.4%).

From Figure 2.8(d) to Figure 2.8(f), shows the memory consumption of both the algorithms in the Retail database, respectively, for different $minSup$ and $maxLa$ values. Figure 2.8 presents the performance of the two algorithms in three cases, minSup = 0.5% and maxLa = 2% (Figure 2.8(d)), minSup = 0.8% and maxLa = 2% (Figure 2.8(e)), minSup = 1% and maxLa = 2% (Figure 2.8(f)). The results all show that SPP-ECLAT consumes less memory than SPP-Growth. The magnitude of the difference in memory consumption is around 70MB.

From Figure 2.8(g) to Figure 2.8(i), shows the memory consumption of both the algorithms in the T20I6D100K database, respectively, for different $minSup$ and $maxLa$ values. Figure 2.8 presents the performance of the two algorithms in three cases, min-

Sup = 3% and maxLa = 2% (Figure 2.8(g)), minSup = 3.5% and maxLa = 2% (Figure 2.8(h)), minSup = 4% and maxLa = 2% (Figure 2.8(i)). in these cases, SPP-ECLAT consumes less memory than SPP-Growth by 140MB on average.

From Figure 2.8(j) to Figure 2.8(l), shows the memory consumption of both the algorithms in the BMS-WebView-1 database, respectively, for different $minSup$ and $maxLa$ values. Figure 2.8 presents the performance of the two algorithms in three cases, minSup = 0.5% and maxLa = 2% (Figure 2.8(j)), minSup = 0.8% and maxLa = 2% (Figure 2.8(k)), minSup = 1% and maxLa = 2% (Figure 2.8(l)). The results show that SPP-ECLAT consumes less memory than SPP-Growth, around 8MB on average.

From Figure 2.8(m) to Figure 2.8(o), shows the memory consumption of both the algorithms in the BMS-WebView-2 database, respectively, for different $minSup$ and $maxLa$ values. Figure 2.8 presents the performance of the two algorithms in three cases, minSup = 0.6% and maxLa = 2% (Figure 2.8(m)), minSup = 0.8% and maxLa = 2% (Figure 2.8(n)), minSup = 1% and maxLa = 2% (Figure 2.8(o)). The results all show that SPP-ECLAT consumes less memory than SPP-Growth. In detail, the difference in memory consumption when maxLa = 2% is around 9MB, and when maxLa is increasing up to 4%, the difference in memory consumption of the two algorithms is around 10MB

From Figure 2.8(p) to Figure 2.8(r), shows the memory consumption of both the algorithms in the Mushrooms database, respectively, for different $minSup$ and $maxLa$ values.Figure 2.8 presents the performance of the two algorithms in three cases, minSup = 6% and maxLa = 3% (Figure 2.8(p)), minSup = 7% and maxLa = 3% (Figure 2.5(q)), minSup = 8% and maxLa = 3% (Figure 2.5(r)). In these cases, SPP-ECLAT consumes less than 56MB on average.

It can be observed that the SPP-ECLAT consumes relatively less memory than the SPP-Growth algorithm. The following are some noteworthy findings that can be derived from this figure: (i) If we increase the $maxPer$ value, then subsequently, both algorithms' memory consumption increase. The primary reason for this observation is that both the algorithms will discover many SPPs in any database if the $maxPer$ value continues to increase. (ii) SPP-ECLAT generates SPPs using a SPP-list structure, which helps reduce the search space on every database. (iii) It should highlight that in BMS-WebView-2 dataset. The processing time of the proposed algorithm for this dataset is comparable to the SPP-Growth algorithm; however, it is interesting to note that SPP-ECLAT algorithm requires much less memory in the memory consumption test than the SPP-Growth algorithm.

2.5.4 Experiment-3: Scalability Analysis

In this experiment, we have used the Kosarak database, a sparse real-world database, to perform the scalability operation. The scalability operation depends on the number of items and the number of records (e.g., transactions). Therefore, to analyze the complexity, we need to think about every operation an algorithm performs and how it is affected by the number of items and records. Thus, the sparse dataset has been chosen because when we divide the database into equal portions, each portion has a different number of items, so we will see clearly how the algorithm is doing. This scalability operation is utilized to discover the efficacy and productivity of the proposed algorithm on big columnar temporal databases. Therefore, in this experiment we divide the Kosarak database into five equal portions, each with 0.2 million transactions. We evaluate the performance of both the SPP-Growth and SPP-ECLAT algorithms, where

the database size is varied from 200000 to 1000000 transactions. Figure 2.9 shows the results in terms of the runtimes and memory consumption levels of both SPP-Growth and SPP-ECLAT algorithms under different database sizes when $maxLa = 0.04$ (in %), $minSup = 0.01$ (in %), and $maxPer = 0.05$ (in %). Some of the important observations that can be drawn from this figure are as follows. (i) If we keep increasing the database size, then both algorithms' runtimes and memory requirements will increase almost linearly. (ii) SPP-ECLAT consumes less runtime and memory than the SPP-Growth algorithm under any given database size.

2.6 Discussion

In this section, we have compared the time complexity analysis of both the algorithms. Let us consider a columnar temporal database containing 'p' number of distinct items and total number of transactions represented as 'q'. Let us assume that all the items in q are interesting, and every item is present in every transaction; i.e., the generated lists contain q entries each.

In the literature, SPP-Growth [46] is the only state-of-the-art algorithm that uses the concept of SPP-list constructions to generate complete SPPs. The major contributions of SPP-Growth and its complexities are as follows: First, the complete database is scanned, and the items in each transaction are stored as a prefix-tree. In the worst case, if every item is present in every transaction, then the time complexity for this operation is $O(p * q)$. Second, we construct the SPP-lists with a complexity of $O(p * q)$. After the initial prefix-tree construction, SPP-Growth recursively performs a depth-first search to find all the interesting patterns. The number of possible patterns is $n = 2^p - 1$. In real-world applications, the number of patterns considered depends on the database's characteristics and the algorithms' parameters. If $minSup$, $maxPer$, or $maxLa$ are increased, fewer patterns may be considered due to applying the search space pruning strategies. Finally, for each considered pattern Γ that extends an pattern Δ, SPP-Growth traverses the node-links of the SPP-list of Δ to create the conditional pattern base, SPP-list, and prefix-tree of Γ. This construction is done in linear time as these structures of Δ are traversed once. Therefore, the overall complexity of SPP-Growth is $O(p * q) + O(p * q * n) = O(p * q * n)$.

We complete the generation of SPPs using the SPP-ECLAT with the help of two algorithms. In Algorithm 1, we scan the complete database once to discover the one-length SPPs by constructing an SPP-list data structure. In the worst case, if every item is present in every transaction, then the time complexity for this operation is $O(p * q)$. In the Algorithm 2, we need to merge the TS-list elements of the two current length patterns to generate the higher length patterns. In the worst case, if every item is present in every transaction as the length of the TS-list of every pattern becomes q, then the time complexity for merging any of the two pattern's TS-lists becomes $O(q)$. This algorithm utilizes the Depth-First Search (DFS) strategy on the pattern lattice. The number of possible patterns is $n = 2^p - 1$. Therefore, the time complexity for generating all the possible interesting patterns is $O(n * q)$. The overall time complexity of SPP-ECLAT is $O(q + n * q) = O(q * n)$.

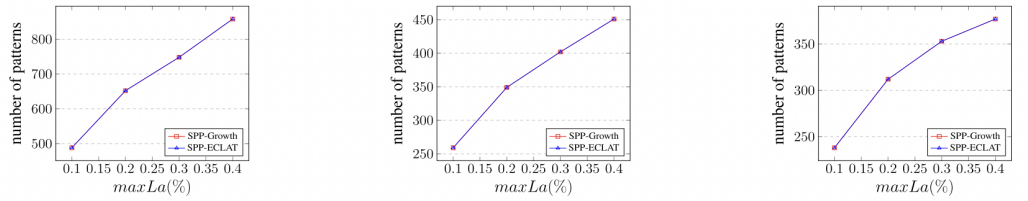
In real-world applications, the overall superiority of SPP-ECLAT ultimately depends on the actual values of the given parameters, such as p, q, and n. Therefore, we conducted rigorous experimentation on six real-world databases to demonstrate that

SPP-ECLAT outperforms the state-of-the-art SPP-Growth algorithm.

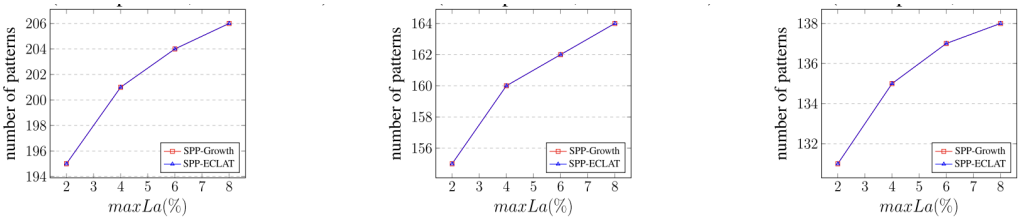
2.7 Conclusions and Future Work

This study has proposed an efficient and novel algorithm, called Stable Periodic-frequent Pattern – Equivalence Class Transformation(SPP-ECLAT), to discover stable periodic-frequent patterns. The output patterns of the algorithm not only satisfy the user-specified *minimum support* and *maximum periodicity* thresholds but also are stable patterns based on the user-specified *maximum lability* threshold in any big columnar temporal databases. The SPP-List structure of the SPP-ECLAT algorithm plays an important role in eliminating many patterns that are not considered to be candidate patterns from the huge search space. An in-depth examination of the proposed SPP-ECLAT approach on six synthetic and real-world databases revealed that its memory consumption and runtime are efficient and highly scalable relative to those of the state-of-the-art SPP-Growth algorithm.

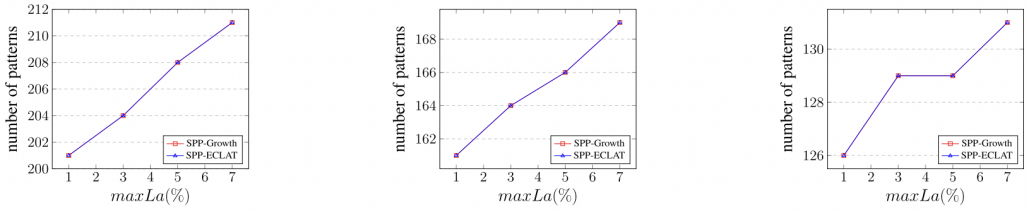
As for the future work, we will study the Lability concept over different types of patterns. It is also interesting to work on discovering stable periodic-frequent patterns in uncertain databases. Furthermore, we will focus on identifying SPPs in static temporal data, and it would be important to investigate stable patterns in graphs, data streams, and symbolic databases in the future.



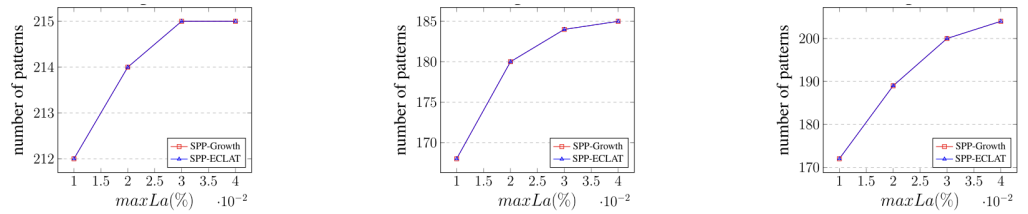
(a) **T10I4D100K**



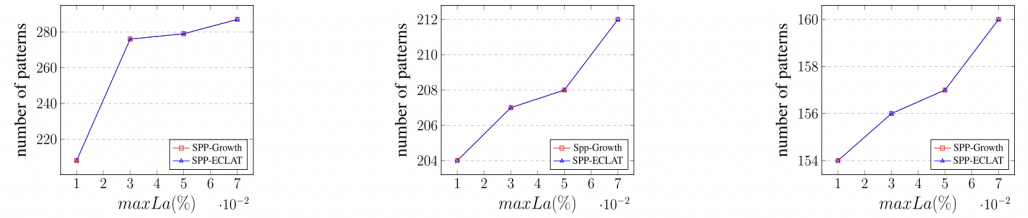
(b) **Retail**



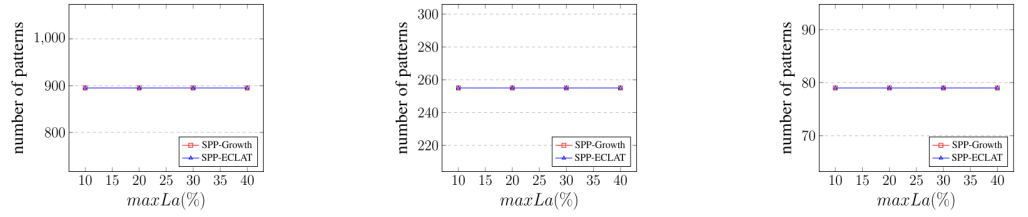
(c) **T20I6D100K**



(d) **BMS-WebView-1**

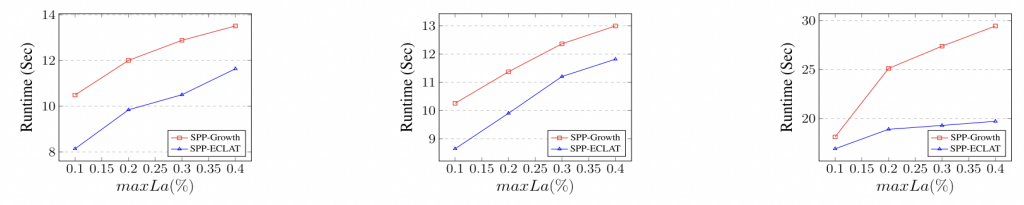


(e) **BMS-WebView-2**

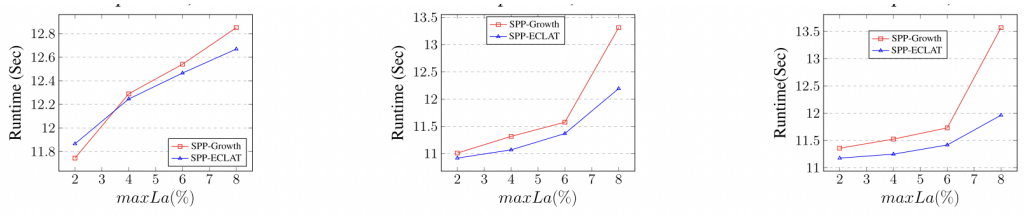


(f) **Mushrooms**

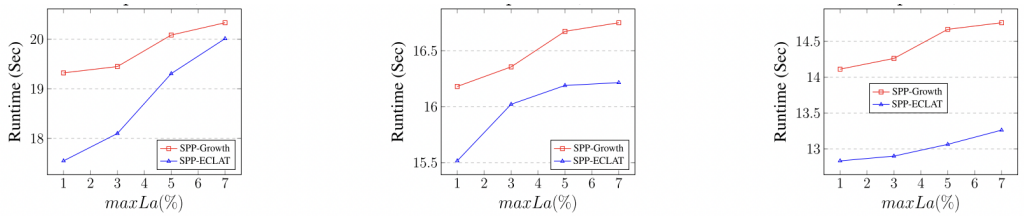
Figure 2.3: Number of stable periodic-frequent patterns generated in various databases by varying $minSup$ and $maxLa$ values



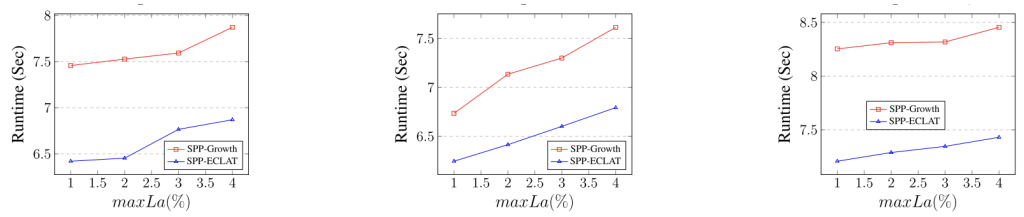
(a) **T10I4D100K**



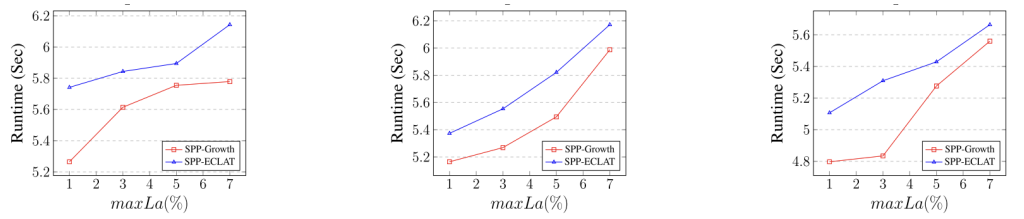
(b) **Retail**



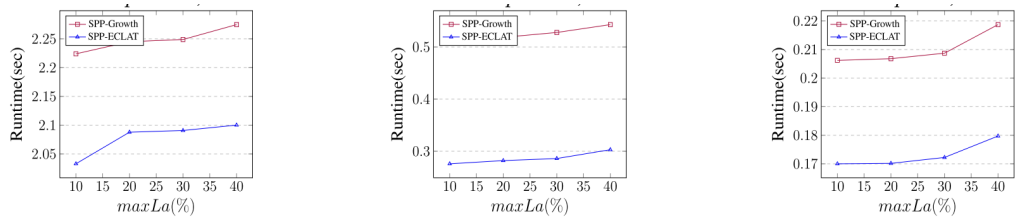
(c) **T20I6D100K**



(d) **BMS-WebView-1**

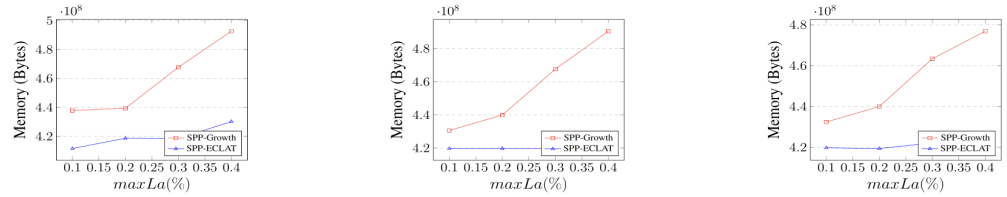


(e) **BMS-WebView-2**

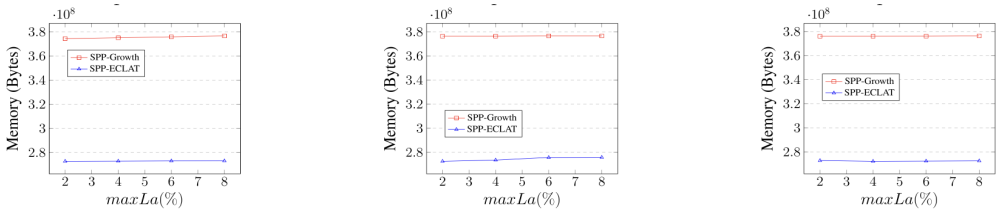


(f) **Mushrooms**

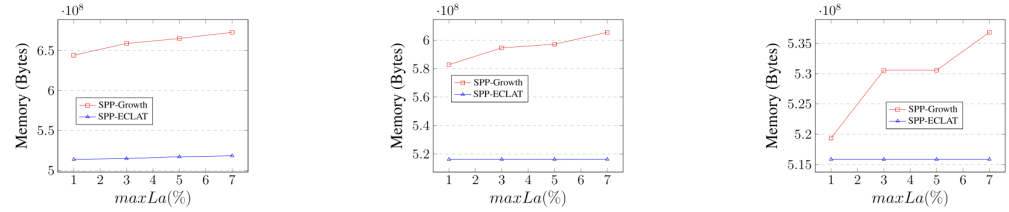
Figure 2.4: Runtime requirements of SPP-Growth and SPP-ECLAT algorithms at different $maxLa$



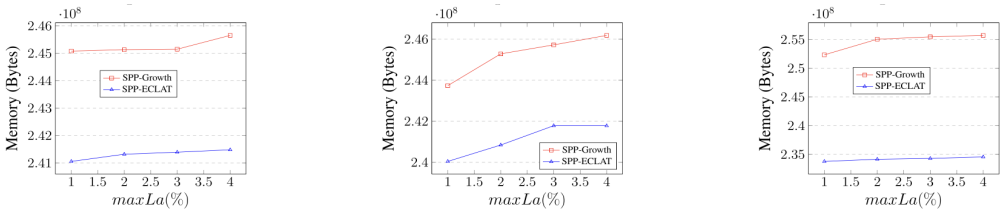
(a) **T10I4D100K**



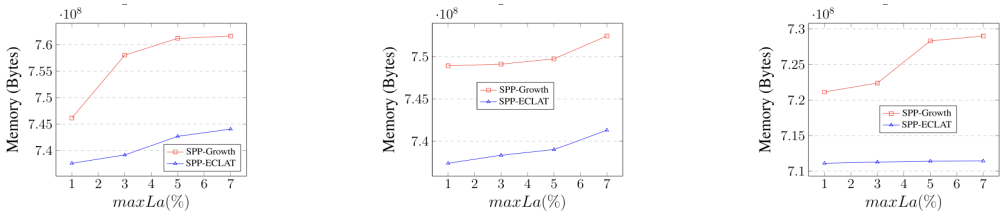
(b) **Retail**



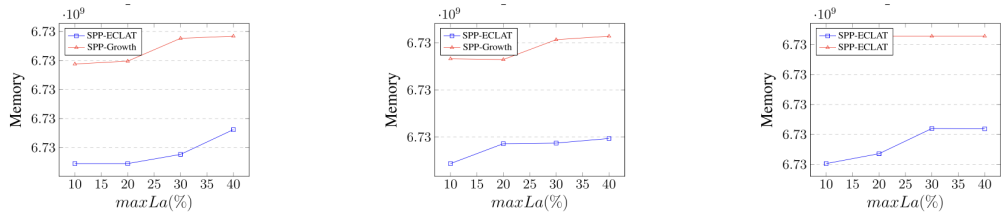
(c) **T20I6D100K**



(d) **BMS-WebView-1**

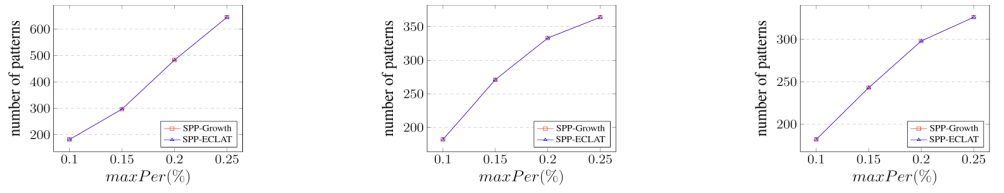


(e) **BMS-WebView-2**

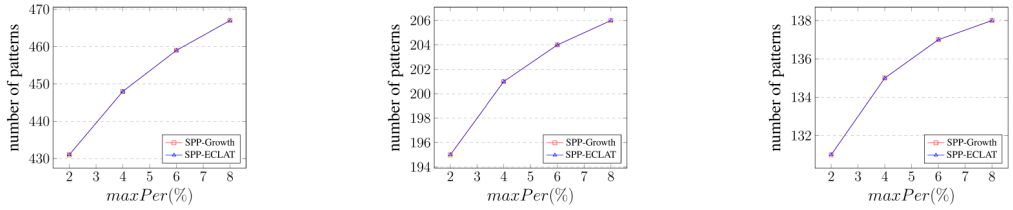


(f) **Mushrooms**

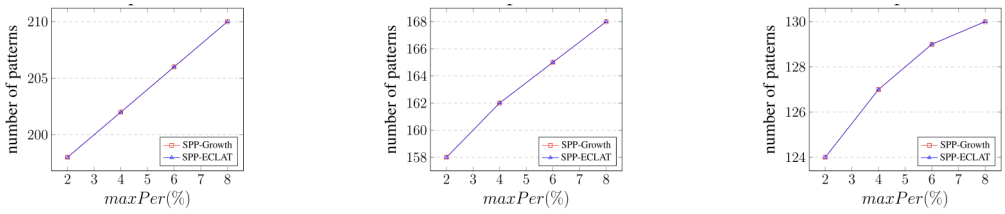
Figure 2.5: Memory consumption of SPP-Growth and SPP-ECLAT algorithms at different $maxLa$



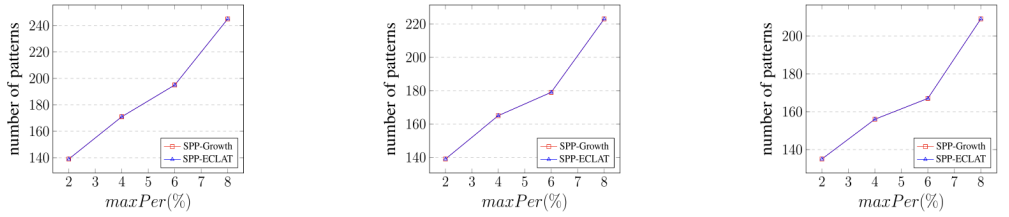
(a) **T10I4D100K**



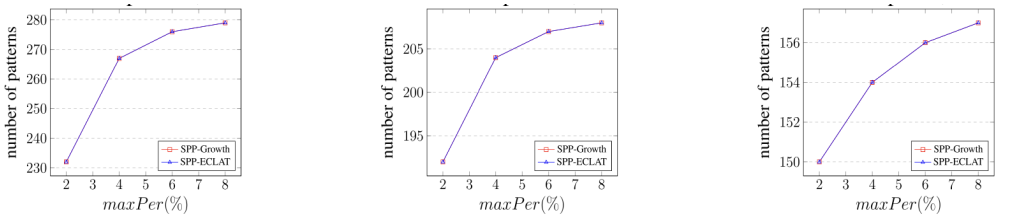
(b) **Retail**



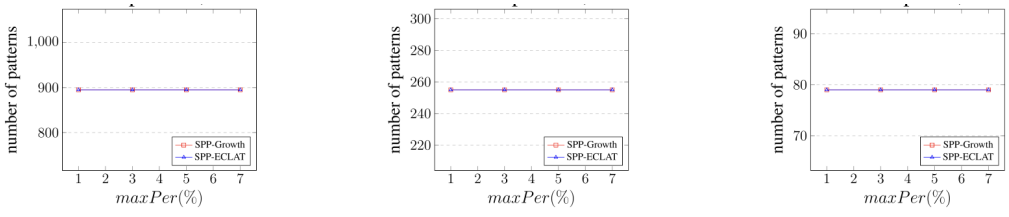
(c) **T20I6D100K**



(d) **BMS-WebView-1**

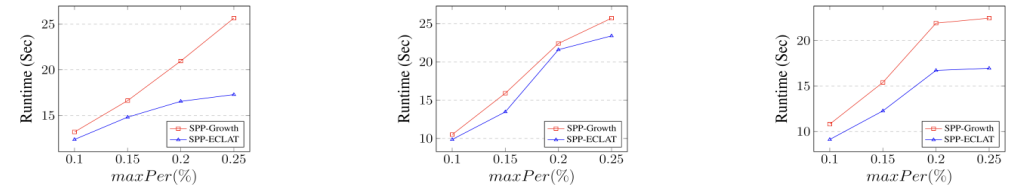


(e) **BMS-WebView-2**

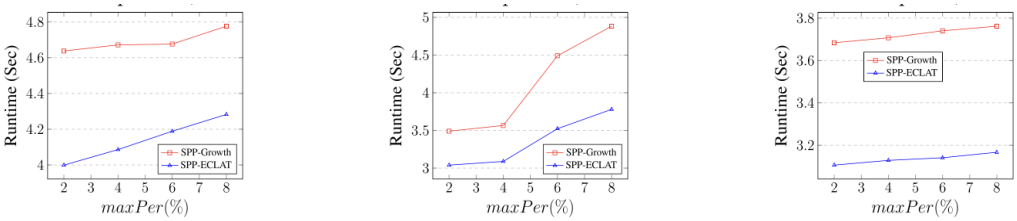


(f) **Mushrooms**

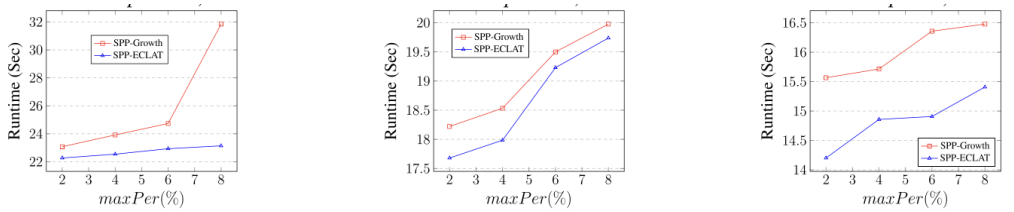
Figure 2.6: Number of stable periodic-frequent patterns generated in various databases by varying $minSup$ and $maxPer$



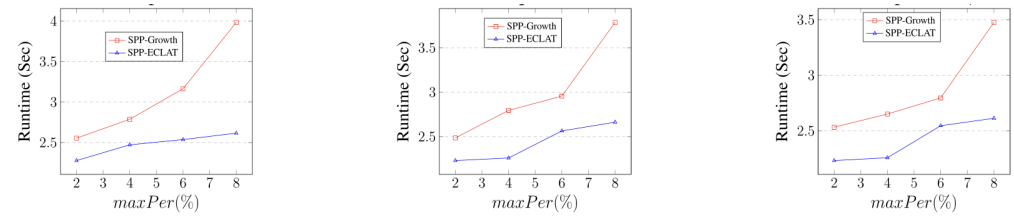
(a) **T10I4D100K**



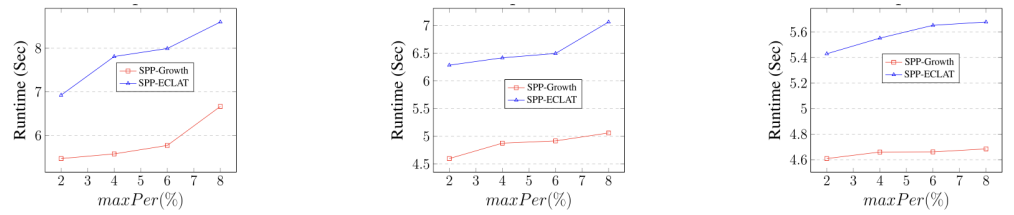
(b) **Retail**



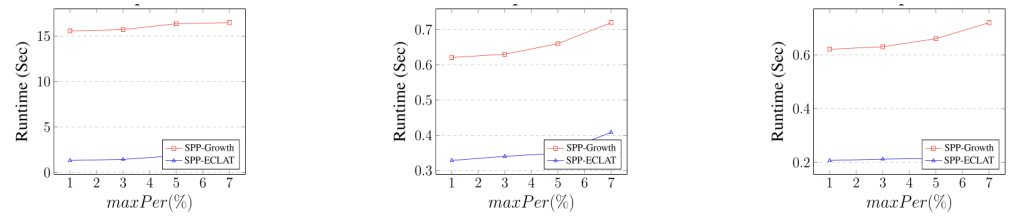
(c) **T20I6D100K**



(d) **BMS-WebView-1**

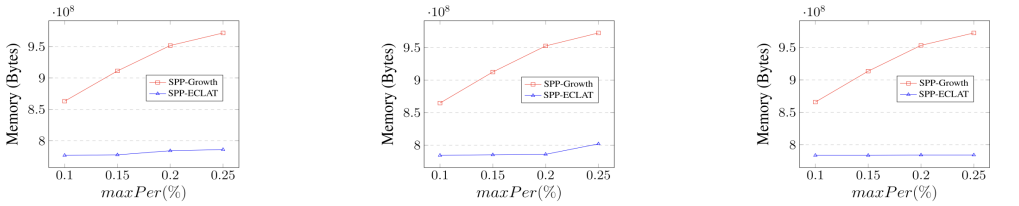


(e) **BMS-WebView-2**

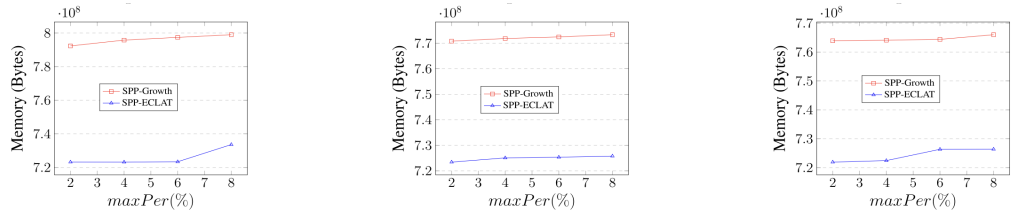


(f) **Mushrooms**

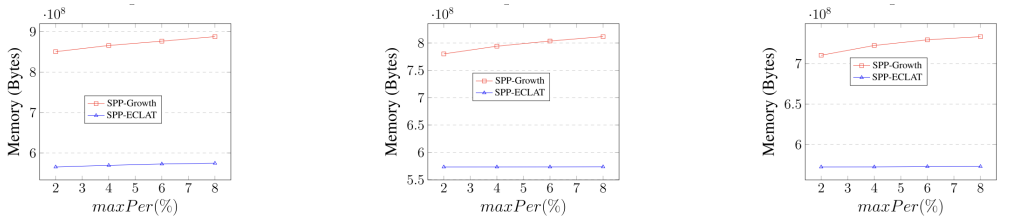
Figure 2.7: Runtime requirements of SPP-Growth and SPP-ECLAT algorithms at differnt $maxPer$



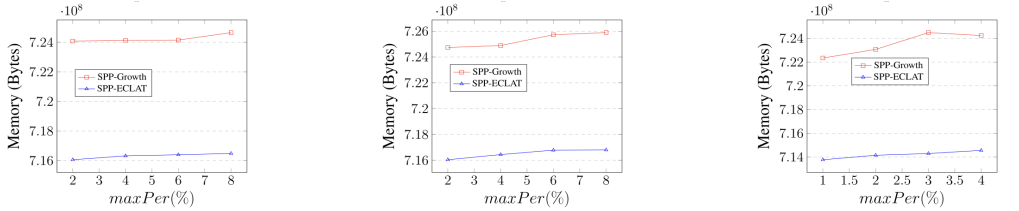
(a) **T10I4D100K**



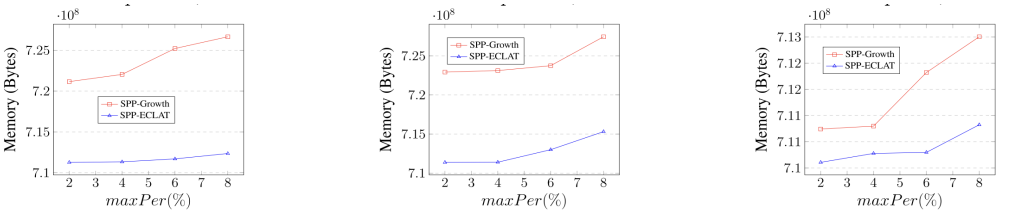
(b) **Retail**



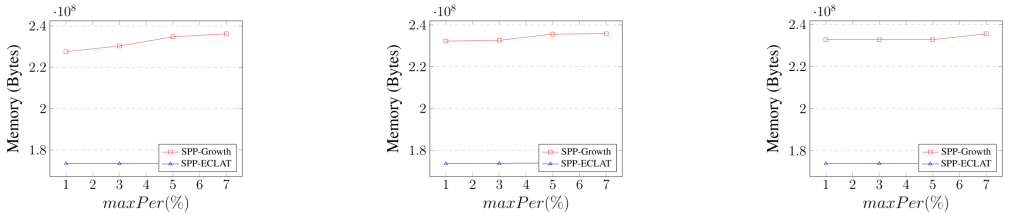
(c) **T20I6D100K**



(d) **BMS-WebView-1**



(e) **BMS-WebView-2**



(f) **Mushrooms**

Figure 2.8: Memory consumption of SPP-Growth and SPP-ECLAT algorithms at different $maxPer$

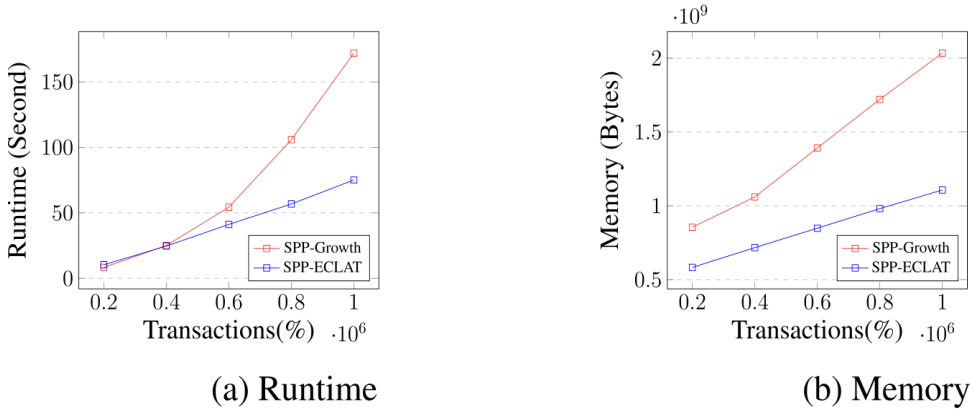


Figure 2.9: Scalability of the SPP-Growth and SPP-ECLAT algorithms

Chapter 3

A Deep Learning Approaches for Unstructured Medical Data

In a smart society, analyzing vast amounts of data using AI techniques has become instrumental in promoting health and well-being. By analyzing diverse data sources such as real-time physiological data, healthcare facility records, and treatment information, AI technologies enable early detection of illnesses, promote healthy living, and optimize treatment strategies. Additionally, integrating robotics in healthcare and caregiving helps alleviate the burden on medical professionals and enhances overall quality of life.

Deep learning (DL) has emerged as a powerful tool in various domains, including logistic supply chain management [77, 78] and smart manufacturing [79–81], due to its ability to handle large and complex datasets with minimal human intervention. DL plays a pivotal role in achieving higher service quality and improving patient outcomes in healthcare systems.

In this chapter, my thesis focuses on developing DL models tailored to address challenges associated with unstructured medical data, particularly medical images and text notes. The primary objectives of our research are twofold:

1. Solving the Problem of Limited Labeled Data: Limited availability of labeled data is a common challenge in the medical domain. Our DL models aim to overcome this limitation by leveraging transfer learning and data augmentation techniques to learn from smaller datasets effectively.

2. Improving Accuracy of Diagnosis and Image Analysis: By integrating multiple medical data modalities, including medical images and text notes, our DL models aim to improve the accuracy of diagnosis and image analysis. By jointly analyzing diverse data sources, we can extract more comprehensive insights and enhance decision-making processes in healthcare settings.

3.1 Transfer Learning for Medical Image Classification

Transfer learning is a key solution to deal with the problem of data scarcity, where the learning process leverages knowledge learned from similar tasks. In this study, we employ PubMedCLIP as a pre-trained model for medical image classification tasks. We evaluate multiple datasets from different body regions simultaneously. Moreover, we

investigate varying amounts of training data to evaluate the effectiveness of PubMedCLIP in cases of limited training data.

3.1.1 Introduction

Deep learning has emerged as the method of choice for medical image classification and segmentation, demonstrating exceptional performance across various medical imaging pathways such as breast [82], lesion [83], and chest and lung [84,85]. However, the effectiveness of deep learning models is often hindered by the limited availability of annotated data and the imbalance in data categories, posing challenges for training. Moreover, the scarcity of labeled medical images due to the intricate labeling process by experienced experts exacerbates the problem of data scarcity.

Transfer learning has emerged as a key solution to address these limitations by leveraging knowledge from related tasks. While many studies in transfer learning for medical image classification utilize pre-trained models such as ResNet [86] or Inception [87] on natural image datasets like ImageNet, the differences between ImageNet classification and medical image classification pose significant challenges.

Recently, OpenAI introduced the Contrastive Language-Image Pre-training (CLIP) model [88], demonstrating remarkable performance in various tasks, including image classification, by leveraging text-image associations in a multimodal framework. However, CLIP's generalization to medical image classification is limited as it needs domain-specific knowledge. To address this gap, Eslami et al. proposed PubMedCLIP [89], a fine-tuned version of CLIP trained on medical images and associated text from PubMed articles. PubMedCLIP enhances CLIP's performance in medical applications by incorporating domain-specific knowledge.

This study uses PubMedCLIP as a pre-trained model for transfer learning in medical image classification tasks. Unlike previous studies focusing on single datasets, we evaluate PubMedCLIP's performance across multiple datasets representing different body regions. Additionally, we investigate PubMedCLIP's effectiveness under varying amounts of training data to assess its robustness and scalability. This study contributes to the advancement of medical image classification by introducing PubMedCLIP as a powerful tool for transfer learning. We provide valuable insights into its potential applications in real-world healthcare settings by demonstrating PubMedCLIP's efficacy across diverse datasets and training data sizes. To the best of our knowledge, this is the first study to employ PubMedCLIP for transfer learning in medical image classification, paving the way for future research.

3.1.2 Transfer Learning Using PubmedCLIP

Machine learning is the process of learning using a model that supports different types of data modalities. In recent years, there has been significant progress in research on modalities of language and vision. Language (i.e., words, sentences. etc.) and visual information (i.e., images, videos) could be jointly exploited to improve model robustness. In medical images, more modalities can be defined such as X-ray, magnetic magnetic resonance imaging (MRI), computerized tomography (CT), histopathologic scan, and ultrasound images [90].

Besides, transfer learning has become an essential method for deep learning applications for medical images. Since the datasets in the medical field are small, that will

lead to the overfitting problem for deep learning models. To deal with a small amount of data, a model which is previously trained on big general datasets such as ImageNet will be employed as the initial (pre-trained) model. Then medical datasets are used to fine-tune either the feature extractor or some layers of the model so that this model can adapt to medical datasets.

Training in original CLIP aims to find which text vectors are more similar to a given image vector. This process is called contrastive learning. CLIP was trained on a large real image dataset of over 400 million image-text pairs sampled from the internet with nearly zero additional human annotation. However, CLIP's limitation is that it does not cover everything and needs to be adapted to specialized domains.

To adapt the CLIP model to a medical domain, the PubMedCLIP was proposed in [89] by fine-tuning CLIP. This approach is helpful for the medical domain as the data annotation requires expert knowledge and time. Specifically, the contrastive language-supervision objective is used with image-text pairs of medical images from diverse body regions. PubMedCLIP was trained on a big medical ROCO dataset [91]. During training, the model encodes the image input and the text input independently with a CLIP image encoder and CLIP text encoder. Both get their inputs and extract the feature representation of their modality. After that, these feature vectors are adapted to have the same size. Then the scaled pairwise cosine similarity of the two vectors are measured and the loss function is calculated. Finally, the values of the weights are updated.

In this work, we studied transfer learning with pre- trained model PubMedClip on the well-known medical datasets MedMnist [90]. Our model used PubMedClip as the feature extractor of a medical image, and then the output vector was fed to two fully connected layers for the classification task. Previous studies on transfer learning only focus on a single type of disease, for example, the identification of eye disease [92], or early detection of Alzheimer's disease [93]. In this study, we focus on multiple datasets of diverse diseases from MedMnist [90]. This is a large- scale MNIST-like collection of standardized biomedical images. This collection includes various datasets of 2D medical images with the corresponding classification labels. It covers primary data modalities in biomedical images, such as CT, X-ray, MRI, histopathologic scan, and ultrasound. In addition, we also evaluate the performance of our transfer learning model in terms of the amount of training data on different datasets.

3.1.3 Proposed Framework

Our transfer learning model is designed to address the challenge of classification across multiple medical image datasets, each representing different diseases. Leveraging the power of transfer learning, our model extracts relevant features from input images across various datasets and utilizes this knowledge to make accurate predictions, even when training data is limited. The flowchart of our model is shown in Fig.3.1

Input Layer: The model commences with an input layer, where medical images from individual datasets are sequentially fed into the system. Each dataset is processed separately, ensuring focused attention on the unique characteristics and features of the images within that specific dataset. This sequential input mechanism allows the model to effectively adapt to the nuances of each dataset during the training phase, enhancing its ability to extract relevant features and make accurate predictions.

Feature Extraction with PubMedClip: The input images are processed through a feature extraction layer utilizing PubMedClip. This sophisticated technique enables

the extraction of discriminative features from medical images, crucial for subsequent classification tasks.

Feature Vector: The output of the feature extraction layer is a compact and informative feature vector representing each input image. These feature vectors encapsulate essential information about the image, learned through the feature extraction process.

Fully Connected Layers: The feature vectors are then passed through two fully connected layers. These layers serve as the backbone of our classification model, leveraging the extracted features to learn complex relationships and patterns within the data.

Prediction: Finally, the model generates predictions based on the learned features, classifying each input image into its respective disease category

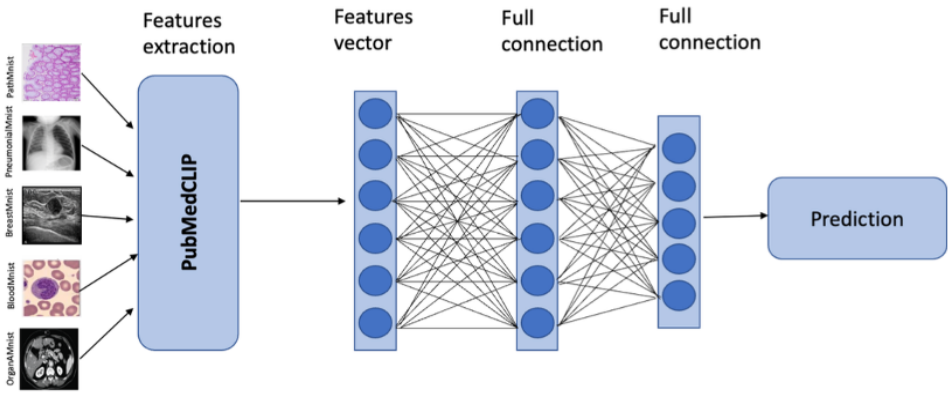


Figure 3.1: Illustration of the Pre-Trained PubMedClip Model Employed for Multi-Modality Medical Image Datasets

3.1.4 Experiments

A. Dataset Description

Our experiments utilize the MedMNIST datasets, curated explicitly for medical image classification across various data scales and task complexities. These datasets offer a diverse array of medical images, ranging from 100 to 100,000 samples, and encompass various classification tasks, including binary/multi-class, ordinal regression, and multi-label.

We selected five representative datasets from the MedMNIST collection, each focusing on distinct diseases and imaging modalities: *PathMnist Dataset*: consists of 100,000 histological image patches from colorectal cancer tissue, categorized into nine tissue types for a multi-class classification task. *PneumoniaMnist Dataset*: Comprises pediatric chest X-ray images for binary-class classification (pneumonia vs. normal) *BreastMnist Dataset*: includes breast ultrasound images categorized into benign, malignant, and normal classes. *BloodMnist Dataset*: contains images of individual blood cells classified into eight types for multi-class classification. *OrganAMnist Dataset*: showcases CT images of 11 body organs from axial, coronal, and sagittal views for multi-class classification.

Each dataset is methodically divided into train, validation, and test sets, facilitating rigorous evaluation of model performance across different phases of development. The characteristics of these datasets, including class distributions and imaging modalities, are summarized in Table3.1 for reference.

Table 3.1: Statistics of the datasets

Dataset	No of classes	Train set	Val set	Test set
Path Mnist	9	89996	10004	7180
Pneumonia Mnist	2	4708	524	624
Breast Mnist	3	546	78	156
Blood Mnist	8	11959	1712	3421
OrganA Mnist	11	34581	6491	17778

B. Evaluation Strategy

The objective of this experiment is to evaluate the performance of our transfer learning model using PubMedCLIP and compare it with the performance of using the pre-trained models CLIP and PubMedCLIP without additional training. The goal is to determine how well our transfer learning model generalizes across different medical imaging datasets and to see if fine-tuning with domain-specific data enhances performance. To evaluate the performance of our model across different datasets and varying amounts of training data, we employ a systematic evaluation strategy:

Data Selection: We randomly select a subset of images from each dataset for training and testing purposes. This ensures representation from each class while maintaining diversity within the datasets.

Training and Testing Split:

Training Data: We consider three different numbers of training samples per class: 50, 100, and 500 images. This variation allows us to observe the impact of training data size on model performance.

Testing Data: We utilize the remaining images from each dataset for testing, ensuring comprehensive evaluation across all classes.

Datasets and Class Distribution:

PathMnist (9 classes): 900 images for training, 7,180 images for testing. PneumoniaMnist (2 classes): 200 images for training, 624 images for testing. BreastMnist (3 classes): 300 images for training, 156 images for testing. BloodMnist (8 classes): 800 images for training, 3,421 images for testing. OrganAMnist (11 classes): 1100 images for training, 17,778 images for testing.

C. Evaluation Results

In this section, we conduct a series of experiments to validate the model on different modalities of medical datasets. To evaluate the performance of our transfer learning model, we compare the results with those of CLIP and PubMedCLIP. Figure 3.2 shows bar charts of the models' performance on five datasets. The results show that our transfer learning model using pre-trained PubMedCLIP performs exceptionally well across different data modalities and tasks (i.e., binary and multi-class classifications). Overall, the model achieves an accuracy above 75% in all datasets. Meanwhile, CLIP shows

Figure 3.2: Performance of the model on all datasets

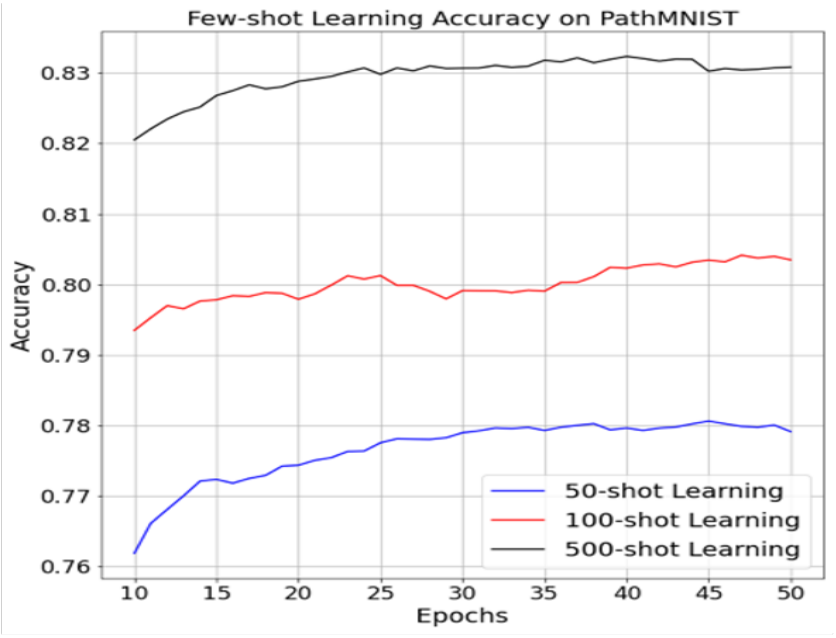


Figure 3.3: Dependence of learning performance on the number of training samples using PathMNIST dataset

Table 3.2: Performance Metrics

Class	**Precision**	**Recall**	**F1-score**
Adipose	0.967	0.958	0.963
Background	0.987	0.909	0.946
Debris	0.439	0.502	0.469
Lymphocytes	0.962	0.907	0.934
Mucus	0.862	0.905	0.883
Smooth muscle	0.609	0.555	0.581
Normal colon mucosa	0.823	0.862	0.842
Cancer-associated stroma	0.450	0.441	0.445
Colorectal adenocarcinoma epithelial	0.728	0.874	0.795

the best performance on the Breast dataset with an accuracy of 62.82%, but its performance is very low on other datasets. PubMedCLIP is similar to CLIP; even though PubMedCLIP is a fine-tuned version of CLIP trained on the large ROCO dataset, it only shows good performance on the Breast dataset (with an accuracy of 73.76%) and on the Pneumonia dataset (with an accuracy of 43.42%), but performs poorly on the other datasets.

Next, we examine in detail how our transfer learning model performs on each dataset. The performance of our model varies across different datasets. For example, the model's accuracy attains 75.64% on the BreastMNIST dataset, 78.25% on the OrganAMNIST dataset, 79.36% on the PathMNIST dataset, 80.32% on the BloodMNIST dataset, and 85.26% on the PneumoniaMNIST dataset. It can be observed that, with the same number of sample images of each class fed into the training model, the performance also depends on the number of classes in a dataset. The highest accuracy values are observed on the PneumoniaMNIST dataset, which only contains two classes: "normal" and "pneumonia." However, in the OrganAMNIST dataset, which contains 11 classes, the accuracy is nearly 6% lower than that of the PneumoniaMNIST dataset.

Furthermore, the results show that the test accuracy also varies across the modality types of images. For example, the test accuracy on ultrasound images (BreastMNIST dataset) and histopathologic scan images (PathMNIST and BloodMNIST datasets) ranges from 75% to 80%. However, the test accuracy on X-ray images (PneumoniaMNIST dataset) is more than 5% higher.

Next, we evaluate the performances with different amounts of training data for the PathMNIST dataset, which includes nine classes. Figure3.3 shows three learning curves representing 50 images, 100 images, and 500 images per class, respectively. Generally, the accuracy increases when more samples are fed into the training model. For example, doubling the size of data from 50 to 100 images per class increases the accuracy by 2%. However, we need to scale the data five times (from 100 to 500 images) to attain the same increase.

To see how the model works inside one dataset, we explored the confusion matrix measurement (Fig.3.4 and its statistics (Table 3.2) of the multi-class PathMnist dataset as an example. The dataset includes nine class labels namely adipose, background, debris,lymphocytes, mucus, smooth muscle, normal colon mucosa, cancer-associated stroma, and colorectal adenocarcinoma epithelial. Note that the high values of Precision-Recall and F1- scores (close to 1) imply that the model performs well in returning the actual label for each image class in the dataset. Table 3.2 indicates

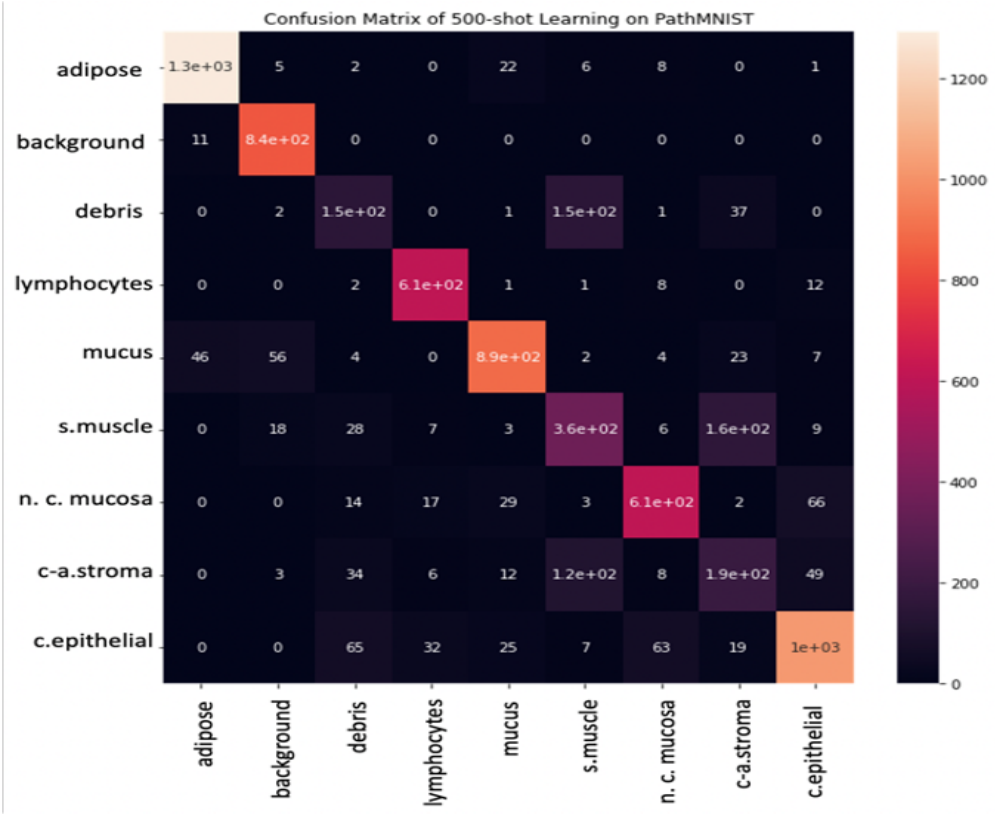


Figure 3.4: Confusion matrix on PathMnist dataset

the model's effectiveness in most classes, except debris, smooth muscle, and cancer-associated stroma. This problem is because there is a large number of images being misclassified among the three classes, as shown in Figure 4. For example, cancer-associated stroma and debris symptom are often classified as smooth muscle symptom. These findings show a limitation of the discrimination capability of the model on highly similar classes.

3.1.5 Conclusions and Future Work

In this study, we explored transfer learning using the PubMedCLIP model as a pre-trained model for image classification across multiple datasets of different modalities.

Our evaluation revealed that the PubMedCLIP model demonstrated strong performance across various datasets. Notably, we found that even small amounts of labeled data (e.g., 100 images per class) yielded promising results, offering potential cost savings in data labeling efforts. However, our in-depth analysis highlighted instances where the model performed suboptimally, underscoring the need for further refinement. Specifically, while feature extraction proved effective for most classes, challenges arose in some instances, indicating the limitations of relying solely on this approach in the medical domain. Moving forward, our future work will focus on enhancing the model to address these challenges, potentially incorporating additional strategies to handle limited data and diverse modalities better.

3.2 A Multimodal Transfer Learning for Medical Image Classification

In machine learning for healthcare, medical image data often suffers from data scarcity and expensive annotation processes. In this study, to develop a robust medical image classification model, we propose a novel approach that leverages the multimodal pre-trained PubMedCLIP model as a backbone. By integrating medical image and text information and utilizing the power of LMM pre-trained models, our method provides a promising solution to address the challenges posed by limited medical image data.

3.2.1 Introduction

Deep learning (DL) is a powerful technique that facilitates significant advancements in medical image analysis [94–96]. However, training DL models can be challenging, especially when faced with limited data in the medical domain.

To address the issue of data scarcity, transfer learning (TL) has been introduced [97]. TL involves transferring pre-trained knowledge from a source task to a similar task. This approach not only reduces training time [98], but also proves beneficial when the target task lacks training data [99, 100]. TL can be applied using two main approaches: (i) utilizing a pre-trained network as a feature extractor and training a new classifier using the extracted features [101, 102], or (ii) fine-tuning the pre-trained network to suit the new task requirements [103].

In the medical domain, TL has been widely employed in medical image classification, addressing the limited availability of labeled medical image datasets. TL has been shown to enhance the performance of DL models for tasks such as breast cancer

classification [104, 105], lung nodule classification [106, 107], and brain tumor classification [108], while reducing the need for extensive labeled data during training. However, despite its successes, applying TL to medical image classification remains challenging. Medical images possess unique characteristics which make it non-trivial to apply pre-trained models. Furthermore, previous studies on TL in medical image classification have primarily focused on a specific case (or a single dataset), for example digital pathology image analysis [109]. The transferability of pre-trained models across different medical image datasets and tasks requires further investigation.

One highly promising pre-trained model for transfer learning is the Contrastive Language-Image Pretraining (CLIP) model, introduced by OpenAI in 2021 [88]. CLIP stands as a state-of-the-art model that establishes associations between images and text through extensive training on a diverse collection of image-text pairs. However, it is worth noting that while the CLIP approach performs admirably in general data domains, it was initially trained on publicly available internet data. Consequently, it lacks domain-specific knowledge, particularly in specialized fields like medicine. To address this limitation, Eslami et al. introduced PubMedCLIP [89], a fine-tuned adaptation of CLIP tailored for the medical domain. Their study revealed that leveraging the pre-trained PubMedCLIP features enhances visual question-answering (VQA) performance, surpassing current state-of-the-art baseline models.

In this work, we propose a model that takes advantage of PubMedCLIP's image and text feature representations. The robust visual-language representations allow our model to handle cases with limited training data. Experimental results demonstrate that the proposed multimodal model achieves excellent results in classifying medical images from different datasets. This paper is an extended version of our previous work [110]. Compared to [110], the main extensions are as follows. First, multiple prompts of different complexities are considered. Interestingly, it is shown that a richer prompt leads to much higher gains in classification accuracy. Second, a better feature fusion method is employed to further improve the performance. Third, two more datasets are used and more experiments are carried out, resulting in many insights into the behaviors of the model and reference methods.

The remainder of this study is organized as follows. Section 3.2.2 presents related work on transfer learning and multimodal learning models. Section 3.2.3 describes the proposed approach and experimental setup. Extensive experimental results and discussions are provided in Section 3.2.4. Finally, conclusions are given in Section 3.2.7

3.2.2 Related work

In this section, we review previous work related to TL in medical image classification, including multimodal models and the applications of pre-trained models.

A. Transfer Learning in Medical Image Classification

Transfer learning has been employed in medical image classification to enhance model performance, particularly when training data is limited. This approach enables models to leverage knowledge of a pre-trained model learned on large datasets to improve the performance on smaller, domain-specific datasets. This saves time and costs, which is crucial in the medical imaging domain where datasets can be relatively small. Previous work related to TL in medical image classification can be categorized as fol-

lows. (i) Feature extraction: A common approach is to use a pre-trained model such as VGG [111], MobileNet [112], DenseNet [113], or EfficientNet [114] as the feature extractors and then train a classifier on top of the extracted features. This approach has been shown to improve classification accuracy in various cases [115–117]. (ii) Fine-tuning a pre-trained model: This approach involves adapting a pre-trained model specifically for the medical image classification task. The parameters of a pre-trained model are updated by training on the target dataset. Fine-tuning has proven to be effective in medical image classification tasks, such as colonoscopy frame classification [118, 119]. (iii) Multi-task learning: This approach involves training a model simultaneously on multiple related tasks. In medical image classification, multi-task learning has been used to improve the accuracy of models by leveraging the relationship between different medical imaging tasks [120,121]. (iv) Domain adaptation: Domain adaptation in TL involves adapting a model trained on a source domain to a target domain with different distributions. In medical image classification, this approach has been used to address the problem of data imbalance and improve model performance on specific target domains [122]. TL has shown practicality in improving the performance of medical image classification models. However, these techniques result in high computational costs as discussed in [106, 123]. Besides, not all pre-trained models that have been trained on large-scale natural image datasets perform optimally across all medical image modalities. For instance, a review paper by Morid et al. [124] highlighted that Inception models were commonly utilized in analyzing X-rays, endoscopic images, and ultrasound images, while GoogLeNet and AlexNet were frequently employed for MRI analysis. On the other hand, VGGNet models were mostly used in studying skin lesions, fundus images, and OCT (optical coherence tomography) data.

Table 3.3: Recent transfer learning studies on medical images

Reference	**Year**	**Pre-trained model**	**Image type**	**Note**
[125]	2020	MobileNet	Path	Unimodal (Image)
[126]	2021	MobileNet	Breast	Unimodal (Image)
[127]	2021	DenseNet	Path	Unimodal (Image)
[128]	2021	DenseNet	Breast	Unimodal (Image)
[129]	2021	EfficientNet	Path	Unimodal (Image)
[130]	2022	DenseNet	Blood	Unimodal (Image)
[131]	2022	EfficientNet	Blood	Unimodal (Image)
[117]	2022	PubMedCLIP	Various image types of MedMNIST (Path; Pneumonia; Blood; Breast)	Unimodal (Image)
[132]	2023	EfficientNet	Breast	Unimodal (Image)
[110]	2023	PubMedCLIP	Breast	Multimodal (Image + text) No Prompt engineering
This study	-	PubMedCLIP	Path; Blood; Breast	Multimodal (Image + text) With Prompt engineering

Recently, more advanced pre-trained models have been investigated (see Table 3.3).

In [127], Ohata et al. considered 18 different image encoders in transfer learning for Path images. They showed that the best result of the experiment was provided by the DenseNet. In the research of Jimenez et al. [128] on breast tumor classification, DenseNet also demonstrated high accuracy in diagnosing benign and malignant tumors when compared with different pre-trained models. Similarly, Sharma et al. [130] employed DenseNet model with preprocessing techniques like normalization and data augmentation for Blood images. Meanwhile, in the study of Shaban et al. [125], they demonstrated that MobileNet exhibits superior performance, achieving the highest average accuracy compared to various classifiers on Path images. Also, Eroglu et al. [126] found that the highest accuracy was obtained with MobileNet features for Breast images. Kallipolitis et al. [129] utilized transfer learning with various pre-trained models on a dataset that is augmented by the Grad-CAM technique to highlight visual patterns relevant to each class. The experimental results showed that EfficientNet outperformed other models. In the study of Chola et al. [131], they employed EfficientNet as the backbone for Blood images which are pre-processed by image processing. In a comparison of different deep learning models for mamography breast images, Jafari et al. [132] demonstrated that among the individual models, EfficientNet consistently outperformed the others. Our study in [117] was the first to employ PubMedCLIP for medical image classification on various image types of MedMNIST dataset. However, the that solution is still unimodal, relying solely on image modality.

B. Multimodal Learning

In recent years, there has been increasing interest in using both text and image data as input for medical image analysis. Combining these two modalities allows for capturing both visual and semantic information, leading to improved accuracy and interpretability of classification results. Several recent studies have utilized medical reports to provide supervision information and learn multimodal representations by maximizing mutual information between the two input modalities [133–135]. Extracting labels from reports using natural language processing (NLP) has also been explored as a means to leverage information from the text [136, 137]. Transformer-based vision-and-language models are used for learning multimodal representations from image and associated reports, which outperform traditional CNN and RNN methods [138]. Attention mechanism have also been used to facilitate interactions between visual and semantic information [139]. Recently, Contrastive Language-Image Pretraining (CLIP) is an advanced pretrained model developed by OpenAI [88]. It applies contrastive learning with a huge dataset of 400 million image-text pairs obtained from the Internet. As a result, CLIP could be employed to retrieve the best matched image given a text and vice versa. One of the interesting advantages of CLIP is its ability to perform zero-shot learning [88]. Also, the high performance of CLIP features enables many new exciting applications, for example, pre-training model to address the challenge of limited labeled data [140], art classification [141], and image captioning [142]. In the medical domain, Eslami et al. [89] investigates the effectiveness of the pre-trained CLIP model for visual question answering (VQA) task. To tailor the CLIP model for applications in the medical field, the authors introduced the PubMedCLIP model by fine-tuning the original CLIP model. This approach employs pairs of medical images and associated text of various anatomical regions from the medical ROCO dataset [91].

In line with the new trend of using LMM in machine learning, our preliminary

work [110] introduced the first multimodal transfer learning approach using PubMedCLIP, where text and image features are combined for classifying Breast images. In this chapter's study, we present an extended solution with a new fusion method and prompt engineering. As a results, the proposed method can works with a small number of data samples and have good performance over different datasets.

3.2.3 Methodology

A. The Proposed Multimodal Model

As mentioned, our method aims to utilize the powerful multimodal representations of the PubMedCLIP. The method takes as input both an image and a description text. First, the image and text are encoded using PubMedCLIP, which produces a vector representation for each modality. These vector representations are then fed into a fusion module to produce a combined feature vector, which is used to predict a similarity score. Finally, the similarity scores are employed for classification. The proposed model con-

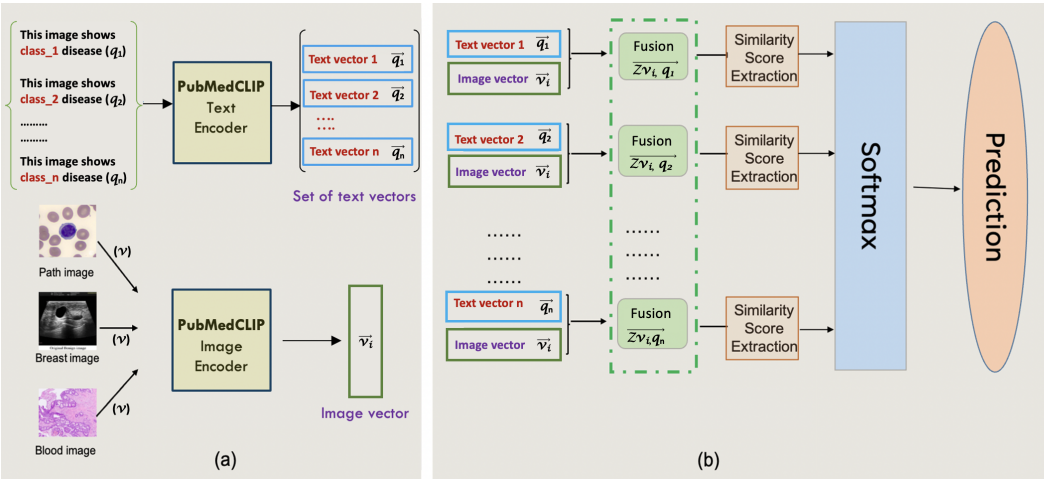


Figure 3.5: Overview of our model. We feed the original image and label templates to the PubMedCLIP-text encoder and PubMedCLIP-Image encoder. Fusion technique MFB is used to combine the two vectors. Finally, the softmax layer is added for classification the disease.

sists of three main stages: feature extraction, feature fusion, and class prediction. As shown in Figure 4.2, in the first stage, the part of image feature extraction provides an image feature vector $\vec{v_i}$ for the input image v. Similarly, the text feature extraction takes as input a text description of image class q_j and outputs a text feature vector $\vec{q_j}$. For each pair of ($\vec{v_i}$, $\vec{q_j}$), the feature fusion component produces a combined vector $\vec{Z}_{v_i,q_j}$ which is used to compute the similarity score between the image and the text. We perform image feature extraction with two options, PubMedCLIP-RN50 and PubMedCLIP-ViT32. These two encoders are based on different technologies, namely CNN (PubMedCLIP-RN50) and Vision Transformer (PubMedCLIP-ViT32). This helps to see behaviors of CNN and Vision Transformer over different medical imaging modes in our study, including microscopic imaging and ultrasound scan imaging.

In our approach to effectively utilize image labels for model training, we draw inspiration from the methodology described in Radford et al.'s paper [88]. This approach

acknowledges the importance of connecting text prompts with image content, a technique that has demonstrated enhanced performance compared to using simple labels alone [88]. In particular, it is shown that adding a simple word like "image" into the prompt can improve the performance. So, in this work, we consider a heuristic approach that gradually increases the contextual information in the prompt templates. The words we select for the prompts are commonly found in electrical health record (EHR), such as medical, image, disease, illness, symptom, sign, patient [143]. For each dataset, we have developed three distinct text prompt templates to guide the proposed model in the task of medical image classification. In addition to these prompts, we also include Prompt-0, which is simply the name of the label for each class. Specifically, the prompt templates are as follows.

1. Prompt-0: "{label}"
2. Prompt-1: "This image shows {label} disease".
3. Prompt-2: "In this medical image, there are indications of {label}".
4. Prompt-3: "Based on this medical image, it appears that the patient may be exhibiting signs or symptoms related to the {label} disease or illness".

As can be seen, these prompts offer varying levels of information, allowing the model to capture different aspects of the image. Specifically, Prompt-0 does not provide any additional context about the image, while more information is increasingly added to Prompt-1, Prompt-2, and Prompt-3. To facilitate this process, each dataset has a dictionary with descriptions of all the diseases present. These descriptions are encoded into text vectors, resulting in a set of text vectors specific to each dataset.

In the second stage, we combine the image and text features into a single feature vector using the feature fusion block. A straightforward approach for combining feature vectors is to multiply them element-wise. However, this method has limitations due to the simple combination of the two vectors. Various fusion techniques have been developed to combine text and image feature vectors to maximize interactions. These approaches usually rely on the idea of making bilinear pooling computationally feasible. In this study, we employs the Multimodal Factorized Bilinear Pooling (MFB) method [144] for multimodal feature fusion because of its simplicity, ease of implementation, and a high convergence rate. MFB [144] is a pooling method that combines information from multiple modalities (e.g., image and text) by computing the outer product of their feature vectors and then factorizing the resulting matrix using a low-rank decomposition. This approach allows for efficient modeling of pairwise interactions between different modalities while reducing the feature dimensionality after pooling [145, 146]. A comparison of MFB with other fusion methods will be discussed in the next section. In the third stage, class prediction is done based on combined feature vectors. Given a set of combined vectors $\{\vec{Z}_{v_i,q_j}\}$ for each pair of $(\vec{v_i}, \vec{q_j})$, we employed a set of fully-connected layer blocks, each of which independently transforms $\vec{Z}_{v_i,q_j}$ to a scalar. These output scalar values will form the similarity scores between the image $\vec{v_i}$ and the text description $\vec{q_j}$. The blocks are denoted as Similarity Score Extraction modules in Figure 3.5. Finally, a softmax layer normalizes the scores, yielding a probability distribution indicating the likelihood of the input image belonging to a description from the dictionary. The prediction is chosen by selecting the highest probability element from the distribution.

B. Datasets

To conduct this research, we use three different medical datasets with different classes and imaging modes. The first is the Blood dataset, consisting of 17,092 microscopic peripheral blood cell images [104]. The images of this dataset are categorized into eight classes: neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes, erythroblasts, and platelets or thrombocytes. The second one is the Path dataset, containing 100,000 images of human colorectal cancer and healthy tissues [147]. The tissue images are organized into nine classes: adipose (ADI), background (BACK), debris (DEB), lymphocytes (LYM), cancer-associated stroma (STR), colorectal adenocarcinoma epithelium (TUM), mucus (MUC), smooth muscle (MUS), normal colon mucosa (NORM). The third is the Breast dataset containing 780 medical images of breast cancer using ultrasound scans [148]. The Breast dataset is organized into three classes: normal, benign, and malignant.

C. Reference Models and Implementation Details

In order to evaluate the improvements of the proposed multimodal model with respect to previous multimodal and unimodal models, the following reference models are employed for our experiments.

- The multimodal model of [110], which is the preliminary version of our work. This model uses PubMedCLIP's image and text encoders without prompt engineering. Note that, in this model, we use only the Transformer-based encoder (PubMedCLIP-ViT32) because, as shown in [110,117], it is always better than the Resnet-based encoder. In the following, this model is denoted as PubMedCLIP-Multi.

- The unimodal model of [117] that only uses the image modality of PubMedCLIP. In the following, this model is denoted with two options PubMedCLIP-ViT32 and PubMedCLIP-RN50. Here, the image encoders of this unimodal model are exactly the same as those of the multimodal models.

- Three unimodal models using a popular pretrained model, namely DenseNet, MobileNet, or EfficientNet. As mentioned above, recent studies (e.g. [126], [127], [131]) just focus on a certain image type (e.g. Blood or Path), so their findings on the best pretrained model vary. In our evaluation, these models will be compared on the three datasets, using the same setting as the above unimodal and multimodal models.

To clearly see the performance differences of the models, our experiments use the same setup for all models. Especially, because we want to see the performances with a small amount of training data, no techniques of data augmentation and preprocessing are applied. The workflow of the unimodal models is shown in Figure 3.6, where the feature vector provided by a pre-trained model is input into a fully-connected layer for classification. For training of both multimodal and unimodal models, the learning rate is set to 1×10^{-3}, and the batch size is 16. All implementations are based on the PyTorch framework [149]. To obtain stable results, we repeat all experiments ten times and report the average scores over all experiment runs.

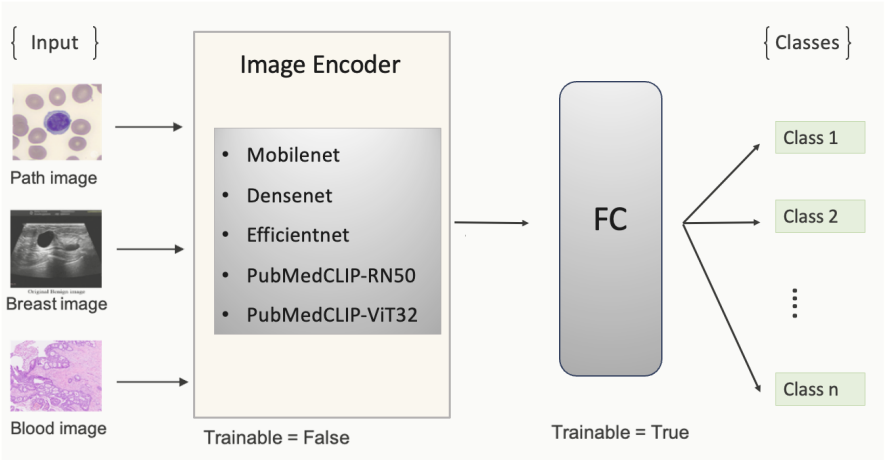


Figure 3.6: Unimodal model of transfer learning for medical image classification.

3.2.4 Experiments

In this section, we shows the performance comparison between the proposed model and the reference models on different datasets. We also perform extensive experiments with different fusion techniques, prompt templates, and different numbers of training samples.

A. Experimental Settings

A key focus of our research was to examine how our model performs under conditions of limited training data. To achieve this, we gradually increase the number of training samples of each class. Specifically, we start with small numbers of training images per class, namely 10, 50, 100, and so on until eventually reaching 80% of the dataset. The images not used for training in each case are set aside for testing. We maintained the same setting for all evaluated models. The incremental increase in training data size enables us to explore the models' learning behaviors as they have access to more training samples. This provides valuable insights into the trade-off between training data volume and performance. Our experiments evaluate the model's performance using accuracy as the primary metric to assess its ability to distinguish between various classes. The accuracy metric, represented by Equation 3.1, provides a comprehensive measure of the overall correctness of the model's predictions. The formula for accuracy metric is represented as follows:

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + F_N + T_N + F_P} \tag{3.1}$$

where T_P, T_N, F_P, F_N denote respectively the true positive, true negative, false positive, and false negative.

B. Experiment-1: Fusion Technique Comparison

In the proposed model, to fuse the text and image vectors for prediction, we employed the MFB fusion technique. To show the benefit of this fusion technique, we com-

pared this technique to two other popular fusion techniques, namely Multimodal Compact Bilinear Pooling(MCB) [150] and Multimodal Tucker Fusion (MUTAN) [151]. For simplicity, template Prompt-1 is used in this evaluation. In Figure 3.7, the performances of the proposed model using one of two vision backbones, PubMedCLIP-RN50 and PubMedCLIP-ViT32, together with the three fusion techniques are shown for the three datasets. For the Blood dataset, the results are shown in Figure 3.7(a), where both PubMedCLIP-RN50 and PubMedCLIP-ViT32 with MFB exhibit increasing accuracy as the number of shots is increased. When the number of shots exceeds 100, the curves reach high accuracy, around 90% for PubMedCLIP-ViT32 and around 85% for PubMedCLIP-RN50. However, when employing the MCB and MUTAN fusion techniques, the curves remain relatively flat, showing minimal improvement even when the number of shots is high. Moreover, the accuracies achieved by MCB and Mutan fusion techniques are significantly lower, approximately 70% for PubMedCLIP-ViT32 with Mutan, 58% for PubMedCLIP-RN50 with Mutan, 66% for PubMedCLIP-ViT32 with MCB, and 43% for PubMedCLIP-RN50 with MCB.

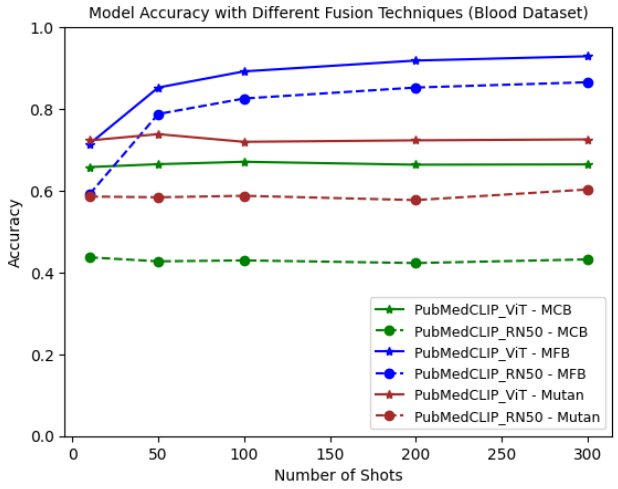
With the Path dataset in Figure 3.7(b), PubMedCLIP-ViT32 with MFB provides the highest curve among all the combinations. When the number of shots exceeds 200, the accuracy surpasses 90%. Besides, the MCB fusion technique provides highly unstable results. With the Breast dataset (Figure 3.7(c)), the behavior is similar to that in the Blood dataset. The MFB fusion technique demonstrates favorable results for both PubMedCLIP-RN50 and PubMedCLIP-ViT32, with increasing accuracy as the number of shots increased. However, the other fusion techniques show much lower results; the accuracy of Mutan with PubMedCLIP-ViT32 (PubMedCLIP-RN50) is consistently around 78% (70%). The MCB fusion technique results in about only 65% for both backbones. Among the three fusion techniques, MUTAN is only better than MFB at very small number of shots (e.g. 10 shots in Path and Breast datasets).

In summary, based on the experiment results, the MFB fusion technique in general shows the best performance across the Blood, Path, and Breast datasets, for both PubMedCLIP-RN50 and PubMedCLIP-ViT32 backbones. In the following evaluations, we will exclusively present the results obtained using the MFB fusion technique.

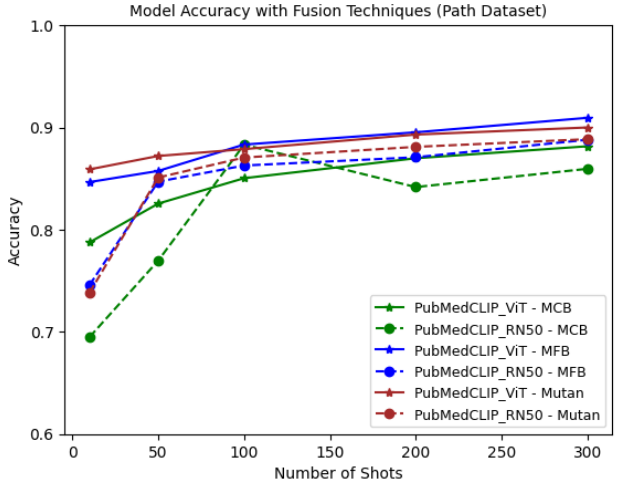
C. Experiment-2: Prompt Template Evaluation

In this part, our evaluation involves testing each prompt template's performance as the number of training samples is increased from 10 samples per class up to 80% of the class. For simplicity, only PubMedCLIP-ViT32 is used the image encoder. The results presented in Table 3.4 highlight the different performances of the prompt templates (i.e. Prompt-0, Prompt-1, Prompt-2, Prompt-3). Futhermore, the results consistently demonstrate that Prompt-3 outperformed Prompt-0, Prompt-1 and Prompt-2 in all datasets. Especially, on the Path dataset, the performance of Prompt-3 quickly jumps to a high level after 500 shots. Meanwhile, on the Breast dataset, the performance of Prompt-3 saturates after 100 shots.

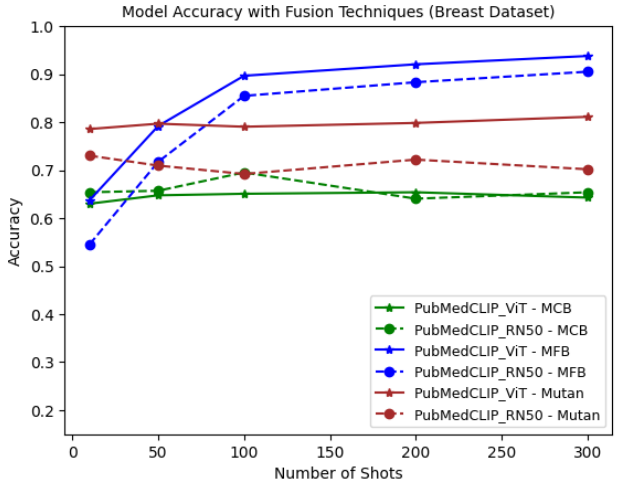
Additionally, the visualization in Figure 3.8 confirms the Prompt-3's consistent and superior performance. The results show that the performance of Prompt-0 is the lowest. More specifically, in Fig. 4, we can see that adding the words "image" and "disease" in Prompt-1 can help improve the performance on Blood and Breast datasets when the number of shots is high, and on Path dataset when the number of shots is medium (from 1500 shots to 5000 shots). Also, in general, Prompt-2 has better performance than



(a) Blood dataset

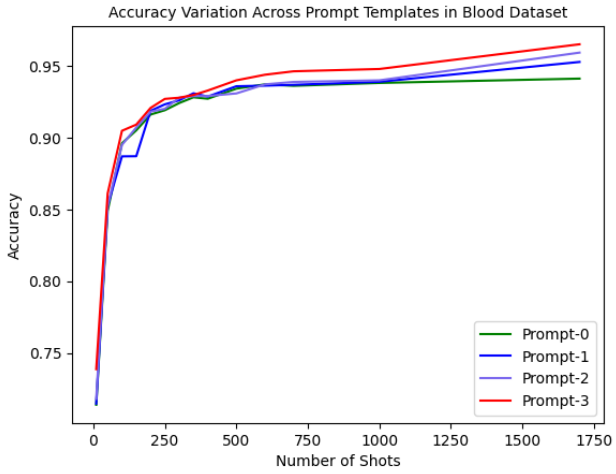


(b) Path dataset

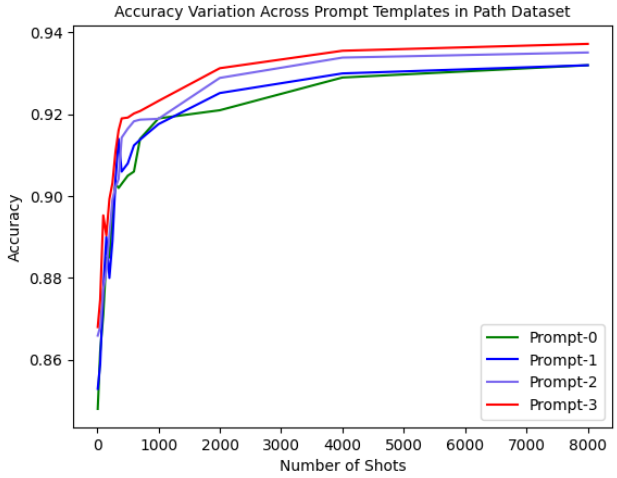


(c) Breast dataset

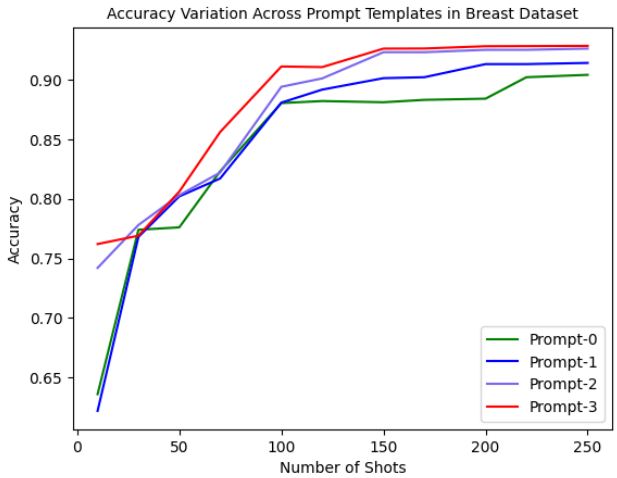
Figure 3.7: Fusion technique comparison



(a) Blood dataset



(b) Path dataset



(c) Breast dataset

Figure 3.8: Prompt techniques comparison

Prompt-1 at most numbers of shots. This observation emphasizes the pivotal role of prompt engineering in the model's performance. The success of Prompt-3 can be attributed to its provision of richer contextual information, which better guides the model in associating image content with the corresponding medical condition. In our future work, we will further investigate the potential of leveraging more intricate and informative language constructs to enhance the performance of multimodal models in medical image classification. In the upcoming evaluation experiments, we will exclusively present results using Prompt-3 in the proposed model.

D. Experiment-3: Model Performance Accuracy

In this section, we compare the performances of the proposed model and reference models on the three datasets. The experimental results are given in Table 3.5. The performances of the models vary across the datasets. Here, we specifically explore the performances when the number of training samples (shots) gradually increases.

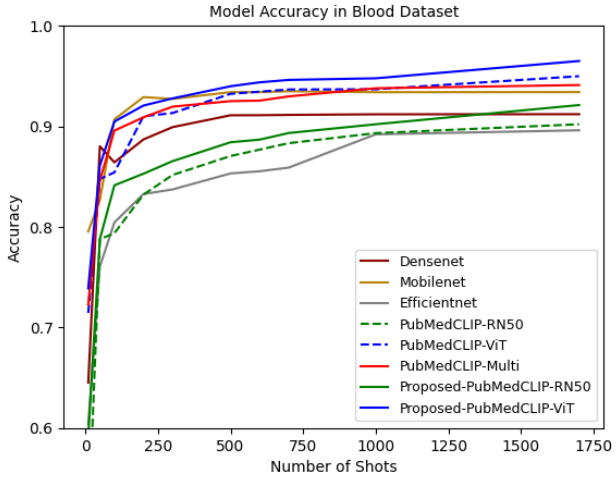
For the 10-shot learning scenario, we trained the models using ten images per class from each dataset and utilized the remaining images for testing. The results indicate that the proposed model (PubMedCLIP-ViT32) achieves the highest or second-highest accuracy across the three datasets. In the Path dataset, our model achieves the highest accuracy score among the models. However, all models perform poorly in the Breast dataset under the ten-shot learning setting.

Notably, PubMedCLIP-ViT32 exhibits superior performance compared to PubMedCLIP-RN50. So, in the following, the proposed model that employs PubMedCLIP-ViT32 is mostly referred to in the discussion.

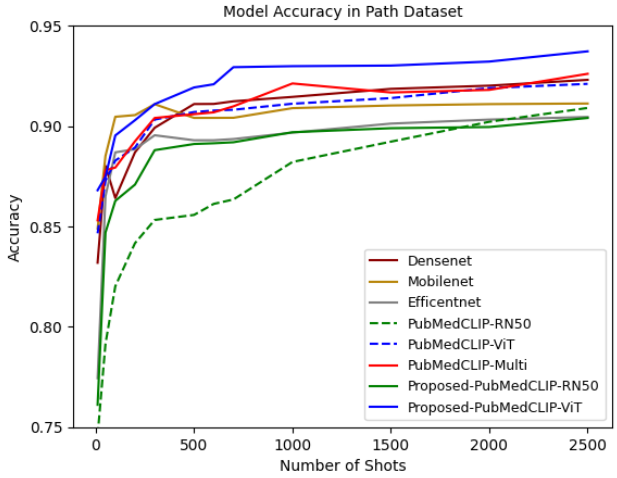
For the 50-shot learning scenario, we increased the training data to 50 images per class. The results show that as the number of training images increases, the overall accuracy of the models improves. Our multimodal model achieves relatively high scores across all three datasets, with accuracy exceeding 80%. Notably, DenseNet and MobileNet perform well on the Blood and Path datasets but poorly on the Breast dataset.

Moving on to the 100-shot learning scenario, we fed 100 images per class into the models for training. The results indicate that our model's accuracy increases slower than MobileNet and DenseNet when transitioning from 50 to 100 training images per class in the Blood and Path datasets. Specifically, MobileNet achieves an accuracy of approximately 90% in the Blood and Path datasets, while DenseNet achieves a similar accuracy in the Path dataset. Nevertheless, our model performs well across all three datasets, with the accuracy surpassing 88%. Notably, in the Breast dataset, our model achieves an accuracy of over 92%, whereas other models fall below 80%.

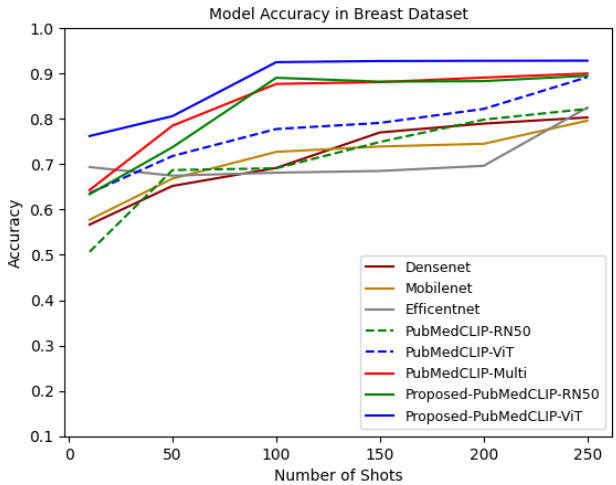
Further increasing the training data to 200 images per class, our model demonstrates outstanding performance across all three datasets. It achieves an accuracy of 92.1% in the Blood dataset, 90.3% in the Path dataset, and 92.8% in the Breast dataset, comparable to those of MobileNet. Compared to DenseNet, our model performs better by approximately 3% in the Blood dataset, 14% in the Breast dataset, and slightly lower by 0.2% in the Path dataset. When we increase the training data to 300 images per class, our model excels across all the datasets. The dependence of model performances on the number of training samples and datasets can be seen more clearly in Figure 3.9. With the Blood dataset (Figure 3.9(a)), our model initially obtained the second-highest accuracy at 200 shots, trailing behind MobileNet. However, from 300 shots onward, the proposed model outperformed all other models. With the Path dataset, initially the



(a) Blood dataset



(b) Path dataset



(c) Breast dataset

Figure 3.9: Performance of the models on each dataset

proposed model again performs worse than MobileNet. However, at 500 shots, the result of MobileNet is lower than the proposed model. Especially with the Breast dataset (Figure 3.9(c)), the proposed model consistently achieved the highest accuracy across all numbers of shots. Meanwhile, all other models, including MobileNet, have much lower performances on this dataset. It can be concluded that the proposed model can consistently achieve good results across different datasets.

Regarding the multimodal model PubMedCLIP-Multi, its performances on Path and Blood datasets are comparable to the unimodal PubMedCLIP-ViT32 (Figure 3.9(a) and (b)); however, on Breast dataset, it is much better than PubMedLCIP-ViT32 and other unimodal models (Figure 3.9(c)). Among the unimodal models, PubMedCLIP-ViT32 is, in general, the best one over all three datasets, except at some small numbers of shots. Meanwhile, the performances of DenseNet, MobileNet, and EfficientNet vary across the datasets. Moreover, the proposed model's results are consistently highest over a wide range of the number of shots. This shows the promising capabilities of both multimodal and unimodal solutions based on PubMedCLIP, thanks to its very large scale.

3.2.5 Ablation study

In this part, we investigate the contributions of the two new components of the proposed model, including the new fusion and the new best prompt (i.e. Prompt-3). So, the comparison includes the following cases:

- Case-1: No new components (i.e. our preliminary model in [110])
- Case-2: Using the new fusion only.
- Case-3: Using the new prompt only.
- Case-4: Using the new fusion and the new prompt (i.e. the proposed model)

Table 3.6: Ablation study's settings and results

Case	**New Fusion**	**New Prompt**	**Dataset**		
			Blood	Path	Breast
Case-1	-	-	0.941	0.919	0.898
Case-2	✓	-	0.953	0.934	0.915
Case-3	-	✓	0.945	0.923	0.911
Case-4	✓	✓	**0.965**	**0.937**	**0.928**

Here, for simplicity, we also employ only PubMedCLIP-ViT32, which is the best encoder for image modality. The accuracy results of the above four cases when training data is 80% of a dataset are shown in Table.3.6. It can be seen that the gains by the new fusion can only be up to 1.7%. Meanwhile, the gains by the new prompt are up to 1.3% and lower than the gains by the new fusion. When both new fusion and new prompt are used, the gains are 2.4%, 1.8%, and 3% on the Blood, Path, Breast datasets, respectively.

These results mean that each new component can improve the performance, and when they are combined, the joint improvement is higher than individual improvements. So, the two new components are complementary to each other, and both are beneficial for the high performance of the proposed model.

3.2.6 Discussions

The above results demonstrate the capabilities of the proposed model, which outperforms reference models in two aspects:

- The superior performances are consistent across three different image types. Whereas previous studies just focus on a certain type (e.g., either Blood, Path, or Breast).
- The behavior is also consistent over a wide range of the number of shots. It should be noted that existing studies mostly try to enlarge the amount of training data (e.g. by various data augmentation techniques) to improve the performance.

The advantages of the proposed model can be attributed to the robustness (or generalizability) of the large-scale and multimodal nature of the pre-trained PubMedCLIP model, together with prompt engineering and feature fusion.

It should be noted that the image encoder in the proposed model is the same (i.e., unmodified) as those used in unimodal models (using either PubMedCLIP-RN50 or PubMedCLIP-ViT32). However, thanks to the processing of both image input and text input, the proposed multimodal model always outperforms the corresponding unimodal model. This is an interesting benefit of large multimodal models like PubMedCLIP.

In addition, the experiments show that PubMedCLIP-ViT32 always performs better than PubMedCLIP-RN50 in both unimodal and multimodal cases. On the Blood dataset, the unimodal model using PubMedCLIP-ViT32 is only worse than the multimodal model using PubMedCLIP-ViT32, which is even better than all other unimodal and multimodal models. This means the vision transformer technology is more effective than CNN in this classification task.

Our results also emphasize the importance of text prompt engineering to enhance a model's performance. In our study, adding more medical context into the prompt template helps the model understand more about the image that the model needs to classify. The improved performance when incorporating such keywords into the prompt can be attributed to the unique capabilities of the PubMedCLIP model, which is a fine-tuned version of CLIP tailored for medical applications. PubMedCLIP has been trained with a huge amount of images and associated text. A text prompt can be considered as a context input into the multimodal model. It seems that when appropriate words are provided in the prompt, the context will be clearer to the model, and thus, the performance at the output will be higher. So, it is important to empower the model with a richer context rather than a simple label or short description.

Furthermore, our model's robustness in image classification accuracy is fortified by fusing feature vectors of image and text inputs. This fusion of image and text vectors, coupled with an extensive text vector dictionary, equips our model to tackle a broad spectrum of medical conditions, ensuring consistent high accuracy across diverse image classification tasks. This multifaceted solution has been shown to be beneficial in medical image classification, with limited training data and adaptability across various datasets.

3.2.7 Conclusions and Future Work

In this work, we have investigated the capability of transfer learning based on PubMedCLIP for medical image classification. We proposed a multimodal model that harnesses text prompts and images to achieve high accuracy even with limited training data, surpassing the performance of traditional transfer learning models. The advantages of the proposed model could be attributed to the multimodal pre-trained backbones, prompt engineering, and feature fusion. Especially, the effective use of prompt templates in our model highlights its potential for various image classification domains. For future work, we will extend this approach by enhancing prompts through developing automated or context-aware prompts, which may improve the model's performance across diverse domains. Additionally, we will further evaluate the adaptability of the proposed model to various medical subfields and exploring cross-domain applications.

3.3 Conclusion and Future Directions for Medical Image Classification

This chapter presented two innovative studies to improve medical image classification through advanced deep-learning approaches. The first study utilized the PubMedCLIP model for transfer learning, demonstrating robust performance across various datasets and highlighting the potential for significant cost savings in data labeling despite certain feature extraction limitations requiring further refinement. The second study introduced a multimodal model that integrates text prompts and images, achieving superior accuracy with limited training data by leveraging multimodal pre-trained backbones, prompt engineering, and feature fusion. Both studies underscore the effectiveness of advanced transfer learning and multimodal techniques in enhancing medical image classification, and they lay the groundwork for future research focused on developing automated or context-aware prompts, improving model adaptability to different medical subfields, and exploring cross-domain applications. In future works, our research will continue to refine these models to enhance their robustness, scalability, and applicability, thereby advancing the capabilities of medical image analysis within smart societies.

Table 3.4: Accuracy values for different prompts.

Dataset	No. of shots	Prompt-0	Prompt-1	Prompt-2	Prompt-3
Blood	10	0.714	0.715	0.718	0.739
	50	0.849	0.852	0.852	0.861
	100	0.896	0.887	0.895	0.905
	150	0.905	0.887	0.907	0.909
	200	0.916	0.919	0.918	0.921
	250	0.919	0.923	0.920	0.927
	300	0.924	0.926	0.928	0.928
	350	0.928	0.931	0.929	0.930
	400	0.927	0.928	0.929	**0.933**
	500	0.934	0.936	0.931	**0.939**
	600	0.937	0.936	0.937	**0.944**
	700	0.936	0.937	0.939	**0.946**
	1000	0.938	0.939	0.940	**0.948**
	80% of data	0.941	0.953	0.959	**0.965**
Path	10	0.848	0.853	0.866	0.868
	50	0.862	0.859	0.868	0.875
	100	0.871	0.877	0.877	0.895
	150	0.885	0.880	0.882	0.890
	200	0.885	0.890	0.890	0.903
	250	0.896	0.889	0.899	0.903
	300	0.903	0.897	0.904	**0.911**
	400	0.902	0.907	0.914	**0.919**
	500	0.903	0.908	0.906	**0.919**
	600	0.905	0.912	0.908	**0.921**
	700	0.906	0.913	0.912	**0.929**
	1000	0.914	0.917	0.913	**0.930**
	80% of data	0.932	0.932	0.935	**0.937**
Breast	10	0.636	0.622	0.742	0.762
	50	0.776	0.802	0.803	0.806
	100	0.883	0.891	0.894	**0.911**
	150	0.881	0.902	0.923	**0.926**
	200	0.884	0.913	**0.925**	**0.928**
	80% of data	0.904	0.913	**0.925**	**0.928**

Table 3.5: Model performance

Few shot	Pre-trained model	Blood dataset	Path dataset	Breast dataset
10-Shots	DenseNet	0.646	0.832	0.567
	MobileNet	0.795	0.849	0.577
	EfficientNet	0.603	0.774	0.694
	PubMedCLIP-RN50	0.497	0.746	0.507
	PubMedCLIP-ViT32	0.714	0.846	0.636
	PubMedCLIP-Multi	0.723	0.847	0.643
	Proposed-PubMedCLIP-RN50	0.691	0.761	0.634
	Proposed-PubMedCLIP-ViT32	0.739	0.858	0.762
50-Shots	DenseNet	0.880	0.889	0.652
	MobileNet	0.826	0.885	0.668
	EfficientNet	0.761	0.865	0.675
	PubMedCLIP-RN50	0.722	0.791	0.687
	PubMedCLIP-ViT32	0.847	0.873	0.778
	PubMedCLIP-Multi	0.852	0.872	0.785
	Proposed-PubMedCLIP-RN50	0.787	0.847	0.737
	Proposed-PubMedCLIP-ViT32	0.861	0.868	0.806
100-Shots	DenseNet	0.864	**0.902**	0.692
	MobileNet	**0.907**	**0.904**	0.727
	EfficientNet	0.804	0.869	0.681
	PubMedCLIP-RN50	0.794	0.821	0.691
	PubMedCLIP-ViT32	0.854	0.883	0.777
	PubMedCLIP-Multi	0.887	0.878	0.877
	Proposed-PubMedCLIP-RN50	0.841	0.862	0.890
	Proposed-PubMedCLIP-ViT32	**0.905**	0.895	**0.927**
200-Shots	DenseNet	0.887	**0.905**	0.789
	MobileNet	**0.929**	**0.905**	0.745
	EfficientNet	0.833	0.888	0.696
	PubMedCLIP-RN50	0.832	0.842	0.749
	PubMedCLIP-ViT32	0.910	0.889	0.822
	PubMedCLIP-Multi	0.911	0.892	0.891
	Proposed-PubMedCLIP-RN50	0.853	0.871	0.883
	Proposed-PubMedCLIP-ViT32	**0.921**	**0.903**	**0.928**
300-Shots	DenseNet	0.899	0.907	-
	MobileNet	**0.927**	**0.910**	-
	EfficientNet	0.837	0.895	-
	PubMedCLIP-RN50	0.851	0.853	-
	PubMedCLIP-ViT32	0.913	0.903	-
	PubMedCLIP-Multi	0.919	0.902	-
	Proposed-PubMedCLIP-RN50	0.865	0.888	-
	Proposed-PubMedCLIP-ViT32	**0.927**	**0.911**	-
500-Shots	DenseNet	0.911	0.908	-
	MobileNet	**0.933**	0.904	-
	EfficientNet	0.853	0.892	-
	PubMedCLIP-RN50	0.870	0.855	-
	PubMedCLIP-ViT32	**0.932**	0.907	-
	PubMedCLIP-Multi	0.924	0.911	-
	Proposed-PubMedCLIP-RN50	0.884	0.891	-
	Proposed-PubMedCLIP-ViT32	**0.939**	**0.919**	-
80% dataset	DenseNet	0.926	0.918	0.803
	MobileNet	**0.939**	0.915	0.796
	EfficientNet	0.873	0.912	0.825
	PubMedCLIP-RN50	0.902	0.895	0.822
	PubMedCLIP-ViT32	**0.949**	0.917	0.892
	PubMedCLIP-Multi	0.938	0.919	0.90
	Proposed-PubMedCLIP-RN50	0.921	0.918	0.892
	Proposed-PubMedCLIP-ViT32	**0.965**	**0.937**	**0.928**

Chapter 4

Patient Similarity Using Semi-structured Data

This chapter introduces a novel approach utilizing self-supervised learning to analyze electronic health records (EHRs) and determine patient similarity. Our method incorporates structured and unstructured data from medical text notes, diagnoses, lab results, demographics, and other sources within EHRs. The primary focus of this chapter is to present a deep learning framework tailored for handling the complexities of both structured and unstructured data inherent in EHRs. By leveraging advanced deep learning techniques, we aim to extract meaningful insights from diverse data types, enabling accurate assessment of patient similarity.

4.1 Introduction

4.1.1 EHR Data

EHRs is a digital repository of a patient's health information, providing real-time access to authorized users. EHR encompasses a wide range of medical data, including laboratory test results, diagnoses, medications, radiology images, medical history, and other clinical information [152]. While EHRs data are primarily intended to enhance healthcare efficiency in operation management, many additional applications in clinical informatics have been investigated. Specifically, patient data stored within EHR systems has been utilized for various purposes, including healthcare concept identification [153], disease prediction [154], clinical decision support systems [155], and beyond.

Figure 4.1 show an illustration of EHR data collected from various sources in structured and unstructured formats [156]. Structured data within EHRs typically includes diagnostic codes (such as ICD-9 codes), vital signs, laboratory test results, demographic information, and medication records. On the other hand, unstructured data encompasses radiology reports, clinical notes, discharge summaries, and patient narratives. Unstructured data has a lot of useful information that offers a comprehensive history of a patient. In our research context, we leverage EHR data to develop and evaluate predictive models capable of forecasting patient outcomes using a combination of structured and unstructured data sources. We harness structured data to guide model training alongside textual data extracted from EHRs.

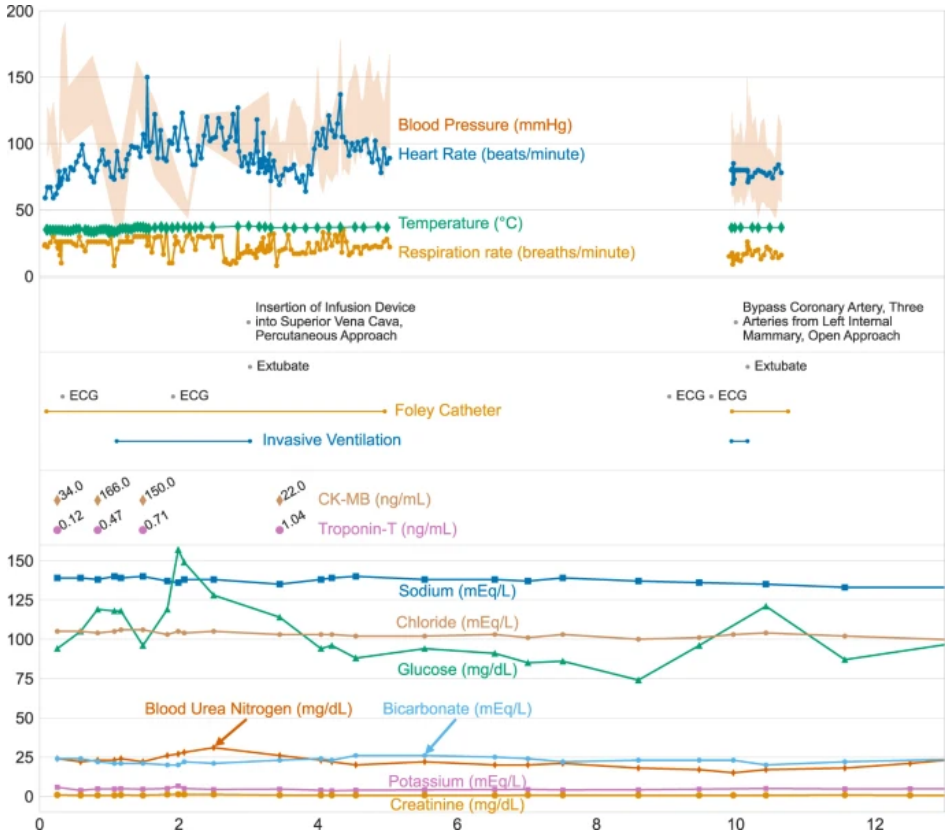


Figure 4.1: An illustration of EHR data of a patient from MIMIC-III Database

4.1.2 Motivations

In the era of big data mining, an abundance of Electronic Health Records (EHR) has become readily available. The central challenge lies in harnessing this data for improved patient care without increasing the burden on healthcare professionals. Patient similarity matching is one of the key tasks to unlocking the potential of EHR data [157, 158]. Patient similarity seeks to quantify the resemblance between two patients by analyzing their EHR records. The implications of achieving meaningful patient similarity are profound. It can revolutionize various applications, from identifying similar patients for a given case to comparing treatments within similar patient groups, ultimately leading to enhanced patient outcomes and more efficient healthcare decision-making [157].

However, a significant hurdle in establishing patient similarity is a good representation of patient features extracted from healthcare professionals' text notes, encompassing contributions from doctors and nursing staff. The process involves using language models to generate feature vectors for each patient. Current methods of patient similarity can be classified into two groups, supervised and unsupervised [159]. In supervised methods, the similarity labels are used to train a similarity prediction model [160, 161]. However, the actual labeled data is very costly and usually not public. In unsupervised methods, a special vector is extracted from each EHR and then used for similarity matching between patients [158, 162]. As surveyed in [159], most of existing studies are unsupervised learning.

In this study, we propose a self-supervised method for better feature extraction. Specifically, we try to employ various tags or guides (e.g. outcomes, diagnosis codes,

categories) to help a deep learning model adjust to the specific characteristics of the given dataset. The experimental results show that when the number of tags is sufficiently provided, the performance of similarity matching is significantly increased. We also carry out an analysis of feature extraction at different layers of a deep learning model. Our research comprehensively evaluates word embedding methods, incorporating both self-supervised and pretrained learning models.

4.2 Related Work

In the field of healthcare informatics, many studies are dedicated to understanding patient similarity. These studies aim to mimic the clinical reasoning of doctors, automatically identify similar patients for a given index patient, and forecast diagnoses based on comparisons with similar or dissimilar patients. Utilizing an appropriate similarity measure facilitates various downstream applications, including personalized medicine [20], medical diagnoses [163],tracking patient trajectories [164], and disease prediction [165].

Numerous similarity learning methods have been proposed for analyzing healthcare datasets [166–169]. These methods typically rely on handcrafted vector representations, such as demographic or average numerical values, to derive similarity measures from EHR data. While successful in mapping medical events to vector spaces, these methods often face limitations due to the lack of comprehensive patient explanations provided by doctors or medical specialists, such as diagnosis summaries and clinical notes.

Extracting meaning from free-text medical notes poses a significant challenge in EHR research. Textual data captures a wide range of information, including symptoms, patient-reported outcomes, progress notes, differential diagnoses, illness trajectory, and behavioral history. Clinical textual records play a crucial role in tracking patient progress and planning their care accordingly. Establishing patient similarity based on their clinical text notes is essential for identifying patients with similar diagnoses.

Because labels of patient similarity need experienced professionals, they are rare and usually are not made public [158]. In [160,161], some supervised learning methods are proposed to find similar patients of different diseases, where convolutional neural networks are employed to obtain a feature vector for each patient. Here, patients of the same labeled class (disease) are considered similar (positive) while patients of different classes are considered disimilar (negative). So the task in these studies is in fact a kind of classification task. It should be noted that, in practice, patients of the same disease may have very different EHRs and outcomes (e.g. dead or discharged) [170].

In unsupervised methods, feature vectors are extracted from patients' EHRs in various manners, usually with the help of some pretrained deep learning models. Then the similarity between two patients is computed as the cosine similarity score of the two corresponding feature vectors. In [158], different pretrained BERT variants are employed to extract features. It is found that the SciBERT model is the only one that is better than the original BERT model. In [162], a wide variety of medical events (e.g. prescriptions, microbiology, laboratory, output, etc.) are extracted to build large feature sets. Also, complex representation structures can be heuristically generated from EHRs. For example, in [171] [159], an EHR is converted to a tree representation with branches and nodes. However, the final feature representation strongly depends on the traversing path across the branches of a tree. Our proposed method can be considered as an

self-supervised method. Specifically, we try to exploit various tags or guides which are available in a datasets to help deep learning models adapt to the characteristics of the given datasets. Though such training may not provide high accuracy (which actually is not our goal), the learned features are expected to be better than those of an original pre-trained model.

4.3 Overview of Methodology

Measuring patient similarity involves the transformation of diverse Electronic Health Records (EHRs) data into standardized formats for the purpose of calculating the proximity between pairs of patients. Clinical narratives offer a succinct depiction of a patient's condition upon admission to the Intensive Care Unit (ICU), providing valuable insights for caregivers. However, these narratives often contain excessive information, including redundant structured data and contributions from various sources, which can pose challenges when attempting to model them for prognosis or embedding.

Machine learning and Natural Language Processing (NLP) have demonstrated remarkable capabilities in learning from data, regardless of its complexity. These technologies have the potential to yield meaningful outputs from even the messiest datasets.

4.3.1 Methodological Flowchart

This research involved a series of stages, comprised of three primary phases as depicted in Figure 4.2. The study workflow can be summarized in the following steps:

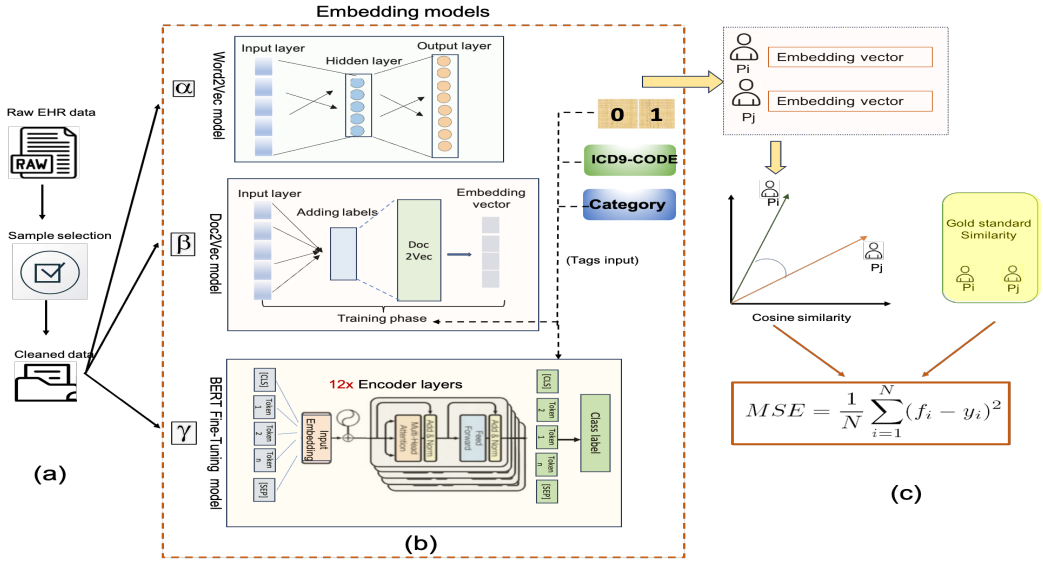


Figure 4.2: Study framework, (a) Data preprocessing; (b) Embedding model: (α) Word2Vec embeddings, (β) Doc2Vec embedding, (γ) BERT-based with twelve encoder layers; (c) Patient similarity calculation and MSE calculation

1. This step encompasses two tasks. Firstly, it involves data sampling and selection from a substantial MIMIC-III notes dataset. Secondly, data cleaning is performed through the Natural language toolkit (NLTK).

2. Utilization of Various Embedding Models: In this phase, a variety of embedding models are employed to train different tags and generate corresponding feature vectors.

3. Similarity Calculation and Mean Squared Error (MSE) Computation: This step involves the calculation of similarity metrics and the subsequent computation of the Mean Squared Error (MSE) in comparison to the Gold standard.

4.3.2 Dataset

For this study, we chose real medical text as the litmus test for evaluating our approach and models to real-life healthcare data. The source of our textual narratives was the Medical Information Mart for Intensive Care-III (MIMIC-III) [172], a publicly accessible multiparameter monitoring system deployed in intensive care units over 11 years. This extensive repository features a wealth of structured medical data, encompassing physiological information and unstructured textual notes contributed by various healthcare professionals.

To focus our study, we needed to find a specific medical condition that offered an ample dataset and a noteworthy mortality rate, which was crucial for training our model with diverse tags. Consequently, we selected "Pneumonia" as our primary disease of interest, which yielded 1405 cases with a mortality rate of 23%. These patients generated 59,727 notes, collectively authored by physicians, nurses, radiologists, and nutritionists.

Each sequence of notes was associated with a binary label indicating patient outcomes, wherein notes from discharged patients were assigned class "0," while those from patients who experienced in-hospital deterioration were designated as class "1." Additionally, each sequence of notes was associated with an ICD9-CODE label extracted from DIAGNOSES-ICD, and the SEQ-NUM was recorded to facilitate the calculation of Gold standard similarity.

In the dataset for pneumonia, we identified a total of 53 unique ICD9-CODEs, which were further categorized into 11 distinct categories as listed in the ICD9-CODE list [173]. All of these exploratory data analyses were conducted using Python libraries.

In this phase, we perform text data cleaning by adhering to fundamental Natural Language Processing (NLP) guidelines, leveraging the capabilities of the Natural Language Toolkit (NLTK). To ensure uniformity, we standardize all text to lowercase and employ regular expressions to eliminate punctuation, excessive white space, line breaks, and non-alphanumeric characters. Furthermore, we incorporate a stop-word dictionary to eliminate irrelevant elements from the text.

4.4 Feature Extraction with Self-supervised Learning

In this study, we consider a range of methods, spanning from traditional techniques to cutting-edge NLP model. For our self-supervised approach, we employed Doc2Vec models and state-of-the-art BERT model.

4.4.1 Doc2Vec Model

Doc2Vec [174] model, as opposed to the Word2Vec model, is used to create a vectorized representation of a group of words taken collectively as a single unit. While the

Doc2Vec model original is unsupervised, it can be used in a supervised context when combined with labeled data. For example, we can use Doc2Vec embeddings as features in a supervised machine learning model (e.g., a classification model) to perform specific tasks like document classification, sentiment analysis, or topic modeling. In this case, the Doc2Vec model is used for feature extraction, but the overall approach is self-supervised because it involves labels (more exactly tags) in a dataset. In our study, we employed a variety of tags, including Binary label tags (0-1), Category tags (comprising 11 distinct categories), and ICD9-Code tags (encompassing 53 distinct codes). Then, the Doc2Vec model was used to create embeddings that capture the inherent semantics of the text data while simultaneously considering these diverse tags. This integration allowed us to generate embeddings that encapsulate the multifaceted relationships between the text content and associated tags.

4.4.2 Bert-based Model

BERT [175], which stands for Bidirectional Encoder Representations from Transformers (BERT), is a state-of-the-art natural language processing model known for its contextual understanding of language. BERT embedding creates different vectors for a word used in different contexts. It utilizes a transformer encoder to represent a word in a higher-dimensional space, capturing relations between distant words more efficiently than traditional bidirectional encoders. A text representation by BERT depends on tokenization, which involves breaking the text into individual words or subword units and adding special tokens for BERT's input format.

In our study, we explore feature extraction from text notes using BERT-based models, akin to our earlier approach with the Doc2Vec model, that is self-superved learning with a diverse range of labels or tags and subsequently extract feature vectors from the text data.

1. **Binary classification**: In the initial approach, we harnessed the capabilities of a BERT-based model to address the binary classification task. To train the model, we use the two distinct labels, 0 and 1 (dead or alive).

2. **Multi-label classification with ICD9-codes**: this multi-label classification is trained with a diverse set of ICD9-Codes containing up to 53 unique codes. These codes span a comprehensive spectrum of medical diagnoses and conditions, reflecting the intricacies of healthcare data.

3. **Multi-label classification with Categories**: In this case, a diverse set of 11 distinct categories of disease is employed.

Feature Extraction from Different Layers: Following each training phase, we extract feature vectors from different layers within the fine-tuned BERT model. Specifically, from the second layer, the sixth layer and the final layer. This feature extraction process captures text representations at various depths within the model, each layer offering a unique perspective on the underlying text data. These rich, context-aware representations enhance patient similarity analyses and provide valuable insights into the layers.

Table 4.1: Hyperparameters and Characteristics

Hyper-parameters	Characteristics
Optimizer	AdamW
Batch Size	32
Dropout	0.1
Loss	CrossEntropyLoss
Learning Rate	2e-5
Max Length	512
Epochs	30

4.4.3 Word2Vec Model

Using Word2Vec [176] to extract features can be considered as an unsupervised method, and this approach is used as a baseline in our evaluation. Word2vector is a neural network-based model designed to learn distributed representations of words, typically in large corpora of text, without needing labeled data or explicit supervision. Word2Vec employs two main algorithms for learning word embeddings: Continuous Bag of Words (CBOW) and Skip-gram. For our study, we used (CBOW) approach; the model is trained to predict a target word (the center word) based on the surrounding context words within a fixed-size window. The surrounding context words are used as input to predict the target word, but no explicit labels or annotations are used for training. The model learns by adjusting word embeddings to make it more accurate at predicting the target word from its context [176].

4.5 Experiments

This section details the methodologies employed in constructing the Gold standard, which is used as reference for predicted similarity values [159, 171]. Subsequently, we provide an overview of the evaluation criteria.

4.5.1 Evaluation Metrics

A. Gold Standard Similarity

As in [159, 171], the final diagnosis codes are used to compare two patients, taking into account the diagnosis code's priority within each patient's EHR. To determine the similarity between two patients, A and B, we employ the formulation, defined by Pokharel et al. [171], as follows

$$\text{Sim}(A, B) = \frac{\sum_{i=1}^{N} \min(f_{a_i} \cdot w_i, f_{b_i} \cdot w_i)}{\text{avg}\left(\sum_{i=1}^{N} (f_{a_i} \cdot w_i), \sum_{i=1}^{N} (f_{b_i} \cdot w_i)\right)}$$

In the equation, the terms $(f_{a_i} \cdot w_i)$ and $(f_{b_i} \cdot w_i)$ represent the weighted values of diagnosis i with priority p expressed as an ICD9 disease code for patients A and B, respectively.

B. Metric for Assessing Patient Similarity

To assess the performance of various feature embedding sets, we employ the Mean Squared Error (MSE) metric [177, 178]. MSE quantifies the prediction error, which is essentially the difference between the gold-standard similarity values and the predicted values. These predicted values are determined by calculating the cosine similarity of patient pairs based on the feature vectors obtained from distinct embedding models and different layers.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(actual_i - predict_i)^2$$

In the MSE equation, n represents the total number of patients in the dataset, and $\sum$ denotes the summation notation. This comparison helps us assess the model's ability to predict patient similarities accurately.

4.5.2 Experimental Results

In analyzing our experimental results, we set out to address two fundamental questions:

1. Can self-supervised learning with tags improve the performance of similarity matching compared to directly using pre-trained models?

2. Do features from different layers have different impact on patient similarity?

The results, presented in Table 4.2, illuminate the findings from our extensive experiments. We begin by examining the performance of the pretrained Word2Vec, which produced a relatively high MSE value of 0.7385. This result suggests that the features extracted by this model fail to differentiate between patients adequately.

With the fine-tuned Doc2Vec model by self-supervision, which was trained in three cases (2 tags, 11 tags, and 53 tags), we observe lower MSE values compared to Word2Vec. However, the MSE values for the different tag sets (2 tags, 11 tags, and 53 tags) are 0.2094, 0.2669, and 0.2239, respectively. That means the variation in tag counts does not significantly impact the model's performance.

Interestingly, the pretrained BERT model results in the highest MSE value, even higher than that of Word2Vec. With fine-tuned BERT models, the results reveal distinct trend with respect to the number of tags. For the model trained with two tags, MSE values are 0.5640 for layer 2, 0.5652 for layer 6, and 0.5768 for the last layer. While these values are lower than those obtained from the pretrained Word2Vec and BERT models, they are still higher than those from the fine-tuned Doc2Vec model.

The situation changes when the model is trained with 11 tags. For this case, when extracting feature vectors from layer 2, the MSE is 0.0395, and 0.0953 for layer 6, which are much lower than previous cases. This outcome indicates that the model trained with broader category labels, reflecting diverse patient diseases rather than just binary life status, is more effective in representing text notes. Additionally, feature vectors from the earlier layers outperform the last layer, which is primarily designed for classification

Table 4.2: MSE of similarity matching using different feature embeddings

Model	No. of tags	Layer	MSE
Word2Vec	-	-	0.7385
Doc2Vec	2 tags	-	0.2094
	11 tags	-	0.2669
	53 tags	-	0.2239
Pre-trained BERT	-	-	0.7533
BERT-based (fine-tunning)	2 tags	Layer 2	0.5640
		Layer 6	0.5652
		Last layer	0.5768
	11 tags	Layer 2	**0.0395**
		Layer 6	**0.0953**
		Last layer	0.2793
	53 tags	Layer 2	**0.0457**
		Layer 6	**0.0975**
		Last layer	0.1211

purposes. The first and middle layer feature vectors effectively capture the patient's text notes within category-specific contexts. In case of training with 53 tags (ICD9-code tags), we find results that are comparable to the case of 11 tags. MSE values are 0.04569 for the second layer, 0.09754 for the sixth layer, and 0.121 for the last layer. The outcome suggests that when the model is trained with more granular tags encompassing multiple ICD9 codes, the feature vectors extracted from the first and middle layers are similar in effectiveness to those obtained from models trained with fewer tags. However, the feature vectors from the last layer of the 53-tag model outperform those from the 11-tag model in representing patient features. Figure 4.3 illustrates the variance in MSE values resulting from feature extraction by the BERT-based fine-tuning model across different tag and layer extraction scenarios. The visualization in Figure 4.3 reveals interesting trends in MSE values depending on the number of tags used during model training and the choice of layers for feature extraction. When training the model with only 2 tags (binary classification), the MSE values remain relatively consistent across different layers. Minimal variation is observed in the MSE values when extracting features from layer 2, layer 6, or the last layer. In contrast, a notable trend emerges when the model is trained with 11 tags. The MSE values are lowest when extracting features from layer 2, indicating better performance at capturing features relevant to the classification task. However, the MSE values increase when extracting features from layer 6, and they reach their highest values when extracting from the last layer. This observation suggests that deeper layers capture more complex features but may also introduce more noise into the feature representation. Similar trends are observed when

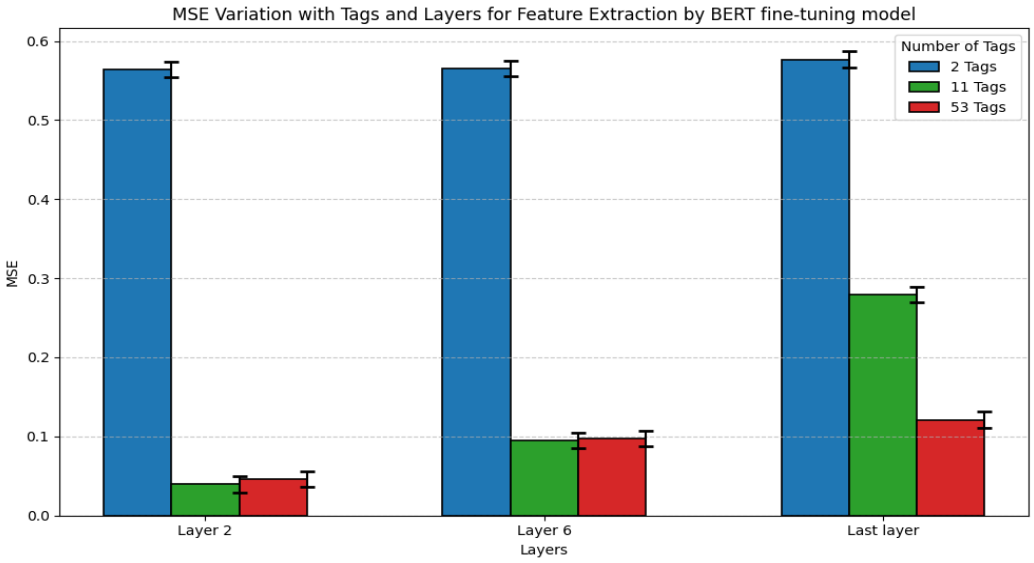


Figure 4.3: MSE Variation with Tags and Layers for Feature Extraction by BERT fine-tuning model

Table 4.3: BERT Fine-Tuning Accuracy in different training tags

Fine-tuned BERT model	Accuracy
2-tag	0.851
11-tag	0.7548
53-tag	0.71

training the model with 53 tags. Once again, feature extraction from layer 2 yields the lowest MSE values, followed by increasing MSE values for extraction from layer 6 and the last layer. Notably, when training with 53 tags, the average MSE values are lower compared to training with 11 tags, indicating improved performance with a larger number of tags.

The observation of BERT (and Word2Vec) underscores the characteristics of feature vectors extracted from pre-trained BERT model. This model is inherently generic and contain rich contextual information, making it suitable for various natural language understanding tasks. However, it is important to note that these feature vectors do not carry task-specific information. They have not been fine-tuned or adapted to a particular task or dataset. As a result, their generality and lack of task-specific information may limit their effectiveness for patient similarity computation.

In table 4.3, we present the accuracy of models trained with varying numbers of tags during two-tag training, 11-tag training, and 53-tag training, all using the same hyper-parameters as shown in Table 4.1. The results show that the 2-tag training achieved the highest validation accuracy, reaching 85.1%. Conversely, the 11-tag training and 53-tag training yielded lower accuracy values, at 75.48% and 71%, respectively. Despite the 2-tag training showing superior accuracy, our feature extraction analysis reveals a different story. The 2-tag fine-tuned model performed less satisfactorily when comparing the feature vectors obtained from three different layers of these trained models for pa-

tient similarity computation. The observed discrepancy in feature vector performance can be attributed to the nature of how the BERT model are fine-tuned. Models trained with a more extensive set of tags allow the model to capture richer information and nuances within the text notes of patients. Consequently, the feature vectors derived from these models serve as superior representations for patient text notes in the context of similarity computation.

From the above discussion, the key findings can be summarized as follows.

1. All cases of self-supervised learning with tags improve significantly the performance of similarity matching compared to using pretrained models (Word2Vec and BERT).

2. When the number of tags is sufficient (i.e. 11 tags or 53 tags), fine-tuned BERT models provide the best performance.

3. The fine-tuned Doc2Vec model has good performance, however its dependence on tag sets is not strong.

4. Early layers provide better feature vectors than the final layer.

5. Pretrained complex BERT model is even worse than the pretrained simple Word2Vec model.

4.6 Conclusion and Future Work

In this study, we have presented a new method for patient similarity using self-supervised learning. It was shown that self-supervised learning with tags improve significantly the performance of similarity matching compared to using pretrained models. Also, when the number of tags is sufficient, fine-tuned BERT models provide the best matching performance. Our findings underscore the importance of carefully selecting the model architecture and the specific layer from which feature vectors are derived. These considerations are crucial in achieving accurate and context-aware patient similarity assessments, particularly when working with clinical text data. Additionally, our results illuminate the substantial influence of tag label granularity on model performance, further underscoring the need for thoughtful model training. Our research endeavors are poised to expand as we look to the future. One avenue of exploration involves extending our analysis to encompass a broader spectrum of feature embedding models, allowing us to gain deeper insights into their efficacy and applicability to patient similarity tasks. Moreover, we are committed to enhancing patient similarity computation by integrating advanced techniques like frequent pattern mining and machine learning methodologies. By pushing the boundaries of feature embedding and fusion techniques, we aim to refine our understanding of patient similarity and contribute to more accurate and insightful clinical decision support systems.

Chapter 5

Conclusion

5.1 Concluding Remarks

Throughout this thesis, we have undertaken a comprehensive exploration of machine learning (ML) and deep learning (DL) approaches for mining both structured and unstructured data within the framework of smart societies. By proposing and analyzing various ML and DL algorithms and models, our aim has been to contribute to developing AI applications that enhance the quality of life for citizens in smart cities.

In Chapter 2, we introduced the SPP-ECLAT algorithm, specifically designed to mine stable periodic-frequent patterns efficiently in large structured datasets. Our algorithm prioritizes the extraction of meaningful patterns while minimizing the search space. Through rigorous experimentation with real-world and synthetic datasets, we have demonstrated the superior performance of the SPP-ECLAT algorithm, particularly in terms of runtime efficiency and memory usage, compared to existing methods. The findings from our research hold significant implications for the advancement of AI applications in smart society contexts. By deriving valuable insights from vast amounts of data quickly, our work contributes to developing innovative solutions to enhance various aspects of urban life. For example, we aim to provide real-time feedback to improve recognition accuracy and processing speed and to address complex real-world challenges through advancements in machine learning techniques.

In Chapter 3, we proposed an unimodal and multimodal transfer learning model for classifying medical images, especially in cases with limited training data. Our research contributes to diagnostic efforts based on medical images, supporting the work of medical professionals by providing high-accuracy classifications. This work performance helps to reduce the burden on medical teams. It facilitates more accurate diagnoses, which is crucial in a smart society where the rapid processing of vast amounts of information is essential. Additionally, our model minimizes the need for expert involvement in initial data labeling, addressing challenges such as medical workforce shortages experienced during events like the COVID-19 pandemic.

Finally, in Chapter 4, we presented a deep self-learning method for semi-structured data, focusing on Electronic Health Records (EHRs). EHRs contain vital information, including structured data such as patient demographics and medical history, and unstructured data from free-text notes. Our model utilizes this comprehensive dataset to identify patients with similar health profiles, providing doctors with valuable second opinions and contributing to quick and effective diagnoses. Furthermore, our research supports the efficient delivery of medical services in rapidly growing urban populations,

contributing to the establishment of a healthier future for our nation.

5.2 Future Work

Moving forward, the findings and insights garnered from each chapter of this thesis pave the way for several promising avenues of future research.

For frequent pattern mining, future research will delve deeper into the lability concept across different types of itemsets, exploring its implications and applications in various domains. Additionally, there is an intriguing opportunity to investigate the discovery of stable periodic-frequent itemsets in uncertain databases, a challenging yet promising area for pattern mining research. Moreover, the focus will extend to identifying Stable Periodic-frequent Patterns (SPPs) in static temporal data, further exploring stable itemsets in graphs, data streams, and symbolic databases, and offering insights into temporal and dynamic patterns across diverse datasets.

Regarding deep learning for medical images, future work entails enhancing unimodal and multimodal transfer learning models for medical image classification. While the PubMedCLIP pre-trained model exhibited promising performance across multiple datasets of different modalities, addressing limitations related to working with limited data and diversifying modalities within the medical domain is imperative. Future enhancements will involve refining the model architecture to effectively leverage features extracted from limited data and exploring novel techniques to enhance model robustness and generalizability. Additionally, the scope will expand to encompass other medical images and explore the potential of leveraging multimodal data across various imaging modalities and image classes, offering comprehensive solutions for medical image analysis tasks.

As for patient similarity, future research will focus on advancing similarity computation using self-supervised learning techniques. Building upon the insights gained from the presented method, future endeavors will extend the analysis to encompass a broader spectrum of feature embedding models, enabling deeper insights into their efficacy and applicability to patient similarity tasks. Moreover, efforts will be directed towards integrating advanced techniques such as frequent pattern mining and machine learning methodologies to enhance the accuracy and context-awareness of patient similarity assessments.

Furthermore, in our future research, comprehensive exploration of smart city data and utilization of large multimodal models (LMM) present promising avenues. These models amalgamate diverse data types, thereby enriching information for each field. Introducing novel techniques such as prompt engineering holds the potential for enhancing the performance of large models, guiding them effectively in specialized tasks. Ultimately, these advancements aim to facilitate the efficient management of smart cities, thereby enhancing the well-being of urban residents.

References

[1] U. M. Fayyad, D. Haussler, and P. E. Stolorz, "Kdd for science data analysis: Issues and examples." in *KDD*, 1996, pp. 50–56.

[2] UN-Habitat. (2022) World Cities Report 2022. Accessed on ¡insert date accessed¿. [Online]. Available: https://unhabitat.org/wcr/

[3] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen, "Security and privacy in smart city applications: Challenges and solutions," *IEEE communications magazine*, vol. 55, no. 1, pp. 122–129, 2017.

[4] N. A. Jasim, H. TH, and S. A. Rikabi, "Design and implementation of smart city applications based on the internet of things." *International Journal of Interactive Mobile Technologies*, vol. 15, no. 13, 2021.

[5] M. M. Rashid, J. Kamruzzaman, M. M. Hassan, T. Imam, and S. Gordon, "Cyberattacks detection in iot-based smart city applications using machine learning techniques," *International Journal of environmental research and public health*, vol. 17, no. 24, p. 9347, 2020.

[6] J. S. Saltz and N. Hotz, "Identifying the most common frameworks data science teams use to structure and coordinate their projects," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 2038–2042.

[7] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data," *Journal of Big Data*, vol. 7, pp. 1–19, 2020.

[8] N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola, "Autogluon-tabular: Robust and accurate automl for structured data," *arXiv preprint arXiv:2003.06505*, 2020.

[9] A. Khan, T. Zia, M. Suleman, T. Khan, S. S. Ali, A. A. Abbasi, A. Mohammad, and D.-Q. Wei, "Higher infectivity of the sars-cov-2 new variants is associated with k417n/t, e484k, and n501y mutants: an insight from structural data," *Journal of cellular physiology*, vol. 236, no. 10, pp. 7045–7057, 2021.

[10] K. H. Goh, L. Wang, A. Y. K. Yeow, H. Poh, K. Li, J. J. L. Yeow, and G. Y. H. Tan, "Artificial intelligence in sepsis early prediction and diagnosis using unstructured data in healthcare," *Nature communications*, vol. 12, no. 1, p. 711, 2021.

[11] C. Lynch and P. Sermanet, "Language conditioned imitation learning over unstructured data," *arXiv preprint arXiv:2005.07648*, 2020.

[12] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *International Conference on Management of Data*, 1993, pp. 207–216.

[13] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 2, pp. 423–443, 2018.

[14] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, "Brain tumor segmentation with deep neural networks," *Medical image analysis*, vol. 35, pp. 18–31, 2017.

[15] N. Codella, V. Rotemberg, P. Tschandl, M. E. Celebi, S. Dusza, D. Gutman, B. Helba, A. Kalloo, K. Liopyris, M. Marchetti *et al.*, "Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic)," *arXiv preprint arXiv:1902.03368*, 2019.

[16] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun *et al.*, "Scalable and accurate deep learning with electronic health records," *NPJ digital medicine*, vol. 1, no. 1, pp. 1–10, 2018.

[17] P. Wang, T. Shi, and C. K. Reddy, "Text-to-sql generation for question answering on electronic medical records," in *Proceedings of The Web Conference 2020*, 2020, pp. 350–361.

[18] L. Li, W.-Y. Cheng, B. S. Glicksberg, O. Gottesman, R. Tamler, R. Chen, E. P. Bottinger, and J. T. Dudley, "Identification of type 2 diabetes subgroups through topological analysis of patient similarity," *Science translational medicine*, vol. 7, no. 311, pp. 311ra174–311ra174, 2015.

[19] P. Zhang, F. Wang, J. Hu, and R. Sorrentino, "Towards personalized medicine: leveraging patient similarity and drug similarity analytics," *AMIA Summits on Translational Science Proceedings*, vol. 2014, p. 132, 2014.

[20] J. Lee, D. M. Maslove, and J. A. Dubin, "Personalized mortality prediction driven by electronic medical data and a patient similarity metric," *PloS one*, vol. 10, no. 5, p. e0127428, 2015.

[21] MySQL, "Mysql," https://www.mysql.com/.

[22] PostGres, "Postgres," https://www.postgresql.org/.

[23] BigQuery, "Bigquery," https://cloud.google.com/bigquery.

[24] L. George, *HBase: the definitive guide: random access to your planet-size data*. " O'Reilly Media, Inc.", 2011.

[25] Snowflake, "Snowflake," https://www.snowflake.com/.

[26] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, 1993, pp. 207–216.

[27] R. Agrawal, R. Srikant *et al.*, "Fast algorithms for mining association rules," in *Proc. 20th int. conf. very large data bases*, vol. 1215, 1994, pp. 487–499.

[28] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *ACM sigmod record*, vol. 29, no. 2, pp. 1–12, 2000.

[29] M. Yasir, M. A. Habib, M. Ashraf, S. Sarwar, M. U. Chaudhry, H. Shahwani, M. Ahmad, and C. M. N. Faisal, "D-gene: Deferring the generation of power sets for discovering frequent itemsets in sparse big data," *IEEE Access*, vol. 8, pp. 27 375–27 392, 2020.

[30] M. Yasir, M. A. Habib, M. Ashraf, S. Sarwar, M. U. Chaudhry, H. Shahwani, M. Ahmad, and C. Muhammad Nadeem Faisal, "Trice: Mining frequent itemsets by iterative trimmed transaction lattice in sparse big data," *IEEE Access*, vol. 7, pp. 181 688–181 705, 2019.

[31] M. Yasir, M. A. Habib, S. Sarwar, C. M. N. Faisal, M. Ahmad, and S. Jabbar, "HARPP: harnessing the power of power sets for mining frequent itemsets," *Inf. Technol. Control.*, vol. 48, no. 3, pp. 415–431, 2019. [Online]. Available: https://doi.org/10.5755/j01.itc.48.3.21137

[32] A. Savasere, E. R. Omiecinski, and S. B. Navathe, "An efficient algorithm for mining association rules in large databases," Georgia Institute of Technology, Tech. Rep., 1995.

[33] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *International Conference on Management of Data*. ACM, 2000, pp. 1–12.

[34] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, and D. Yang, "H-mine: Fast and space-preserving frequent pattern mining in large databases," *IIE Transactions*, vol. 39, pp. 593–605, 03 2007.

[35] G. Grahne and J. Zhu, "High performance mining of maximal frequent itemsets," in *6th International workshop on high performance data mining*, vol. 16, 2003, p. 34.

[36] M. J. Zaki, S. Parthasarathy, M. Ogihara, W. Li *et al.*, "New algorithms for fast discovery of association rules." in *KDD*, vol. 97, 1997, pp. 283–286.

[37] M. J. Zaki and C.-J. Hsiao, "Charm: An efficient algorithm for closed itemset mining," in *Proceedings of the 2002 SIAM international conference on data mining*. SIAM, 2002, pp. 457–473.

[38] S. K. Tanbeer, C. F. Ahmed, B.-S. Jeong, and Y.-K. Lee, "Discovering periodic-frequent patterns in transactional databases," in *PAKDD*, 2009, pp. 242–253.

[39] R. U. Kiran, C. Saideep, P. Ravikumar, K. Zettsu, M. Toyoda, M. Kitsuregawa, and P. K. Reddy, "Discovering fuzzy periodic-frequent patterns in quantitative temporal databases," in *2020 (FUZZ-IEEE)*, 2020, pp. 1–8.

[40] R. U. Kiran and P. K. Reddy, "Mining rare periodic-frequent patterns using multiple minimum supports," in *15th International Conference on Management of Data*, 2009, pp. 7–8.

[41] Jiawei Han, Guozhu Dong, and Yiwen Yin, "Efficient mining of partial periodic patterns in time series database," in *ICDE*, 1999, pp. 106–115.

[42] R. U. Kiran, P. Veena, P. Ravikumar, C. Saideep, K. Zettsu, H. Shang, M. Toyoda, M. Kitsuregawa, and P. K. Reddy, "Efficient discovery of partial periodic patterns in large temporal databases," *Electronics*, vol. 11, no. 10, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/10/1523

[43] P. Fournier-Viger, J. C. Lin, Q. Duong, and T. Dam, "PHM: mining periodic high-utility itemsets," in *ICDM*, 2016, pp. 64–79.

[44] R. U. Kiran, M. Kitsuregawa, and P. K. Reddy, "Efficient discovery of periodic-frequent patterns in very large databases," *Journal of Systems and Software*, vol. 112, pp. 110–121, 2016.

[45] S. K. Tanbeer, C. F. Ahmed, B.-S. Jeong, and Y.-K. Lee, "Discovering periodic-frequent patterns in transactional databases," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2009, pp. 242–253.

[46] P. Fournier-Viger, P. Yang, J. C. Lin, and R. U. Kiran, "Discovering stable periodic-frequent patterns in transactional data," in *IEA/AIE*, 2019, pp. 230–244.

[47] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 3, pp. 372–390, 2000.

[48] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur, "Dynamic itemset counting and implication rules for market basket data," in *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, 1997, pp. 255–264.

[49] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. I. Verkamo *et al.*, "Fast discovery of association rules." *Advances in knowledge discovery and data mining*, vol. 12, no. 1, pp. 307–328, 1996.

[50] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 3, pp. 372–390, 2000.

[51] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the eleventh international conference on data engineering*. IEEE, 1995, pp. 3–14.

[52] M. L. Hetland, "A survey of recent methods for efficient retrieval of similar time sequences," in *Data mining in time series databases*. World Scientific, 2004, pp. 23–42.

[53] C.-F. Huang, Y.-C. Chen, and A.-P. Chen, "An association mining method for time series and its application in the stock prices of tft-lcd industry," in *Industrial Conference on Data Mining*. Springer, 2004, pp. 117–126.

[54] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong, and Y.-K. Lee, "Efficient tree structures for high utility pattern mining in incremental databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 12, pp. 1708–1721, 2009.

[55] M. Liu and J. Qu, "Mining high utility itemsets without candidate generation," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 55–64.

[56] R. Chan, Q. Yang, and Y.-D. Shen, "Mining high utility itemsets," in *Third IEEE international conference on data mining*. IEEE Computer Society, 2003, pp. 19–19.

[57] W. Wang, C. Wang, Y. Zhu, B. Shi, J. Pei, X. Yan, and J. Han, "Graphminer: a structural pattern-mining system for large disk-based graph databases and its applications," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 2005, pp. 879–881.

[58] D. J. Cook and L. B. Holder, "Graph-based data mining," *IEEE Intelligent Systems and Their Applications*, vol. 15, no. 2, pp. 32–41, 2000.

[59] R. U. Kiran, J. Venkatesh, P. Fournier-Viger, M. Toyoda, P. K. Reddy, and M. Kitsuregawa, "Discovering periodic patterns in non-uniform temporal databases," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2017, pp. 604–617.

[60] D. Zhang, K. Lee, and I. Lee, "Hierarchical trajectory clustering for spatio-temporal periodic pattern mining," *Expert Systems with Applications*, vol. 92, pp. 1–11, 2018.

[61] R. U. Kiran, A. Anirudh, C. Saideep, M. Toyoda, P. K. Reddy, and M. Kitsuregawa, "Finding periodic-frequent patterns in temporal databases using periodic summaries," *Data Science and Pattern Recognition*, vol. 3, no. 2, pp. 24–46, 2019.

[62] R. U. Kiran, Y. Watanobe, B. Chaudhury, K. Zettsu, M. Toyoda, and M. Kitsuregawa, "Discovering maximal periodic-frequent patterns in very large temporal databases," in *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2020, pp. 11–20.

[63] K. Amphawan, P. Lenca, and A. Surarerks, "Mining top-k periodic-frequent pattern from transactional databases without support threshold," in *Advances in Information Technology*, 2009, pp. 18–29.

[64] R. Uday Kiran and P. Krishna Reddy, "Towards efficient mining of periodic-frequent patterns in transactional databases," in *International Conference on Database and Expert Systems Applications*. Springer, 2010, pp. 194–208.

[65] R. U. Kiran and P. K. Reddy, "An alternative interestingness measure for mining periodic-frequent patterns," in *International Conference on Database Systems for Advanced Applications*. Springer, 2011, pp. 183–192.

[66] P. Ravikumar, P. Likhitha, B. Venus Vikranth Raj, R. Uday Kiran, Y. Watanobe, and K. Zettsu, "Efficient discovery of periodic-frequent patterns in columnar temporal databases," *Electronics*, vol. 10, no. 12, 2021.

[67] R. U. Kiran and M. Kitsuregawa, "Novel techniques to reduce search space in periodic-frequent pattern mining," in *International Conference on Database Systems for Advanced Applications*. Springer, 2014, pp. 377–391.

[68] K. Amphawan, P. Lenca, and A. Surarerks, "Mining top-k periodic-frequent pattern from transactional databases without support threshold," in *International conference on advances in information technology*. Springer, 2009, pp. 18–29.

[69] A. Surana, R. U. Kiran, and P. K. Reddy, "An efficient approach to mine periodic-frequent patterns in transactional databases," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2011, pp. 254–266.

[70] R. He, J. Chen, C. Du, and Y. Duan, "Stable periodic frequent itemset mining on uncertain datasets," in *2021 IEEE 4th International Conference on Computer and Communication Engineering Technology (CCET)*. IEEE, 2021, pp. 263–267.

[71] P. Fournier-Viger, Y. Wang, P. Yang, J. C.-W. Lin, U. Yun, and R. U. Kiran, "Tspin: Mining top-k stable periodic patterns," *Applied Intelligence*, vol. 52, no. 6, pp. 6917–6938, 2022.

[72] R. U. Kiran, "PAMI-PyKit: PAttern MIning-Python Kit," https://github.com/udayRage/pami_pykit/tree/master/traditional, 2020, [Online; accessed 4-June-2020].

[73] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *SIGMOD*, 1993, pp. 207–216.

[74] "KDD-cup-2000," https://kdd.org/kdd-cup/view/kdd-cup-2000, accessed: 2010-09-30.

[75] T. Brijs, "Retail market basket data set," in *Workshop on Frequent Itemset Mining Implementations (FIMI'03)*, 2003.

[76] P. Fournier-Viger, J. C.-W. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, and H. T. Lam, "The spmf open-source data mining library version 2," in *Machine Learning and Knowledge Discovery in Databases*, B. Berendt, B. Bringmann, É. Fromont, G. Garriga, P. Miettinen, N. Tatti, and V. Tresp, Eds. Cham: Springer International Publishing, 2016, pp. 36–40.

[77] M. Woschank, E. Rauch, and H. Zsifkovits, "A review of further directions for artificial intelligence, machine learning, and deep learning in smart logistics," *Sustainability*, vol. 12, no. 9, p. 3760, 2020.

[78] T. Ghazal and H. Alzoubi, "Modelling supply chain information collaboration empowered with machine learning technique," *Intelligent Automation & Soft Computing*, vol. 29, no. 3, pp. 243–257, 2021.

[79] T. Kotsiopoulos, P. Sarigiannidis, D. Ioannidis, and D. Tzovaras, "Machine learning and deep learning in smart manufacturing: The smart grid paradigm," *Computer Science Review*, vol. 40, p. 100341, 2021.

[80] J. Lee, M. Azamfar, J. Singh, and S. Siahpour, "Integration of digital twin and deep learning in cyber-physical systems: towards smart manufacturing," *IET Collaborative Intelligent Manufacturing*, vol. 2, no. 1, pp. 34–36, 2020.

[81] A. Essien and C. Giannetti, "A deep learning model for smart manufacturing using convolutional lstm neural network autoencoders," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6069–6078, 2020.

[82] R. Yan, F. Ren, Z. Wang, L. Wang, T. Zhang, Y. Liu, X. Rao, C. Zheng, and F. Zhang, "Breast cancer histopathological image classification using a hybrid deep neural network," *Methods*, vol. 173, pp. 52–60, 2020.

[83] W. Shen, M. Zhou, F. Yang, C. Yang, and J. Tian, "Multi-scale convolutional neural networks for lung nodule classification," in *Information Processing in Medical Imaging: 24th International Conference, IPMI 2015, Sabhal Mor Ostaig, Isle of Skye, UK, June 28-July 3, 2015, Proceedings 24*. Springer, 2015, pp. 588–599.

[84] K.-L. Hua, C.-H. Hsu, S. C. Hidayati, W.-H. Cheng, and Y.-J. Chen, "Computer-aided classification of lung nodules on computed tomography images via deep learning technique," *OncoTargets and therapy*, pp. 2015–2022, 2015.

[85] F. Liao, M. Liang, Z. Li, X. Hu, and S. Song, "Evaluate the malignancy of pulmonary nodules using the 3-d deep leaky noisy-or network," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3484–3495, 2019.

[86] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[87] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[88] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[89] S. Eslami, G. de Melo, and C. Meinel, "Does clip benefit visual question answering in the medical domain as much as it does in the general domain?" *arXiv preprint arXiv:2112.13906*, 2021.

[90] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification," *Scientific Data*, vol. 10, no. 1, p. 41, 2023.

[91] O. Pelka, S. Koitka, J. Rückert, F. Nensa, and C. M. Friedrich, "Radiology objects in context (roco): a multimodal image dataset," in *Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis: 7th Joint International Workshop, CVII-STENT 2018 and Third International Workshop, LABELS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 3*. Springer, 2018, pp. 180–189.

[92] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros *et al.*, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *jama*, vol. 316, no. 22, pp. 2402–2410, 2016.

[93] Y. Ding, J. H. Sohn, M. G. Kawczynski, H. Trivedi, R. Harnish, N. W. Jenkins, D. Lituiev, T. P. Copeland, M. S. Aboian, C. Mari Aparici *et al.*, "A deep learning model to predict a diagnosis of alzheimer disease by using 18f-fdg pet of the brain," *Radiology*, vol. 290, no. 2, pp. 456–464, 2019.

[94] A. Janowczyk and A. Madabhushi, "Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases," *Journal of pathology informatics*, vol. 7, no. 1, p. 29, 2016.

[95] H. Sharma, N. Zerbe, I. Klempert, O. Hellwich, and P. Hufnagl, "Deep convolutional neural networks for automatic classification of gastric carcinoma using whole slide images in digital histopathology," *Computerized Medical Imaging and Graphics*, vol. 61, pp. 2–13, 2017.

[96] D. S. W. Ting, C. Y.-L. Cheung, G. Lim, G. S. W. Tan, N. D. Quang, A. Gan, H. Hamzah, R. Garcia-Franco, I. Y. San Yeo, S. Y. Lee *et al.*, "Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations with diabetes," *Jama*, vol. 318, no. 22, pp. 2211–2223, 2017.

[97] R. Ribani and M. Marengoni, "A survey of transfer learning for convolutional neural networks," in *2019 32nd SIBGRAPI conference on graphics, patterns and images tutorials (SIBGRAPI-T)*. IEEE, 2019, pp. 47–57.

[98] D. Sarkar, R. Bali, and T. Ghosh, *Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras*. Packt Publishing Ltd, 2018.

[99] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in neural information processing systems*, vol. 27, 2014.

[100] O. Hadad, R. Bakalo, R. Ben-Ari, S. Hashoul, and G. Amit, "Classification of breast lesions using cross-modal deep learning," in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. IEEE, 2017, pp. 109–112.

[101] S. Saxena, S. Shukla, and M. Gyanchandani, "Pre-trained convolutional neural networks as feature extractors for diagnosis of breast cancer using histopathology," *International Journal of Imaging Systems and Technology*, vol. 30, no. 3, pp. 577–591, 2020.

[102] A. Rakhlin, A. Shvets, V. Iglovikov, and A. A. Kalinin, "Deep convolutional neural networks for breast cancer histology image analysis," in *Image Analysis and Recognition: 15th International Conference, ICIAR 2018, Póvoa de Varzim, Portugal, June 27–29, 2018, Proceedings 15*. Springer, 2018, pp. 737–744.

[103] A. Mahbod, I. Ellinger, R. Ecker, Ö. Smedby, and C. Wang, "Breast cancer histological image classification using fine-tuned deep network fusion," in *Image Analysis and Recognition: 15th International Conference, ICIAR 2018, Póvoa de Varzim, Portugal, June 27–29, 2018, Proceedings 15*. Springer, 2018, pp. 754–762.

[104] A. Acevedo, S. Alférez, A. Merino, L. Puigví, and J. Rodellar, "Recognition of peripheral blood cell images using convolutional neural networks," *Computer methods and programs in biomedicine*, vol. 180, p. 105020, 2019.

[105] Y. Wang, E. J. Choi, Y. Choi, H. Zhang, G. Y. Jin, and S.-B. Ko, "Breast cancer classification in automated breast ultrasound using multiview convolutional neural network with transfer learning," *Ultrasound in medicine & biology*, vol. 46, no. 5, pp. 1119–1132, 2020.

[106] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[107] G. Liang and L. Zheng, "A transfer learning method with deep residual network for pediatric pneumonia diagnosis," *Computer methods and programs in biomedicine*, vol. 187, p. 104964, 2020.

[108] A. Tiwari, S. Srivastava, and M. Pant, "Brain tumor segmentation and classification from magnetic resonance images: Review of selected methods from 2014 to 2019," *Pattern Recognition Letters*, vol. 131, pp. 244–260, 2020.

[109] M. Salvi, U. R. Acharya, F. Molinari, and K. M. Meiburger, "The impact of pre-and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis," *Computers in Biology and Medicine*, vol. 128, p. 104129, 2021.

[110] H. N. Dao, Q. T. Nguyen, C. Mugisha, and I. Paik, "A multimodal transfer learning approach for medical image classification," in *2023 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE, 2023.

[111] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[112] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[113] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[114] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[115] D. Varshni, K. Thakral, L. Agarwal, R. Nijhawan, and A. Mittal, "Pneumonia detection using cnn based feature extraction," in *2019 IEEE international conference on electrical, computer and communication technologies (ICECCT)*. IEEE, 2019, pp. 1–7.

[116] T. Kaur and T. K. Gandhi, "Automated brain image classification based on vgg-16 and transfer learning," in *2019 International Conference on Information Technology (ICIT)*. IEEE, 2019, pp. 94–98.

[117] H. N. Dao, Q. T. Nguyen, and I. Paik, "Transfer learning for medical image classification on multiple datasets using pubmedclip," in *2022 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE, 2022.

[118] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.

[119] Z. Zhou, J. Shin, L. Zhang, S. Gurudu, M. Gotway, and J. Liang, "Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7340–7351.

[120] J. Li, G. Zhao, Y. Tao, P. Zhai, H. Chen, H. He, and T. Cai, "Multi-task contrastive learning for automatic ct and x-ray diagnosis of covid-19," *Pattern recognition*, vol. 114, p. 107848, 2021.

[121] J. Gan, L. Xiang, Y. Zhai, C. Mai, G. He, J. Zeng, Z. Bai, R. D. Labati, V. Piuri, and F. Scotti, "2m beautynet: Facial beauty prediction based on multi-task transfer learning," *IEEE Access*, vol. 8, pp. 20 245–20 256, 2020.

[122] P. Zhang, J. Li, Y. Wang, and J. Pan, "Domain adaptation for medical image segmentation: a meta-learning method," *Journal of Imaging*, vol. 7, no. 2, p. 31, 2021.

[123] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, 2018.

[124] M. A. Morid, A. Borjali, and G. Del Fiol, "A scoping review of transfer learning research on medical image analysis using imagenet," *Computers in biology and medicine*, vol. 128, p. 104115, 2021.

[125] M. Shaban, R. Awan, M. M. Fraz, A. Azam, Y.-W. Tsang, D. Snead, and N. M. Rajpoot, "Context-aware convolutional neural network for grading of colorectal cancer histology images," *IEEE transactions on medical imaging*, vol. 39, no. 7, pp. 2395–2405, 2020.

[126] Y. Eroğlu, M. Yildirim, and A. Cinar, "Convolutional neural networks based classification of breast ultrasonography images by hybrid method with respect to benign, malignant, and normal using mrmr," *Computers in biology and medicine*, vol. 133, p. 104407, 2021.

[127] E. F. Ohata, J. V. S. d. Chagas, G. M. Bezerra, M. M. Hassan, V. H. C. de Albuquerque, and P. P. R. Filho, "A novel transfer learning approach for the classification of histological images of colorectal cancer," *The Journal of Supercomputing*, pp. 1–26, 2021.

[128] Y. Jiménez Gaona, M. J. Rodriguez-Alvarez, H. Espino-Morato, D. Castillo Malla, and V. Lakshminarayanan, "Densenet for breast tumor classification in mammographic images," in *International Conference on Bioengineering and Biomedical Signal and Image Processing*. Springer, 2021, pp. 166–176.

[129] A. Kallipolitis, K. Revelos, and I. Maglogiannis, "Ensembling efficientnets for the classification and interpretation of histopathology images," *Algorithms*, vol. 14, no. 10, p. 278, 2021.

[130] S. Sharma, S. Gupta, D. Gupta, S. Juneja, P. Gupta, G. Dhiman, S. Kautish *et al.*, "Deep learning model for the automatic classification of white blood cells," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[131] C. Chola, A. Y. Muaad, M. B. Bin Heyat, J. B. Benifa, W. R. Naji, K. Hemachandran, N. F. Mahmoud, N. A. Samee, M. A. Al-Antari, Y. M. Kadah *et al.*, "Bc-net: A deep learning computer-aided diagnosis framework for human peripheral blood cell identification," *Diagnostics*, vol. 12, no. 11, p. 2815, 2022.

[132] Z. Jafari and E. Karami, "Breast cancer detection in mammography images: A cnn-based approach with feature selection," 2023.

[133] T.-M. H. Hsu, W.-H. Weng, W. Boag, M. McDermott, and P. Szolovits, "Unsupervised multimodal representation learning across medical images and reports," *arXiv preprint arXiv:1811.08615*, 2018.

[134] G. Chauhan, R. Liao, W. Wells, J. Andreas, X. Wang, S. Berkowitz, S. Horng, P. Szolovits, and P. Golland, "Joint modeling of chest radiographs and radiology reports for pulmonary edema assessment," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part II 23*. Springer, 2020, pp. 529–539.

[135] S.-C. Huang, L. Shen, M. P. Lungren, and S. Yeung, "Gloria: A multimodal global-local representation learning framework for label-efficient medical image recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3942–3951.

[136] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya *et al.*, "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 590–597.

[137] A. E. Johnson, T. J. Pollard, N. R. Greenbaum, M. P. Lungren, C.-y. Deng, Y. Peng, Z. Lu, R. G. Mark, S. J. Berkowitz, and S. Horng, "Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs," *arXiv preprint arXiv:1901.07042*, 2019.

[138] Y. Li, H. Wang, and Y. Luo, "A comparison of pre-trained vision-and-language models for multimodal representation learning across medical images and reports," in *2020 IEEE international conference on bioinformatics and biomedicine (BIBM)*. IEEE, 2020, pp. 1999–2004.

[139] Z. Zhang, P. Chen, X. Shi, and L. Yang, "Text-guided neural network training for image recognition in natural scenes and medicine," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 5, pp. 1733–1745, 2019.

[140] N. Mu, A. Kirillov, D. Wagner, and S. Xie, "Slip: Self-supervision meets language-image pre-training," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*. Springer, 2022, pp. 529–544.

[141] M. V. Conde and K. Turgutlu, "Clip-art: Contrastive pre-training for fine-grained art classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3956–3960.

[142] M. Barraco, M. Cornia, S. Cascianelli, L. Baraldi, and R. Cucchiara, "The unreasonable effectiveness of clip features for image captioning: an experimental analysis," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4662–4670.

[143] D. N. Hong and I. Paik, "Patient similarity using electronic health records and self-supervised learning," in *Proceedings of IEEE McSOC 2023 Conference*. Singapore: IEEE, 2023.

[144] Z. Yu, J. Yu, J. Fan, and D. Tao, "Multi-modal factorized bilinear pooling with co-attention learning for visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1821–1830.

[145] C. Zhang, Z. Yang, X. He, and L. Deng, "Multimodal intelligence: Representation learning, information fusion, and applications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 3, pp. 478–493, 2020.

[146] D. Sharma, S. Purushotham, and C. K. Reddy, "Medfusenet: An attention-based multimodal deep learning model for visual question answering in the medical domain," *Scientific Reports*, vol. 11, no. 1, p. 19826, 2021.

[147] J. N. Kather, N. Halama, and A. Marx, "100,000 histological images of human colorectal cancer and healthy tissue," Apr. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1214456

[148] W. Al-Dhabyani, M. Gomaa, H. Khaled, and A. Fahmy, "Dataset of breast ultrasound images," *Data in brief*, vol. 28, p. 104863, 2020.

[149] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[150] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," *arXiv preprint arXiv:1606.01847*, 2016.

[151] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome, "Mutan: Multimodal tucker fusion for visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2612–2620.

[152] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis," *IEEE journal of biomedical and health informatics*, vol. 22, no. 5, pp. 1589–1604, 2017.

[153] M. Jiang, Y. Chen, M. Liu, S. T. Rosenbloom, S. Mani, J. C. Denny, and H. Xu, "A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 601–606, 2011.

[154] D. Zhao and C. Weng, "Combining pubmed knowledge and ehr data to develop a weighted bayesian network for pancreatic cancer prediction," *Journal of biomedical informatics*, vol. 44, no. 5, pp. 859–868, 2011.

[155] G. J. Kuperman, A. Bobb, T. H. Payne, A. J. Avery, T. K. Gandhi, G. Burns, D. C. Classen, and D. W. Bates, "Medication-related clinical decision support in computerized provider order entry systems: a review," *Journal of the American Medical Informatics Association*, vol. 14, no. 1, pp. 29–40, 2007.

[156] A. E. Johnson, L. Bulgarelli, L. Shen, A. Gayles, A. Shammout, S. Horng, T. J. Pollard, S. Hao, B. Moody, B. Gow *et al.*, "Mimic-iv, a freely accessible electronic health record dataset," *Scientific data*, vol. 10, no. 1, p. 1, 2023.

[157] S.-A. Brown, "Patient similarity: emerging concepts in systems and precision medicine," *Frontiers in physiology*, vol. 7, p. 561, 2016.

[158] J. Patricoski, K. Kreimeyer, A. Balan, K. Hardart, J. Tao, V. Anagnostou, T. Botsis, J. H. M. T. B. Investigators *et al.*, "An evaluation of pretrained bert models for comparing semantic similarity across unstructured clinical trial texts," *Stud Health Technol Inform*, vol. 289, pp. 18–21, 2022.

[159] H. Memarzadeh, N. Ghadiri, M. Samwald, and M. L. Shahreza, "A study into patient similarity through representation learning from medical records." *Knowledge and Information Systems*, vol. 64.12, pp. 3293–3324, 2022.

[160] Z. Zhu, C. Yin, B. Qian, Y. Cheng, J. Wei, and F. Wang, "Measuring patient similarities via a deep architecture with medical concept embedding," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 749–758.

[161] Q. Suo, F. Ma, Y. Yuan, M. Huai, W. Zhong, J. Gao, and A. Zhang, "Deep patient similarity learning for personalized healthcare," *IEEE transactions on nanobioscience*, vol. 17, no. 3, pp. 219–227, 2018.

[162] S. Pokharel, X. Li, X. Zhao, A. Adhikari, and Y. Li, "Similarity computing on electronic health records." in *PACIS*, 2018, p. 198.

[163] A. Gottlieb, G. Y. Stein, E. Ruppin, R. B. Altman, and R. Sharan, "A method for inferring medical diagnoses from patient similarities," *BMC medicine*, vol. 11, pp. 1–10, 2013.

[164] S. Ebadollahi, J. Sun, D. Gotz, J. Hu, D. Sow, and C. Neti, "Predicting patient's trajectory of physiological data using temporal trends in similar patients: a system for near-term prognostics," in *AMIA annual symposium proceedings*, vol. 2010. American Medical Informatics Association, 2010, p. 192.

[165] C. Che, C. Xiao, J. Liang, B. Jin, J. Zho, and F. Wang, "An rnn architecture with dynamic temporal matching for personalized predictions of parkinson's disease," in *Proceedings of the 2017 SIAM international conference on data mining*. SIAM, 2017, pp. 198–206.

[166] J. Sun, F. Wang, J. Hu, and S. Edabollahi, "Supervised patient similarity measure of heterogeneous patient records," *Acm Sigkdd Explorations Newsletter*, vol. 14, no. 1, pp. 16–24, 2012.

[167] M. Zhan, S. Cao, B. Qian, S. Chang, and J. Wei, "Low-rank sparse feature selection for patient similarity learning," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 1335–1340.

[168] A. Sharafoddini, J. A. Dubin, J. Lee *et al.*, "Patient similarity in prediction models based on health data: a scoping review," *JMIR medical informatics*, vol. 5, no. 1, p. e6730, 2017.

[169] Y. Sha, J. Venugopalan, and M. D. Wang, "A novel temporal similarity measure for patients based on irregularly measured data in electronic health records," in *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2016, pp. 337–344.

[170] C. Mugisha and I. Paik, "Comparison of neural language modeling pipelines for outcome prediction from unstructured medical text notes," *IEEE Access*, vol. 10, pp. 16 489–16 498, 2022.

[171] S. Pokharel, G. Zuccon, X. Li, C. P. Utomo, and Y. Li, "Temporal tree representation for similarity computation between medical patients," *Artificial Intelligence in Medicine*, vol. 108, p. 101900, 2020.

[172] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[173] E. R. Dubberke, K. A. Reske, L. C. McDonald, and V. J. Fraser, "Icd-9 codes and surveillance for clostridium difficile–associated disease," *Emerging infectious diseases*, vol. 12, no. 10, p. 1576, 2006.

[174] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*. PMLR, 2014, pp. 1188–1196.

[175] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[176] e. a. Mikolov, Tomas, "Advances in neural information processing systems," *Distributed representations of words and phrases and their compositionality: 3111-3119*, 2013.

[177] L. Dai, H. Zhu, and D. Liu, "Patient similarity: methods and applications," *arXiv preprint arXiv:2012.01976*, 2020.

[178] S. Pokharel, G. Zuccon, X. Li, C. P. Utomo, and Y. Li, "Temporal tree representation for similarity computation between medical patients," *Artificial Intelligence in Medicine*, vol. 108, p. 101900, 2020.