Privacy Preserving Machine Learning for Next Day Pain Prediction

Ashutosh Vilas Engavle

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2024

Committee:

Martine De Cock

Anderson Nascimento

Weichao Yuwen

Program Authorized to Offer Degree:
Computer Science and Systems

University of Washington

Abstract

Privacy Preserving Machine Learning for
Next Day Pain Prediction

Ashutosh Vilas Engavle

Chair of the Supervisory Committee:
Martine De Cock
School of Engineering and Technology

The availability of healthcare data is critically limited due to stringent privacy regulations, ethical considerations, and the intrinsic sensitivity of medical information. This scarcity hampers research and development in medical science, ultimately affecting the advancement of healthcare services and patient outcomes. Synthetic data emerges as a potent solution to this challenge, offering a pathway to bolster data accessibility while safeguarding patient privacy. This thesis explores the multifaceted issue of next day pain prediction with machine learning models and the limited availability of patient data to train such models, and delves into the potential of synthetic data to bridge this gap. By generating realistic, non-personal data that mimics the statistical properties of real healthcare datasets, synthetic data provides a viable alternative for research and analysis, circumventing privacy concerns. We use methodologies for synthetic data generation with and without privacy, and evaluate their effectiveness and utility for next day pain prediction in patients with Juvenile Idiopathic Arthritis and lupus. We compare the utility of synthetic data with that of real data by training models on both kinds of data and evaluating the trained models on real data. The findings indicate that machine learning models are able to do next day pain prediction. We also see that marginal based synthetic data generation methods can create synthetic data with good utility with substantial privacy guarantee for this task.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to extend my deepest gratitude to a number of individuals whose support and guidance have been invaluable throughout the course of this endeavor.

First and foremost, I must express my profound appreciation to my parents. Their unwavering belief in my abilities, endless patience, and constant encouragement have been the bedrock of my resilience and determination. Their sacrifices and unconditional love have shaped me in ways words can scarcely capture.

I am immensely grateful to Professor Martine De Cock, whose expertise and insightful guidance have been instrumental in navigating the complexities of my research. Her mentorship extended beyond academic instruction, offering life lessons that I will carry with me. Professor Martine, your passion for Privacy Preserving Machine Learning is truly inspiring, and I am honored to have had the opportunity to learn from you.

To my research group members, your collaboration and camaraderie have made this journey not only educational but also enjoyable. Each discussion, brainstorming session, and challenge we faced together contributed significantly to my personal and professional growth. Your diverse perspectives and rigorous critiques have enriched this work in countless ways.

I would also like to acknowledge Professor Anderson Nascimento and Professor Weichao Yuwen. Your contributions were essential to the completion of this project.

Lastly, I extend my gratitude to the University of Washington, whose resources and support were invaluable throughout my research.

This accomplishment is not solely my own, but a testament to the collective effort, support, and encouragement of everyone mentioned and many more who have touched this journey in various capacities. Thank you from the bottom of my heart.

DEDICATION

This work is dedicated

To my family, whose unwavering support has been my foundation

To my mentors, who have guided me with patience and wisdom

To the pursuit of a better future, aspiring for a future where technology empowers without compromising individual integrity.

Chapter 1

INTRODUCTION

In this thesis, we tackle two interconnected challenges in the realm of healthcare machine learning: firstly, the development of accurate machine learning models for next-day pain prediction for patients suffering with JIA[1] and lupus[2]; and secondly, overcoming the hurdle of data availability for such predictive models due to the prevalent data logjam in healthcare. This data logjam, largely due to privacy concerns and stringent legislation, restricts access to valuable clinical data for research purposes. Addressing this, we explore the use of Synthetic Data Generation (SDG) as a Privacy-Enhancing Technology (PET) to facilitate wider data accessibility without compromising patient confidentiality.

1.1 Background

Chronic diseases remain the leading cause of death and disability in the U.S., with six in 10 adults having a chronic disease, and four in 10 having two or more.[3] These conditions often present with both disease-specific and common symptoms like pain, sleep disturbances, and mood disorders. Despite the effectiveness of self-management educational programs in reducing symptom burdens, chronic diseases continue to demand a significant portion of healthcare resources, accounting for 90% of the nation's $4.1 trillion annual healthcare expenditures [6].

Machine learning (ML) is increasingly used to enhance diagnosis, treatment, self-management, and prevention of chronic diseases, ultimately leading to healthier lives of higher quality. This

---

[1] `https://en.wikipedia.org/wiki/Juvenile_idiopathic_arthritis`

[2] `https://en.wikipedia.org/wiki/Lupus`

[3] `https://www.cdc.gov/chronicdisease/about/index.htm`

thesis specifically addresses the development of ML systems capable of predicting symptoms before their onset, aiding in preemptive healthcare measures. A key focus will be on chronic inflammatory disease in children and adolescents, where early prediction of pain levels is vital for effective disease management.

Machine learning offers promising avenues for such predictive analytics, yet its potential is often hampered by limited access to large-scale, diverse patient data. Addressing this, we delve into the root causes of the healthcare data logjam, emphasizing the pivotal role of privacy concerns and regulatory barriers. Drawing insights from key resources and recent explorations of SDG's role in healthcare and pertinent literature on privacy-preserving technologies, we investigate to what extent state-of-the-art SDG techniques can be used to generate synthetic data that is suitable for pain prediction modeling.

1.2 Research Focus

This thesis pivots towards a novel approach in ML model development for next-day symptom prediction, particularly addressing the issue of generating synthetic data for medical datasets while retaining utility with an emphasis on using synthetic data to circumvent the challenges posed by restricted access to clinical data.

Firstly, we will integrate various data sources, including medical records, wearable device outputs, and diary entries. These diverse sources offer comprehensive insights into patient sleep patterns, physical activity, and pain levels. Advanced ML algorithms, such as regression tree ensembles and support vector machines, will be employed for data analysis and prediction. A significant portion of the work will be dedicated to data understanding and feature engineering, essential for minimizing prediction errors in ML models.

Secondly, and more crucially, we propose the creation of synthetic datasets to replace these real data sources. Synthetic data, artificially generated yet resembling real-world data, is increasingly recognized for its potential in ML (see e.g. [4, 14]). These datasets will be created using cutting-edge generative algorithms, including marginal-based methods (MST), Generative Adversarial Networks (GANs) and Large Language Models (distilgpt2). This

synthetic data could be used by other researchers with minimal or no approvals required.

The efficacy of these methods will be rigorously tested on patient data across 3 datasets, namely SIPA [4] [5], Todd [19] and SLE [16] datasets, the former two of which are based on Juvenile Idiopathic Arthritis (JIA) while the latter is based on systemic lupus erythematosus (SLE), evaluating the accuracy and robustness of the models against actual pain levels. This approach could significantly enhance pain management for patients, improving their quality of life. This study aims to develop effective pain prediction machine learning models while also comparing the utility and privacy trade offs for different synthetic data generation techniques for this machine learning task for SLE. A detailed analysis of SDG on JIA data is not included because preliminary research did not yield good results, which we believe is due to the small size of the JIA data.

1.3 Novelty and Impact

The introduction of synthetic data in this research marks a fresh approach to ensure patient privacy in pain prediction for chronic diseases. Also, accurate pain prediction could revolutionize healthcare provision, allowing for tailored medication dosages, timely pain-relief therapies, and optimized patient care. Overall, this study holds the potential to significantly improve chronic pain management and improvement in healthcare research methodology.

Our contributions are:

- The design and comparison of various machine learning models for next day pain prediction, trained on patient data from multiple sources, including diary and actigraphy data.

- A significant amount of work in feature engineering for patient diary and actigraphy data, leading to an increase in accuracy of next day pain prediction.

---

[4] `https://depts.washington.edu/sipa/`

[5] `https://www.clinicaltrials.gov/study/NCT04354337`

- The generation of synthetic data which achieves similar accuracy using CTGAN and distilgpt.

- The generation of privacy preserving synthetic data using MST and MWEM PGM.

The remainder of this thesis is structured as follows:

- Chapter 2 goes into related works on machine learning for symptom prediction, machine learning for actigraphy data, and synthetic data generation.

- Chapter 3 goes over the 3 datasets used in this study. Various feature extraction and feature engineering steps are explained. In addition, we also define the machine learning task.

- Chapter 4 starts with some advanced feature construction. we define the machine learning models and the general parameters used in this study. In the end, we discuss various synthetic data generation algorithms.

- Chapter 5 goes over results for all 3 datasets and a merged dataset over SIPA and Todd. We compare various synthetic data generation techniques on the SLE dataset.

- Chapter 6 gives the conclusion and suggests some interesting directions for future work.

Chapter 2

RELATED WORK

This section outlines existing work pertinent to our thesis, spanning three dimensions and focusing mainly on number 3: (1) machine learning applications in symptom prediction, (2) utilization of actigraphy data in machine learning models, and (3) advancements in synthetic data generation, particularly in patient data.

2.1 Machine Learning for Symptom Prediction

There have been extensive works on machine learning for symptom prediction. Matsangidou et al. conducted a comprehensive survey of 26 studies on pain prediction using machine learning, finding algorithms like boosting, regression, and neural networks to be highly effective [10]. Unlike these studies, our work incorporates actigraphy data in the machine learning models to predict pain, representing a novel integration of sleep-related data for this purpose.

Yuwen et al.'s research on juvenile idiopathic arthritis (JIA) offers valuable insights into the variables impacting pain, using statistical analysis of actigraphy and diary data [19]. However, this study did not leverage machine learning for pain prediction, which is a key focus of our thesis.

Current work on Arthritis includes work by Loetsch et al. [9] where they use Random Forest Classifier to get feature importance and their analysis seems to divide JIA patients in 3 groups of low, medium and high pain based on their features instead of everyday pain prediction. Another work is by Ho et al. [7] who use thermal images of the joints of the hand of Rheumatoid Arthritis (RA) patients for classification using bagging, J48 (a type of decision tree algorithm), and AdaBoost classifiers.

2.2 Machine Learning with Actigraphy Data

Actigraphy data, tracking activity levels at fine-grained intervals, is a rich source for inferring sleep patterns and wake states. Khademi et al. provide a valuable reference for extracting relevant features from actigraphy data [8], which was useful to process actigraphy data. Given our focus on synthetic data generation, we decided not to use personalized approach in our research for predicting future symptoms.

While Ananth et al.'s [1] study on sleep apps and Sathyanarayana et al.'s [15] work on predicting sleep quality using deep learning approaches highlight the challenges and potential of using actigraphy data. We also use deep learning specifically transformer based methods to generate synthetic data without privacy as a comparison. Our analysis is also different as we focus on predicting next day pain instead of sleep quality. Our study focuses on patients suffering with JIA and SLE.

2.3 Synthetic Data Generation

The emerging field of Generative AI, notable in text (ChatGPT [1]), image generation (DALL·E [2]) and even video generation(Sora [3]), provides a framework for our thesis's focus on Synthetic Data Generation (SDG) for patient data. Unlike generative AI for content creation, our objective is to address the privacy of clinical data while ensuring privacy using generative methods.

The pivotal role of synthetic data in healthcare is its potential to address the significant challenges of data sharing and privacy while fostering clinical and scientific research [11]. Due to stringent privacy regulations like HIPAA and GDPR, and internal policies prioritizing patient confidentiality, the sharing of patient data within the healthcare sector is notably restricted and complex. This situation is further compounded by the proprietary treatment

[1] `https://openai.com/blog/chatgpt/`

[2] `https://openai.com/dall-e-2/`

[3] `https://openai.com/sora`

of data by healthcare organizations and a general lack of incentive for data sharing within the medical research community, which is hindered by misaligned professional incentives and privacy concerns. Despite these challenges, synthetic data emerges as a promising solution by enabling the generation of data that maintains privacy by blending characteristics across individuals, thereby mitigating the risk of re-identification inherent in de-identified patient data. This approach not only facilitates data sharing and collaboration across commercial and academic entities but also paves the way for innovative applications in clinical diagnostics and decision support systems, without compromising patient privacy.

In light of Giuffrè and Shung's insights on synthetic data [5], it becomes clear that the integration of AI in healthcare necessitates a balanced approach that addresses biases and prioritizes privacy through mechanisms like differential privacy and a digital chain of custody. Their emphasis on the need for collaborative efforts in shaping regulations highlights the collective responsibility of the healthcare ecosystem to ensure the ethical use of synthetic data.

The synthetic data generation techniques we explore are MST, MWEM-PGM, CTGAN and LLM transformers (distilgpt) and are further explained in chapter 4.

These studies collectively form the foundation of this thesis, where we aim to leverage synthetic data to address the challenges of privacy and regulations.

Chapter 3

DATA AND PROBLEM DESCRIPTION

3.1 Introduction

Before diving into the specifics, it is pertinent to clarify the kind of data available. The study relies on two kinds of primary data: diary data, which captures the experiences and metrics noted down by the parents or guardians of the patients, and actigraphy data, which records the physical activity of the patients. Both kinds of data are structured around a unique identifier known as *study_id*, which aids in collating and merging data.

3.2 Raw Data Description

3.2.1 SIPA Data [1]

The SIPA ("Sleep Innovations for Preschoolers With Arthritis") dataset contains data for 19 patients from two sources: actigraphy data and diary data. The actigraphy data is recorded every 30 seconds during a period of 10 to 12 days (the exact number of days varies per patient), resulting in 28,800 to 34,560 rows of actigraphy measurements in total. The diary data contains some demographic information as well as entries that are filled out by the parents or guardians for each night. The pain level of the child when the child wakes up is manually recorded in the diary by the parents as an integer on a scale from 1 to 100. Table 3.1 contains general statistics, which are explained in further detail below.

[1] `https://www.clinicaltrials.gov/study/NCT04354337`

Characteristic	**Details**
# Patients	19
# Studies Conducted (Some patients participated twice)	23
# Actigraphy summary per patient	10-12 days
# Actigraphy measurements per patient	Measured every 30 seconds for 10-12 days
# Diary entries per patient	7-14 days with most having 10-14 days

Table 3.1: SIPA dataset characteristics

- **Diary Data:** The diary data collection process is fairly straightforward. The data for all participants is stored in a single CSV file. The *study_id* column within this file allows us to distinguish data between different participants. Table 3.2 shows some of the useful data in the Diary.

Input Column	**Description of Data**	**Method of collection**
Date	Date of measurement	Once per day
Time	Time of measurement	Once per day
Your age in years	Age of parent / person who did survey	At the start of study
Your race or ethnicity	Race / ethnicity of parent	At the start of study
Education level USA type	Education level of parent	At the start of study

Continued on next page

Table 3.2 – continued from previous page

Input Column	**Description of Data**	**Method of collection**
Your employment status	Employment status of parent	At the start of study
Your marital or partner status	Marital status of parent	At the start of study
Which category represents your total household income for the past 12 months?	Household income category	At the start of study
Your child's age (in years)	Age of child	At the start of study
Child gender	Gender of child	At the start of study
When was your child first diagnosed with JIA?	Age of child when they received diagnosis	At the start of study
Does your child currently take any medication/s?	Yes/No answer to taking medicine	Once per day
Medicine	Name(s) of medicines taken	Once per day
Dose	Dose of medicine taken	Once per day
How often?	How often does the child take medicine	Once per day
What time did your child first get into bed last night?	Time when child went to bed	Once per day

Continued on next page

Table 3.2 – continued from previous page

Input Column	**Description of Data**	**Method of collection**
What time did your child fall asleep last night?	Time when child fell asleep	Once per day
How many times did your child wake up during the night after falling asleep?	Times child woke up during the night	Once per day
Please rate your child's sleep quality for last night	Sleep quality of child as rated by parent	Once per day

Table 3.2: Diary features in the SIPA dataset

- **Actigraphy Data:** The actigraphy data, as shown in table 3.3, is a bit more intricate. For every participant, there exists a unique CSV file, with the file name containing the respective *study_id*. This file provides two main types of data:

Input Column	**Description of Data**
Date	Date of measurement. Represents the day on which the measurements were taken.
Activity	Activity level indicating the physical movement intensity. A higher value denotes more intense physical activity.
White Light	Level of ambient white light. Represents the brightness level from natural or artificial sources.

Continued on next page

Table 3.3 – continued from previous page

Input Column	**Description of Data**
Red Light	Level of ambient red light. Indicates the intensity of red wavelengths in the surrounding environment.
Green Light	Level of ambient green light. Indicates the intensity of green wavelengths in the surrounding environment.
Blue Light	Level of ambient blue light. Indicates the intensity of blue wavelengths in the surrounding environment.
Sleep/Wake	Binary indicator: 0 implies the subject is sleeping and 1 implies they are awake.
Interval Status	Describes the current status in the given time interval. Categories include: Active (engaged in physical activity), Rest (awake but resting), Sleep (asleep), or Excluded (data not considered). The difference between this and "Sleep/Wake" is that this offers more granularity on the activity status beyond just sleep or wakefulness.

Table 3.3: Detailed descriptions of actigraphy features in the SIPA dataset. These columns are present in both Actigraphy Summary and Actigraphy Raw Data.

- **Actigraphy Summary Data:** This data is generated by a specialized software that processes the raw actigraphy data. It provides a summarized view of the night data for each day. For example, the day on Oct 14 will have the data from Oct 13 night to Oct 14 morning. This data is created using Respironics actigraphy software.[2]
- **Actigraphy Raw Data:** This data consists of actigraphy measurements cap-

[2] `https://www.actigraphy.respironics.com/solutions/actigraphy.aspx`. Each actigraphy file has 1 row of this type of data for each day.

tured at 30-second intervals. Each actigraphy file has multiple rows (up to a maximum of 2880 rows) of this type of data for each day recorded every 30 seconds.

- **Actigraphy Day Data:** This data isn't given in the actigraphy file but rather generated from the Actigraphy Raw Data. To derive this data, the raw actigraphy measurements from the period of 10 am on day d to 8 pm on day d are used. This provides a comprehensive view of the participant's activity during the primary active hours of the day. We need to stop at 8 pm as we need to generate output for intervention at that time. The data is summed up for the time period and then divided by the total number of measurements so we get the average. We am using the day data to check if daytime activity on day d has any correlation to the symptoms of the patient on day d+1. Please refer to "Diary WakeUpPain + Day" row in table 5.2 for the results.

Column Renaming: Table 3.4 provides explanations for the renaming of certain diary columns, which has been done to simplify their usage and enhance comprehension.

Merging: The first step for merging involves sifting through each file, with each participant uniquely associated with a *study_id*. This identifier corresponds to a specific set of measurements for a participant's duration in the study. Our primary objective is to seamlessly merge the diary data with the actigraphy data using this *study_id* as a reference.

Given the nature of the analysis and specific requirements, we can merge various combinations of data. The merging process hinges on the date of the entry to maintain uniformity. When merging actigraphy data with Diary, 3.5 shows how the data is merged. Since the pain prediction is done early in the morning and we can only use the current day actigraphy day data, current day diary data and the previous day night (summary) data to do our predictions. The pain value is taken from the next morning i.e. day $d + 1$.

When we merge diary data with actigraphy, we lose 63 data points out of the 258 in diary due to participants not wearing the device during those days usually at the start or end of

Original Column Name	**New Column Name**
What is the date you are filling this out for?	Date
Please rate your child's pain at the time your child woke up.	WakeUpPain
Please rate your child sleep quality for last night	SleepQuality
Please rate your child mood at the time your child woke up.	WakeUpMood
Your gender	ParentGender
Your child's race or ethnicity	PatientRace
Your marital or partner status	ParentsMaritalStatus
Which category represents your total household income for the past 12 months?	Income
Your child's age (in years)	PatientAge
Child gender	PatientGender
Did your child have any caffeinated beverages with dinner, or after dinner?	PatientCaffeine
Was your child sick or ill today?	PatientHealth
Did your child take any medications today?	PatientMedicineTaken

Table 3.4: Renaming of columns in Diary data. This table illustrates the transformation of column names from their original verbose state to a more succinct version for ease of data manipulation.

the study between actigraphy and diary ending up with 195 data points after merging as shown in table 3.9.

It is essential to highlight that the merging process uses both *study_id* and dates. This is primarily because 8 participants have engaged in the study on multiple occasions, resulting in numerous actigraphy files. In these instances, they are considered as separate entities, akin to different patients.

Source	**Data**
Day d-1 Actigraphy Night Summary Data	Columns
Day d Actigraphy Day Data	Columns
Day d Diary Data	Columns
Day d+1 WakeUpPain (From Diary)	Value

Table 3.5: Instance creation: to create a labeled instance for participant p for day d in the dataset, we combine columns for p on day $d-1$ from the actigraphy data with columns for p on day d from the diary data."

Non-time dependent features, such as the gender, are added to each row of the data, as illustrated in table 3.6.

Date	**WakeUpPain**	**Gender**
2023-10-20	65	Male
2023-10-21	58	Male
2023-10-22	72	Male

Table 3.6: Some unprocessed rows from the dataset which include time-dependent (Date and WakeUpPain) and non-time dependent (Gender) features. These rows are made up to preserve privacy.

# Patients	14 with JIA, 17 without JIA, Total 31
# Actigraphy summary per patient	8-12 days
# Actigraphy measurements per patient	Not available
# Diary entries per patient	8-12 days

Table 3.7: Todd dataset characteristics

3.2.2 Todd Data [19]

The Todd dataset (see table 3.7) contains data for 31 children in a single CSV file combining both actigraphy and diary data. 14 of them are JIA patients and 17 of them do not have JIA. We exclude people who do not have JIA as they do not have any pain for any of the days and are not a valid target for this study. The data contains some static demographic information as well as entries that are filled out by the parents or guardians for each night. This demographic data is the same as that of SIPA dataset although the column names might differ. The pain level of the child when the child wakes up is manually recorded in the diary for each day by the parents as an integer on a scale from 1 to 10. Although Todd dataset 3.10 has fewer patients than SIPA dataset 3.9, it has more data points available.

3.2.3 SLE Data [16]

The SLE dataset (see table 3.8) encompasses data from a cohort of patients diagnosed with Systemic Lupus Erythematosus (SLE), commonly known as lupus. The dataset is consolidated into a single CSV file, which combines diary and actigraphy data. It includes data for a total of 20 patients.

The dataset offers a comprehensive view into the patients' health status with dynamic health metrics. This dataset does not have any static demographic values. These dynamic entries are recorded on a daily basis, either by the patients themselves or by their caregivers. This includes daily symptom severity, with a particular focus on morning wake up pain

# Patients	20
# Actigraphy summary per patient	10 days
# Actigraphy measurements per patient	Not available
# Diary entries per patient	10

Table 3.8: SLE dataset characteristics

recorded on a scale from 1 to 10.

3.3 Problem Description: Machine Learning Task Definition

The goal is to design and train ML models to predict whether there will be an increase of more than 10% in a child's pain level (WakeUpPain) upon waking up on day $d+1$, compared to the pain level they experienced upon waking up on day d. Let's name this target column as WakeUpPainDelta. Refer to figure 3.2 and figure 3.3 for an illustration on SIPA and Todd datasets respectively. Please note that the scale of these 2 datasets for WakeUpPain is different.

$$\text{WakeUpPainDelta} = \begin{cases} 1 & \text{if WakeUpPain}[d+1] > 1.10 \times \text{WakeUpPain}[d] \\ 0 & \text{otherwise} \end{cases}$$

Where:

- WakeUpPain[d] represents the WakeUpPain value on day d.
- WakeUpPain[$d+1$] represents the WakeUpPain value on the following day, $d+1$.
- If there's no previous day WakeUpPain value available for day d, then WakeUpPainDelta is set to 0. This is done for day 1 where there is no previous day WakeUpPain available. Another approach was discarding this data which leads to worse accuracy as tested over multiple experiments.

Total # Diary labeled instances	258
Total # Actigraphy Summary labeled instances	195
Total # Actigraphy Day labeled instances	195
Total # patients with JIA	19
# Combined labeled instances per patient	10-11
Total # Combined labeled instances	195

Table 3.9: SIPA dataset labeled. Some data is lost during merging due to participants not wearing the actigraphy device during start/end of study.

Refer to Figure 3.1 for the model design pipeline. The prediction will be based on what parents or guardians write in the diary in response to the question, "Please rate your child's pain at the time your child woke up." To make this prediction, the models can utilize all the data that is recorded and measured up until the evening of day d. This enables parents to use the predictive capabilities of the ML model to ascertain the likelihood of a significant increase in their child's pain level the following morning. Consequently, parents can then decide whether to pre-medicate their child on the night of day d to potentially mitigate this pain. It is crucial to emphasize that for accurate predictions for day $d+1$, our models do not incorporate actigraphy data recorded during the night from day d to day $d+1$. This is because this information is not available in time for parents to take preventative action. The labeled data summary with WakeUpPainDelta as target variable can be seen in Tables 3.9, 3.10 and 3.11.

Creation of the labeled instances involves renaming attributes (see table 3.4), filling 0 for NA values for numeric columns since we would have to drop a significant part of the data if we didn't fill them and filling them with average wasn't an option since the data is supposed to have temporal nature i.e. future data isn't available to take average (there are 13 to 23 missing values for each patient) and doing one-hot-encoding for categorical columns

Total # labeled instances	143
Total # patients with JIA	14
# Labeled instances per patient	8-12

Table 3.10: Todd dataset labeled data

Total # labeled instances	197
Total # patients with SLE	20
# Labeled instances per patient	8-12

Table 3.11: SLE dataset labeled data

such as 'PatientMedicineTaken'. We split the data into train and test instances 4 and then normalize both using MinMaxScaler fitted on the train data.

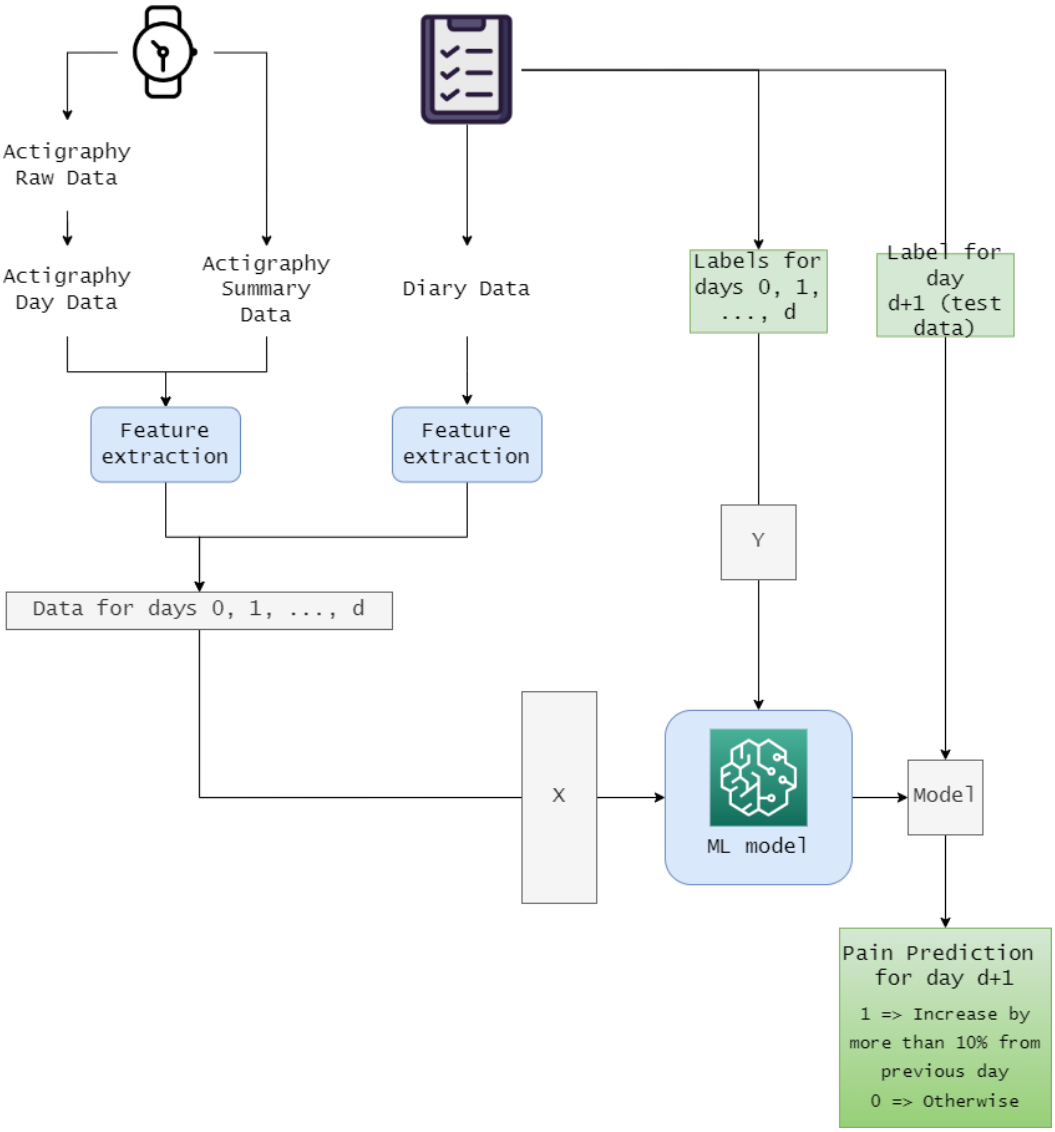
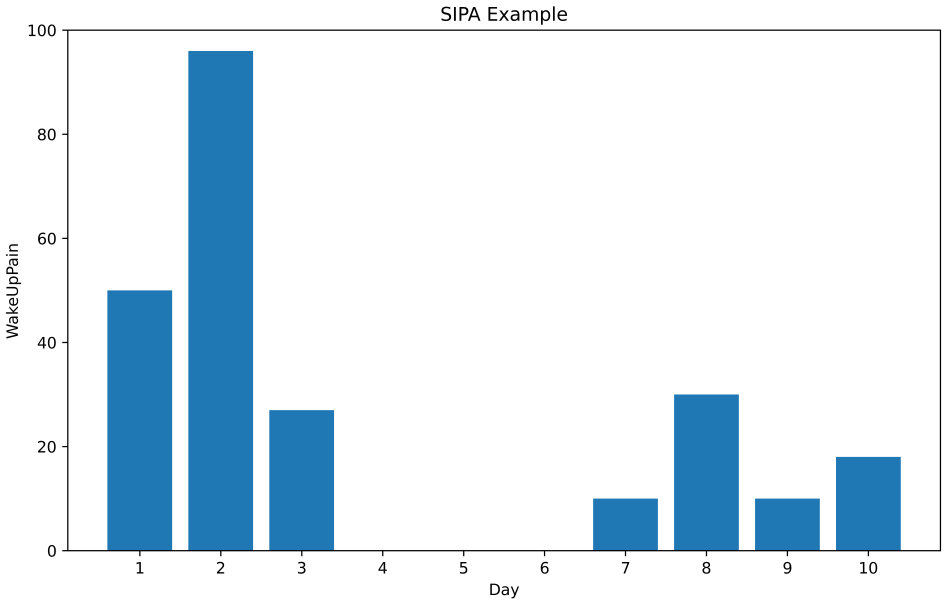
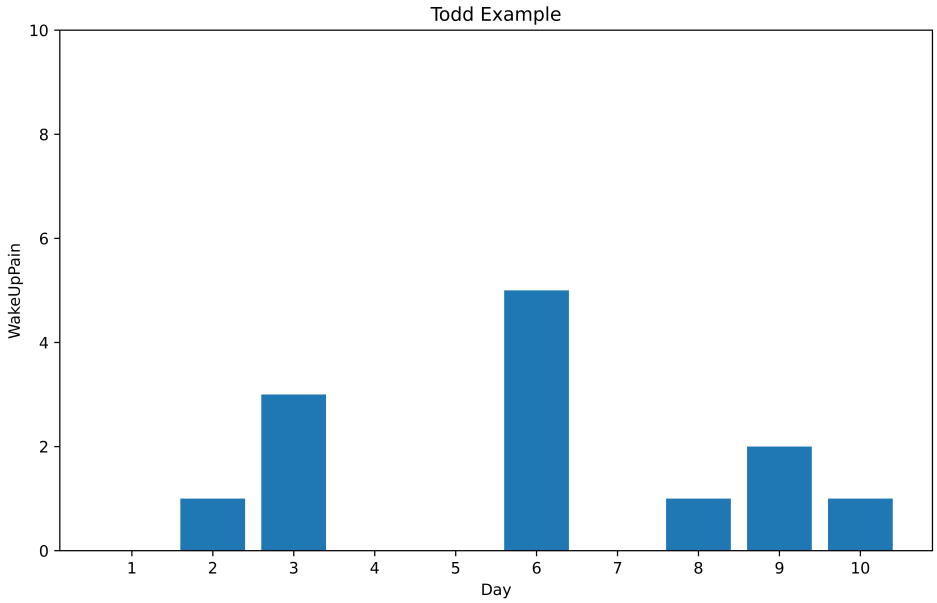


Figure 3.1: Overview diagram



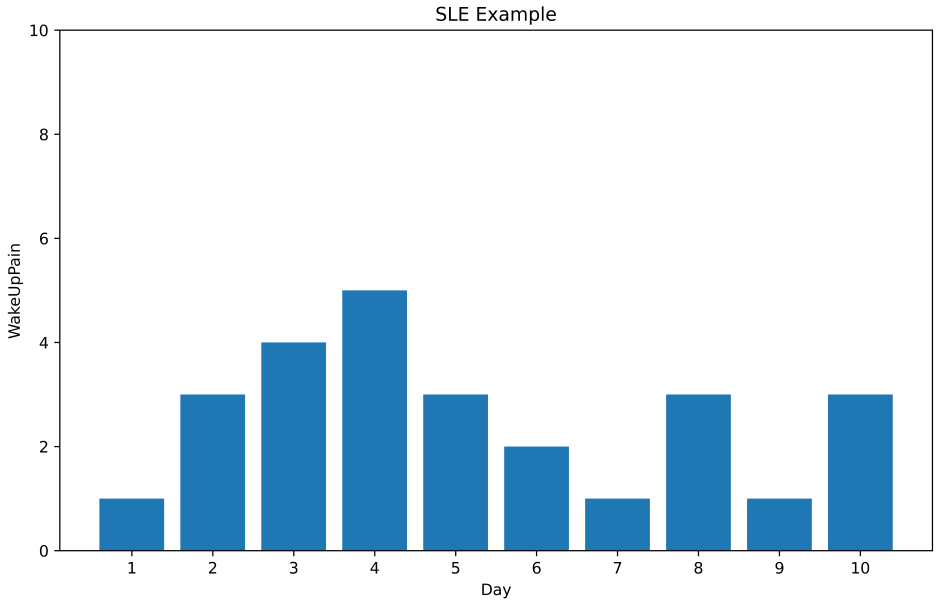
Delta = 0 1 0 0 0 0 0 1 0 0

Figure 3.2: Illustration of pain level for a hypothetical patient from SIPA Dataset over consecutive days and then the Delta which indicates at least 10% increase from the previous day. To maintain patient privacy, the above charts are made up. However the data points resemble actual SIPA data. In this example, the task is to predict the spikes in pain for Day 2 and Day 8. The chart has a Y scale from 0 to 100 while the Delta is 0 or 1.



Delta = 0 0 1 0 0 1 0 0 0 0

Figure 3.3: Illustration of pain level for a hypothetical patient from Todd Dataset over consecutive days and then the Delta which indicates at least 10% increase from the previous day. To maintain patient privacy, the above charts are made up. However the data points resemble the actual Todd data. In this example, the task is to predict the spikes in pain for Day 3 and Day 6. The chart has a Y scale from day 0 to 10 while the Delta is 0 or 1.



Delta = 0 1 0 0 0 0 0 1 0 1

Figure 3.4: Illustration of pain level for a hypothetical patient from SLE Dataset over consecutive days and then the Delta which indicates at least 10% increase from the previous day. To maintain patient privacy, the above charts are made up. However the data points resemble the actual Todd data. In this example, the task is to predict the spikes in pain for Day 2, Day 8 and Day 10. The chart has a Y scale from 0 to 10 while the Delta is 0 or 1.

Chapter 4

METHODOLOGY

4.1 Data Filtering and Preprocessing

4.1.1 SIPA data

Let's start with the columns we have from chapter 3 that will be used for our analysis. These columns are mentioned in tables 4.1, 4.2 and 4.3.

Diary Data Feature	**Description**
WakeUpPain	Pain the patient feels when they wake up. This feature for day d isn't used for prediction or training in day d.
WakeUpPainDelta	Pain the patient feels when they wake up compared to the previous day 3.3. This is the target variable.
PatientMedicineTaken	Notes if the patient took medicine today.
WakeUpMood	Notes the mood level from 0 to 100 when the patient wakes up.

Table 4.1: Features of Diary Data

Actigraphy Summary Feature	**Description**
Efficiency	Sleep efficiency of patient
Exposure Green	Exposure to green light when sleeping
Wake Time	Time spent woken up during the night
	Continued on next page

Table 4.2 – continued from previous page

Actigraphy Summary Feature	**Description**
SleepQuality	Sleep quality of the night
Sleep Time	Total time spent sleeping during the night
Duration	Total duration of the measurement from start to finish.

Table 4.2: Features of Actigraphy Summary Data

Actigraphy Day Feature	**Description**
Daytime_Activity	Daytime activity level generated by averaging raw data from 10 am to 8pm
Daytime_Blue Light	Daytime blue light level recorded generated by averaging raw data from 10 am to 8pm
Daytime_Sleep/Wake	Daytime sleep time generated by averaging raw data from 10 am to 8pm
Daytime_Red Light	Daytime red light level recorded generated by averaging raw data from 10 am to 8pm
Daytime_White Light	Daytime white light level recorded generated by averaging raw data from 10 am to 8pm

Table 4.3: Features of Actigraphy Day Data

1. **Column Construction:** The following table 4.4 summarizes the constructed features derived from the primary dataset and the method of construction:

Column Name	**Description**
PreviousDayWakeUpPain	Represents the pain level upon waking up from the previous day (using 'WakeUpPain'). For day 1, the value is 0.
PreviousDayWakeUpMood	Indicates the mood upon waking up from the previous day (using 'WakeUpMood'). For day 1, the value is 0.
WakeUpPain_MA_2	2-day moving average of 'WakeUpPain'. Calculated over last 2 days.
WakeUpPain_MA_3	3-day moving average of 'WakeUpPain'. Calculated over last 3 days.
WakeUpPain_MA_4	4-day moving average of 'WakeUpPain'. Calculated over last 4 days.
WakeUpPain_MA_5	5-day moving average of 'WakeUpPain'. Calculated over last 5 days.
WakeUpPain_Ever1	1 for day d if the WakeUpPainDelta was 1 for any day from day 1 to day d-1 otherwise 0.
WakeUpPain_Max	for day d, its the maximum of WakeUpPain from day 1 to day d-1. For day 0, it's 0.
WakeUpPain_Min	for day d, its the minimum of WakeUpPain from day 1 to day d-1. For day 0, it's 0.
Medicine_{medicine-group}	1 if the patient takes the medicine from the medicine group given in suffix else 0.
Income_{income-group}	1 if participant is in the given income group given in suffix else 0.
ParentsMaritialStatus_-{marital-status}	1 if participant's parent is in the given marital status group given in suffix else 0.

Continued on next page

Table 4.4 – continued from previous page

Column Name	Description
PatientGender_{gender}	1 if participant is in the given gender group given in suffix else 0.
PatientRace_{race}	1 if participant is in the given racial group given in suffix else 0.
PreviousDayPatient-MedicineTaken_Yes	1 if patient took medicine on day d-1 else 0.
PreviousDayPatient-MedicineTaken_No	0 if patient took medicine on day d-1 else 1.

Table 4.4: Summary of constructed features

2. **Column Selection:** Columns of prime interest, such as 'WakeUpPain', 'PreviousDayWakeUpPain', and 'PreviousDayWakeUpMood', are singled out and mentioned in tables 4.1, 4.2, 4.3 and 4.4. We also computed moving averages for 'WakeUpPain' over a span of 2 to 5 days to provide more contextual data. This column selection is done using domain knowledge, experimentation and also using forward selection and backward elimination techniques. For each merge option, specific columns are selected to ensure that the most relevant data is combined.

   The columns selected for each option are as follows:

   - **Diary data:**
     - WakeUpPain
     - PreviousDayWakeUpPain
     - WakeUpPain MA2
     - WakeUpPain MA3

  - WakeUpPain MA4
  - WakeUpPain MA5
- **Actigraphy Summary data:**
  - Efficiency
  - Exposure Green
  - Wake Time
  - Exposure Red
  - SleepQuality
  - Sleep Time
  - Duration
- **Actigraphy Day data:**
  - Daytime Activity
  - Daytime Blue Light
  - Daytime Sleep/Wake
  - Daytime Red Light
  - Daytime White Light

3. **Finalizing:** Upon the consolidation of all data files, a final layer of preprocessing is initiated. This phase mainly targets missing or duplicate data points. Missing values are typically filled based on their nature to ensure data integrity. The total number of missing values filled are around 13 to 23 per patient including all columns. The missing values filled also depends on the columns selected. The value filled is 0 for all columns.

4. **Splitting:** For our analysis, we employ a "leave-one-out" cross-validation method. In this approach, one patient's data is kept aside as the test set, while the rest serve as the training set. This process is repeated iteratively, each time leaving out a different patient's data for testing. After all iterations are complete, we compute the average of

all accuracy scores obtained from each test set to determine the overall performance of the model.

5. **Input Features and Target Separation:** The culmination of the preprocessing stage is the separation of input features from the target. Our target variable in this study is 'WakeUpPainDelta'. All other variables are treated as potential predictors or features. This separation lays the groundwork for subsequent modeling and analysis.

4.1.2 Todd Data

The Todd dataset is already pre-processed. The pain index in the Todd dataset goes from 0 to 10 so we multiply it by 10 to have the same scale as the SIPA dataset. There are 17 people who do not have JIA. We exclude these 17 participants as they do not have any pain for any of the days and are not a valid target for this study. The features are mentioned in table 4.5.

Column Name	**Description**
Subject ID	Unique identifier for each subject in the study.
child type	JIA Type of the child participant, such as control or specific JIA condition.
nap duration	Duration of each nap taken during the day.
Child am restless total score	Total score of the child's restlessness in the morning.
pain.am.o	Total score of the child's pain in the morning between 0 to 10 (also known as WakeUpPain).
percent time in wake	Percentage of the interval that the subject spent awake.
sleep onset latency for SLEEP interval	Time taken to fall asleep after bedtime during the sleep interval.
	Continued on next page

Table 4.5 – continued from previous page

Column Name	Description
severity of child pain discom during sleep	The severity of pain or discomfort experienced during the night.
severity of pain discom	The severity of pain or discomfort experienced during the day.

Table 4.5: Summary of Todd features relevant to the study

4.1.3 SLE Data

The SLE dataset is already pre-processed. The pain when woken up, i.e. WakeUpPain (renamed from pain_m), in the SLE dataset goes from 0 to 10 so we multiply it by 10 to have the same scale as the SIPA dataset. There are 20 people all of whom have an ailment. The columns are explained in detail in table 4.6.

Column Name	Description
ID	Unique identifier for each participant in the study.
no	Sequential number indicating the order of data entries for each participant.
date	Date of the recorded data in the format mm/dd/yyyy.
day_of_week	Numeric representation of the day of the week (e.g., 3 for Tuesday).
day_of_week_acti	Actual day of the week as a number for actigraphy measurements (e.g., 2 for Monday).
weekday	Binary indicator where 1 represents a weekday and 0 represents a weekend.

Continued on next page

Table 4.6 – continued from previous page

Column Name	Description
bedtime_sd	Scheduled bedtime in 24-hour format.
fall_asleep_sd	Time taken to fall asleep after going to bed, in 24-hour format.
sleep_onset	Actual time of falling asleep, in 24-hour format.
waketime_sd	Scheduled wake-up time, in 24-hour format.
oob_sd	Out of bed time, in 24-hour format.
sleep_offset	Actual wake-up time, in 24-hour format.
SP	Total sleep period in seconds from sleep onset to sleep offset.
TST	Total sleep time in seconds, excluding awakenings.
SOL	Sleep onset latency, time in seconds to transition from full wakefulness to sleep.
wakenings	Number of awakenings during the sleep period.
WASO	Wake after sleep onset, total time in seconds spent awake after initially falling asleep.
SQ_sd	Sleep quality score, on a scale from 1 to 10.
mood_sd	Mood score upon waking, on a scale from 1 to 10.
pain_m	Pain when woken up in the morning. Renamed to WakeUpPain for clarity.
pain_avg	Average pain level on waking, on a scale from 1 to 10.
pain_n	Pain when sleeping in the previous night.
WASO_sd	Standard deviation of WASO across different days for the same individual.

Continued on next page

Table 4.6 – continued from previous page

Column Name	Description
SE_ac	Sleep efficiency as a percentage, calculated as (TST / SP) * 100.

Table 4.6: Summary of SLE features

4.2 Classifier and Parameters Overview

Our analysis is based on a machine learning approach. We utilize various classifiers for predictions. Each classifier's primary role is to discern patterns within the data, and their parameters are crucial in determining the model's accuracy and effectiveness.

1. **RandomForestClassifier:** This machine learning model employs decision trees to make its predictions. This is the default model used wherever not specified otherwise for this study. The following parameters are set for the classifier:

   - `n_estimators`: Set to 25. This denotes the number of trees in the forest.
   - `random_state`: Set to 42, which ensures the same sequence of random numbers, allowing for reproducible results.
   - `criterion`: Set to 'entropy', which measures the quality of the splits.

2. **LogisticRegression:** It is a statistical method for predicting binary classes. The main parameters for this classifier are:

   - `C`: Inverse regularization strength; smaller values indicate stronger regularization. Set to default which is 1.0.
   - `solver`: Algorithm used for optimization (e.g., 'liblinear', 'saga', etc.). Set to 'lbfgs'.

- `max_iter`: Maximum number of iterations for the solver to converge. Set to 1000.

3. **Decision Tree:** It is a flowchart-like tree structure where an internal node represents a feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. Main parameters are:

   - `criterion`: The function to measure the quality of a split (e.g., 'gini' or 'entropy'). Set to default which is 'gini'.
   - `max_depth`: The maximum depth of the tree. Set to default which is 'None' which implies no max depth.

4. **SVC (Support Vector Classifier):** A classifier that aims to find the hyperplane that best separates classes by maximizing the margin. Key parameters include:

   - `kernel`: Specifies the kernel type to be used in the algorithm. For this analysis, it is set to 'linear'.
   - `probability`: Whether to enable probability estimates. Here, it is set to 'True'.

5. **XGBClassifier (Extreme Gradient Boosting):** This method is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. Essential parameters are:

   - `n_estimators`: Number of boosting rounds. Set to 100 for this analysis.
   - `learning_rate`: Boosting learning rate (xgb's "eta"). It is set to 0.1.

Training and Evaluation: After preparing the data, each classifier is trained using the `Data` DataFrame and subsequently evaluated using the `Test_Data` DataFrame that we got after preprocessing.

Results: Post evaluation, the average false positive rate (FPR) and average accuracy are computed and presented for each classifier. For the evaluation of classifiers, a method known as leave-one-out cross-validation (LOOCV) is employed. In this method, during each iteration, a different patient is set aside for evaluation while the classifier is trained on the data of the remaining patients. The final performance metrics are then derived by averaging the results across all iterations.

1. **For SIPA Dataset:**
    - The classifiers are exclusively trained on data from the SIPA dataset.
    - Evaluation is carried out using the SIPA dataset itself, employing the LOOCV approach.
2. **For Todd Dataset:**
    - The classifiers are trained only using the data from the Todd dataset.
    - Evaluation, again, uses the LOOCV approach but on the Todd dataset.
3. **For Combined Analysis on JIA data (SIPA and Todd):**
    - Training uses a combined set of data from both the SIPA and Todd datasets.
    - The LOOCV approach is applied for evaluation, where patients (from either dataset) are left out in turns. This means that the evaluation loop includes patients from both SIPA and Todd.
4. **For SLE Dataset:**
    - The classifiers are exclusively trained on data from the SLE dataset.
    - Evaluation is carried out using the SLE dataset itself, employing the LOOCV approach.

4.3 Synthetic Data Generation

Synthetic data generation is a pivotal aspect of modern data science, enabling researchers and practitioners to augment datasets, improve model training, and ensure privacy. In our case, we use synthetic data to enhance privacy. This section delves into three innovative approaches: MST, GAN, and LLM, each offering unique advantages in the generation of synthetic data.

Differential Privacy [3] is a statistical technique that ensures the privacy of individual entries in a dataset by introducing a degree of randomness to the data. The ϵ value, central to this concept, quantifies the tradeoff between privacy and accuracy in the data released. A smaller ϵ value indicates stronger privacy guarantees but potentially less accurate or useful data, whereas a larger ϵ value allows for more accurate data at the cost of weaker privacy protections. This framework enables the safe use of data for analysis without compromising the confidentiality of individual data points. In the context of the results presented in Chapter 5, understanding the role and implications of varying ϵ values is paramount for interpreting the performance and privacy guarantees of the models developed.

4.3.1 MST - Maximum Spanning Tree [12]

Maximum Spanning Tree (MST) is a technique often used in network analysis and graph theory. In the context of synthetic data generation, MST is a synthetic data generation algorithm that follows the "select-measure-generate" paradigm [12] and can be applied to create data structures that represent the minimal possible connections within a dataset, preserving essential relationships without the complexities of real-world data. This approach is particularly useful in scenarios where the underlying structure of data is more important than the data itself, such as in certain types of simulation models. MST requires the data to be categorical so we convert the numerical data into categorical using binning into 3 buckets. cliques used in Table 5.9 indicate column pairs that we know are related using prior domain knowledge.

4.3.2 MWEM PGM - Multiplicative Weights and Exponential Mechanisms in Probabilistic Graphical Models [13]

Multiplicative Weights and Exponential Mechanisms in Probabilistic Graphical Models (MWEM PGM) is a sophisticated technique employed in the field of data privacy and synthetic data generation. This approach is particularly adept at preserving the statistical properties of a dataset while ensuring individual data points' privacy. Unlike methods that focus solely on the structural representation of data, MWEM PGM emphasizes statistical accuracy and privacy preservation through an iterative process. The process involves updating probability distributions using multiplicative weights and selecting queries to improve using the exponential mechanism. The common hyperparameters used in all experiments are given in table 4.7. MWEM PGM from smartnoise guesses a reasonable number of iterations to run [1].

Hyperparameters	**Value**
split_factor	Size of the groupings of features for the histograms. Set to 10.
preprocessor_eps	The maximum budget used for preprocessing [2] data. Set to $\epsilon/3$.

Table 4.7: MWEM PGM Hyperparameters

The steps of the MWEM PGM algorithm can be broken down as follows:

1. **Initialize**: Begin with an initial model or distribution that represents the original data. This model is typically simplistic and does not yet accurately reflect the true data distribution.

[1] `https://docs.smartnoise.org/synth/synthesizers/mwem.html`

[2] One usecase would be to infer bounds for continuous columns

2. **Iterate**: Perform a series of iterations that consist of three key phases:

   (a) **Query Selection**: Use the exponential mechanism to select queries that are most informative about the discrepancies between the current model and the real data. This selection is aimed at identifying areas where the model's accuracy can be most improved.

   (b) **Error Estimation**: For the selected queries, estimate the error between the query's results on the real dataset and on the synthetic data generated from the current model. This step highlights the specific inaccuracies in the model.

   (c) **Model Update**: Update the model by adjusting the probability distributions using multiplicative weights. The update is guided by the error estimates from the previous step, aiming to reduce these discrepancies and improve the model's accuracy.

3. **Generate**: After a predefined number of iterations or once the model reaches a satisfactory level of accuracy, use the final model to generate synthetic data. This data is expected to closely mimic the statistical properties of the original dataset while maintaining the privacy of individual data points.

The MWEM PGM approach is distinguished by its iterative refinement of the model, which allows for progressively improving the synthetic data's resemblance to the real data in terms of statistical properties. This method is particularly useful in scenarios where the preservation of statistical characteristics is crucial, and privacy concerns are paramount.

4.3.3 GAN - CTGAN [17]

Generative Adversarial Networks (GANs), and specifically CTGAN, have revolutionized the field of synthetic data generation. CTGAN, or Conditional Tabular Generative Adversarial Network, is designed to generate synthetic tabular data. It addresses common challenges

in traditional GANs, such as imbalanced data, by using a conditional generator and training-by-sampling approach. CTGAN is particularly effective in scenarios where preserving the statistical properties of the original dataset is crucial, such as in data anonymization and augmentation for machine learning models. The values for the hyperparamters are given in table 4.8.

Hyperparameters	**Value**
embedding_dim	Size of the random sample passed to the Generator. Set to 128.
generator_dim	Size of the output samples for each one of the Residuals. A Residual Layer will be created for each one of the values provided. Set to (256, 256).
discriminator_dim	Size of the output samples for each one of the Discriminator Layers. A Linear Layer will be created for each one of the values provided. Set to (256, 256).
generator_lr	Learning rate for the generator. Set to 2e-4.
generator_decay	Generator weight decay for the Adam Optimizer. Set to 1e-6.
discriminator_lr	Learning rate for the discriminator. Set to 2e-4.
discriminator_decay	Discriminator weight decay for the Adam Optimizer. Set to 1e-6.

Table 4.8: CTGAN Hyperparameters

4.3.4 LLM - distilgpt [2]

DistilGPT, a distilled version of the Generative Pre-trained Transformer, represents a breakthrough in language modeling for synthetic data generation. As a lightweight model, it retains much of the power of its predecessor but with reduced complexity and resource

requirements. DistilGPT is adept at generating coherent and contextually relevant text, making it an invaluable tool for tasks such as chatbot training, content creation, and any application where natural language generation is key. Its efficiency and effectiveness make it a popular choice for generating high-quality synthetic textual data however we adapt it here to generate tabular data using training with special prompts. For each dataset row, we finetune a DistilGPT model using the following format: 'T is t, A is a, B is b, C is c, etc.', where 'T' represents the target variable with a value of 't', and 'A', 'B', 'C', etc., represent other column variables with respective values of 'a', 'b', 'c', etc. Some examples are given below. Note that only some columns are mentioned to keep it short and understandable.

Example Training 1. *WakeUpPainDelta is 1, Sleep_Time is 632, Age is 5, Gender is Male*

Example Training 2. *WakeUpPainDelta is 0, Sleep_Time is 654, Age is 4, Gender is Male*

During generation, we initiate the model with start tokens 'T is t' and let the language model (LLM) complete the sentence. The output is then parsed back into row format after we let the model complete the sentence.

Example Generation 1. *WakeUpPainDelta is 0*

Example Generation 2. *WakeUpPainDelta is 1*

In our efforts to generate synthetic tabular data while ensuring differential privacy, we experimented with Opacus [18]. Despite the attempt, the approach did not yield successful results; even the column names became obfuscated, making it impossible to correlate the generated values with the corresponding rows.

Chapter 5

RESULTS

This study makes use of three datasets: the SIPA Dataset, Todd Dataset, and SLE Dataset, as described in chapter 3. We discuss the results over each dataset in the following sections. We go over synthetic data generation over the SLE dataset. We also combine the SIPA and Todd datasets and create a model for this combined dataset.

5.1 SIPA Dataset

The baseline model for the SIPA dataset, i.e. a classifier that always predicts 0, has an accuracy of 88% and an FPR of 0%. The column names are grouped together in table 5.1 for easier reading and understanding. "WakeUpPainDelta" is the target variable and is not used as a input for the model. When computing a moving average over a specified period, like 4 days, but with limited data, such as only 2 days available, we adjust by using the data we have. Instead of a 4-day average, we calculate a 2-day average. This is same for all the moving averages. Refer to table 4.4 for more details on these columns.

The results of models trained using Random Forest Classifier are shown in table 5.2. We notice that using "WUP Columns" gives the highest accuracy showing that the last ailment history of the patients has a lot of significance. The accuracy drops if we use more columns as it leads to overfitting. We try the experiment with various windows of WakeUpPain Moving Averages to check how much past data is relevant for pain prediction and we observe that going back 5 days gives the best results.

Table 5.3 shows what happens if we set "WakeUpPainDelta" to 1 if the patient takes NSAID type of medicine on the previous night. The reasoning behind changing the "WakeUpPainDelta" to 1 is that if they didn't take the medicine, they would have seen a pain

increase the next day. This experiment is important as we want to see if we can predict the pain if some patients are already taking pain relief medicine. We see that the baseline changes to 70% due to the change in values for the target variable. The actual best accuracy is 94% since we modify the pain predictor to accurately reflect the status if patient took a pain reliving medicine.

Table 5.4 shows the results for different classifiers using the "WUP Columns". It shows that classifier models based on decision trees like Random Forest Classifier work best.

Figure 5.1 shows limited depth of a decision tree trained on "WUP Columns" to visualize some important columns. We can see that depending on PreviousDayWakeUpPain, different values of 3 and 4 days WakeUpPain moving averages are relevant. We observe that the initial layers of the tree predominantly consider the wake up pain from the most recent day, whereas, in the deeper sections, historical data gains more significance.

Group Name	**Columns**
WUP Columns	WakeUpPainDelta, PreviousDayWakeUpPain, WakeUpPain_MA2, WakeUpPain_MA3, WakeUpPain_MA4, WakeUpPain_MA5
All WUP Columns	WUP Columns, WakeUpPain_Ever1, WakeUpPain_Max, WakeUpPain_Min
Income Columns	Income_$140,000 or more, Income_$100,000 to $119,999, Income_$120,000 to $139,999, Income_$20,000 to $39,999, Income_$40,000 to $59,999, Income_$60,000 to $79,999, Income_$80,000 to $99,999, Income_Unknown

Continued on next page

Table 5.1 – continued from previous page

Group Name	**Columns**
Med Columns	Medicine_Antihistamines, Medicine_Biologic agents, Medicine_DMARD, Medicine_Dietary Supplements, Medicine_Interleukin-1 Inhibitors, Medicine_NSAID, Medicine_Proton-pump inhibitors, Medicine_fluorides, Medicine_folic avid, Medicine_unknown, PreviousDayPatientMedicineTaken_0, PreviousDayPatientMedicineTaken_No, PreviousDayPatientMedicineTaken_Yes,
Demographic Columns	ParentGender_Female, ParentGender_Unknown, ParentsMaritialStatus_Divorced, ParentsMaritialStatus_Domestic partnership, ParentsMaritialStatus_Married, ParentsMaritialStatus_Unknown, PatientAge, PatientAge_Unknown, PatientCaffeine_No, PatientCaffeine_Yes, PatientGender_Female, PatientGender_Male, PatientGender_Unknown, PatientHealth_No, PatientHealth_Yes, PatientRace_American Indian/Native American, PatientRace_Hispanic/Latino, PatientRace_Native Hawaiian or Other Pacific Islander, PatientRace_Other (describe below), PatientRace_Unknown, PatientRace_White/not Hispanic
Patient Health Columns	PreviousDaySleepQuality, PreviousDayWakeUpMood
Actigraphy Daytime Columns	Daytime_Activity, Daytime_Blue Light, Daytime_Sleep/Wake, Daytime_Red Light, Daytime_White Light

Continued on next page

Table 5.1 – continued from previous page

Group Name	**Columns**
Actigraphy Night Summary Columns	Efficiency, Exposure Red, Duration
All Diary Columns	All WUP Columns, Income Columns, Med Columns, Demographic Columns, Patient Health Columns

Table 5.1: Abbreviation Key for SIPA Dataset Columns

Merge Method	**Attributes**	**Avg. Accuracy**	**False Positive Rate**
Diary	WUP Columns	98%	0%
Diary	WUP Columns, WakeUpPain_MA6	95%	1.24%
Diary	WUP Columns, WakeUpPain_MA6, WakeUpPain_MA7	95%	0.82%
Diary	WakeUpPainDelta, PreviousDayWakeUpPain, WakeUpPain_MA2, WakeUpPain_MA3, WakeUpPain_MA4	96%	0.36%
Diary	WakeUpPainDelta, PreviousDayWakeUpPain, WakeUpPain_MA2, WakeUpPain_MA3	96%	0.36%
Diary	WakeUpPainDelta, PreviousDayWakeUpPain, WakeUpPain_MA2	94%	0.91%
Diary	WakeUpPainDelta, PreviousDayWakeUpPain	88%	3.65%

Continued on next page

Table 5.2 – continued from previous page

Merge Method	**Attributes**	**Avg. Accuracy**	**False Positive Rate**
Diary WakeUpPainDelta + Summary	WakeUpPainDelta, Efficiency, Exposure Green, Wake Time	87%	0.57%
Diary WakeUpPainDelta + Day	WakeUpPainDelta, Daytime_Activity, Daytime_Blue Light, Daytime_Sleep/Wake, Daytime_Red Light	88%	1.3%
Diary + Summary	WUP Columns, Actigraphy Night Summary Columns	88%	0.57%
Diary + Summary	WakeUpPainDelta, PreviousDayWakeUpPain, WakeUpPain_MA2, Actigraphy Night Summary Columns	86%	2.86%
Diary + Summary	WakeUpPainDelta, PreviousDayWakeUpPain, WakeUpPain_MA2, Efficiency	84%	4.65%
Diary + Summary	WakeUpPainDelta, PreviousDayWakeUpPain, WakeUpPain_MA2, SleepQuality	85%	5.04%
Diary + Summary	WakeUpPainDelta, PreviousDayWakeUpPain, WakeUpPain_MA2, Sleep Time	82%	7.51%
Diary + Summary	WakeUpPainDelta, PreviousDayWakeUpPain, WakeUpPain_MA2, Exposure Green	85%	5.62%
Diary + Summary	WakeUpPainDelta, Efficiency	82%	7.46%
Diary + Summary	WakeUpPainDelta, SleepQuality	83%	6.07%
Diary + Summary	WakeUpPainDelta, Sleep Time	76%	13.8%
Diary + Summary	WakeUpPainDelta, Exposure Green	80%	11.24%

Continued on next page

Table 5.2 – continued from previous page

Merge Method	**Attributes**	**Avg. Accuracy**	**False Positive Rate**
Diary + Summary	WakeUpPainDelta, PreviousDayWakeUpPain, WakeUpPain_MA2, Sleep Time, SleepQuality, Efficiency	84%	5.72%
Diary + Day	WUP Columns, Daytime_Activity, Daytime_Blue Light	89%	1.58%
Diary WakeUpPainDelta + Summary + Day	WakeUpPainDelta, Actigraphy Daytime Columns, Actigraphy Night Summary Columns	88%	0.65%
Diary + Night + Day	WUP Columns, Actigraphy Daytime Columns, Duration, Efficiency	88%	0.51%

Table 5.2: Results for SIPA Dataset using Random Forest Classifier.

Merge Method	**Attributes**	**Avg. Accuracy**	**False Positive Rate**
Diary	WakeUpPainDelta, All Diary Columns	94%	0%
Diary	WakeUpPainDelta, WUP Columns	88%	1.99%
Diary	WakeUpPainDelta, PreviousDayPatientMedicineTaken_Yes, WUP Columns	93%	1.51%

Table 5.3: Results for SIPA Dataset using Random Forest Classifier after modifying the target variable WakeUpPainDelta to 1 if patient takes NSAID Medicine in previous night otherwise don't modify it. Baseline changes to 70% due to change in target variable.

Classifier	**Avg. Accuracy**	**False Positive Rate**
Random Forest	98%	0%
SVC	88%	0%
XGB	94%	1.72%
Logistic Regression	88%	0%
Decision Tree	95%	2.09%

Table 5.4: SIPA Classifier Performance Metrics comparison for best feature set (1st row in SIPA dataset as shown in table 5.2). Random Forest Classifier works best for this type of data.

5.2 Todd Dataset

The baseline model for the Todd dataset, which always predicts 0, has an accuracy of 81% and an FPR of 0%. The column names are grouped together in table 5.5 for easier reading and understanding.[1] Table 5.6 shows the results on the Todd dataset using Random Forest Classifier. We see that the "WUP Columns" do not work as well as for the SIPA dataset however we see an accuracy of 94% with "WUP Columns", "Other Columns" and "after you went ot bed, you woke up how mnay times".

Figure 5.2 shows limited depth of a decision tree trained on "Other Columns" and "after you went ot bed, you woke up how mnay times" to visualize some important columns. To

[1]Any misspellings in column names are from the dataset.

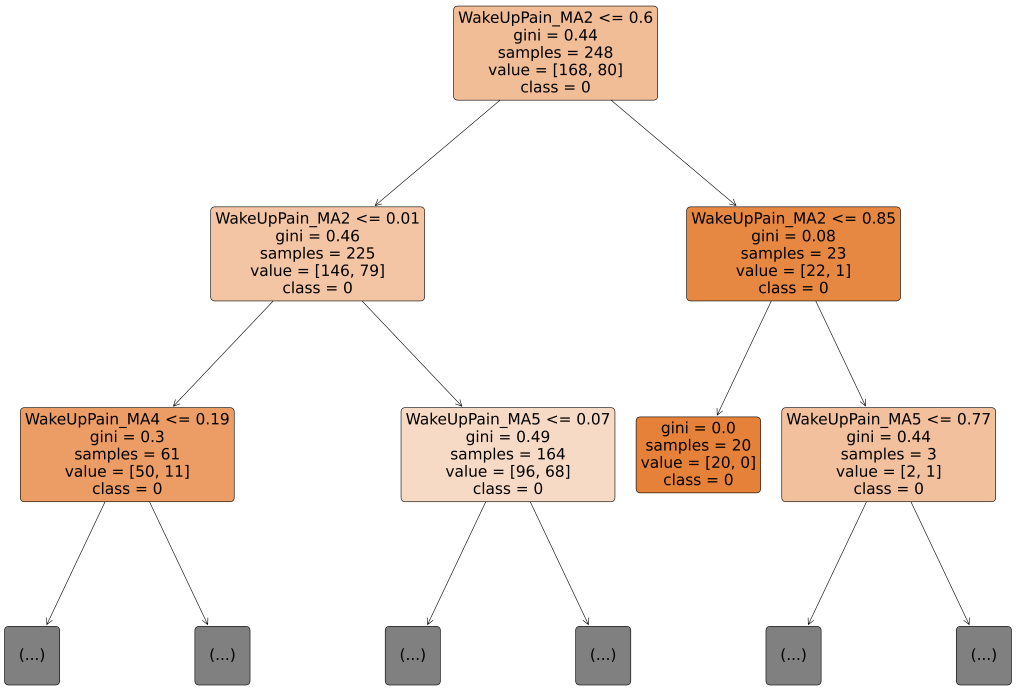


Figure 5.1: Limited Depth Decision Tree Visualization for "WUP columns" of the SIPA experiments showing important columns and thresholds.

predict the WakeUpPain on day d, The root indicates the discomfort (not_at_all value) on day $d-2$ to day $d-1$ night. If no discomfort is felt, the value would be 1 and the decision tree predicts no WakeUpPain for day d. If discomfort is felt, this value would be 0 and then the decision tree checks previous day (day $d-1$) WakeUpPain threshold. If it is low and the time that the patient is awake at day $d-2$ to $d-1$ night is low (the patient had a good night of sleep), it predicts the day d WakeUpPain would be 0 as well otherwise it predicts that day d WakeUpPain would be 1.

Group Name	Columns
WUP Columns	WakeUpPainDelta, PreviousDayWakeUpPain, WakeUpPain_MA2, WakeUpPain_MA3, WakeUpPain_MA4, WakeUpPain_MA5
Other Columns	severity of child pain discom during sleep_not al all, nap duration, severity of pain discom_slightly, severity of child pain discom during sleep_moderately, Child am restless total score, sleep onset latency for SLEEP interval, child type_extended Oligoartiuclar JIA parent, PreviousDayWakeUpPain, percent time in wake

Table 5.5: Abbreviation Key for Todd Dataset Columns

Merge Method	Attributes	Avg. Accuracy	False Positive Rate
Already Merged	WUP Columns	75%	5.14%
Already Merged	WakeUpPainDelta, Other Columns, 'after you went ot bed, you woke up how mnay times'	94%	2.61%
Already Merged	WakeUpPainDelta, Other Columns	93%	2.81%

Table 5.6: Results for Todd Dataset using Random Forest Classifier. For Todd dataset, which is already a combined mix of diary and actigraphy, 'Other Columns' works best.

5.3 SLE Dataset

The baseline model for the SLE dataset, which always predicts 0, has an accuracy of 76% and an FPR of 0%. The column names are grouped together in table 5.7 for easier reading and understanding. The results are shown in table 5.8 using Random Forest Classifier. We

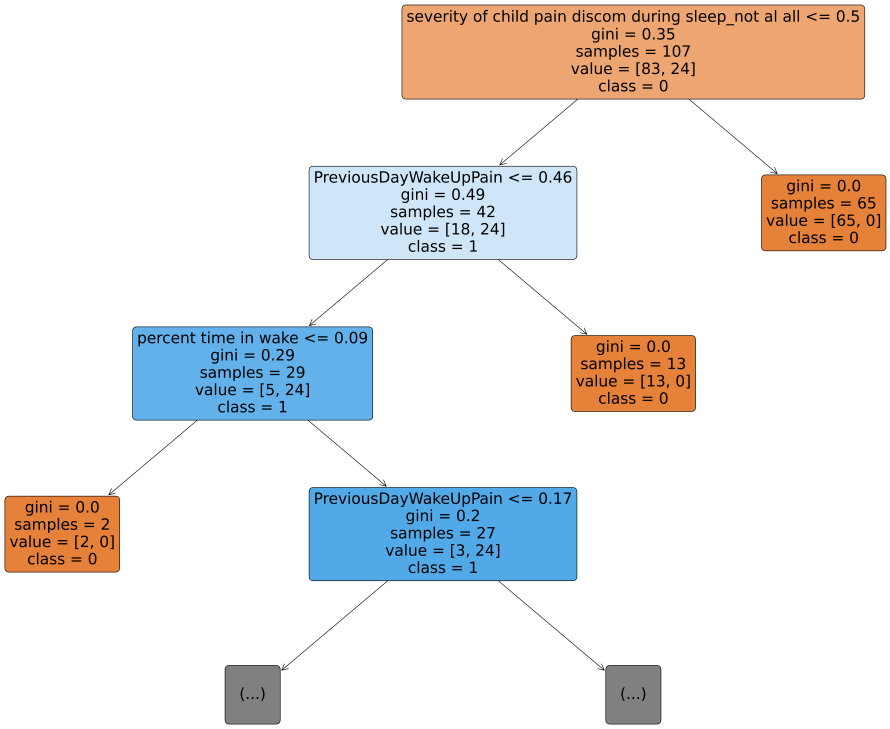


Figure 5.2: Limited Depth Decision Tree Visualization for the Todd experiment trained on "Other Columns" and "after you went ot bed, you woke up how mnay times" showing important columns and thresholds.

can see that the best accuracy of 85% is achieved using a mix of 3 day WakeUpPain moving average and some sleep parameters. Figure 5.3 shows limited depth of a decision tree trained on "fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg" columns for visualization. The important columns are average pain until the previous morning (day $d-1$) and 3 day moving averages for WakeUpPain (day $d-3$ to day $d-1$) along with some important sleep parameters such as time taken to fall asleep and sleep time for the night from day $d-2$ to day $d-1$.

Group Name	**Columns**
WUP Columns	WakeUpPainDelta, PreviousDayWakeUpPain, WakeUpPain_MA2, WakeUpPain_MA3, WakeUpPain_MA4, WakeUpPain_MA5
Sleep Metrics	WASO, fall_asleep_sd_minute, bedtime_sd_minute, SE_ac, TST, sleep_onset_minute, SP, mood_sd, pain_avg
Time Metrics	weekday, day_of_week
All	SP, TST, SOL, wakenings, WASO, SQ_sd, mood_sd, pain_avg, pain_n, WASO_sd, SE_ac, bedtime_sd_hour, bedtime_sd_minute, fall_asleep_sd_hour, fall_asleep_sd_minute, sleep_onset_hour, sleep_onset_minute, waketime_sd_hour, waketime_sd_minute, oob_sd_hour, oob_sd_minute, sleep_offset_hour, sleep_offset_minute

Table 5.7: Abbreviation Key for SLE Dataset Columns

Merge Method	**Attributes**	**Avg. Accuracy**	**False Positive Rate**
Already Merged	WUP Columns	77%	12.12%
Already Merged	WakeUpPainDelta, WakeUpPain_MA3, WakeUpPain_MA4, Sleep Metrics, Time Metrics	80%	1.69%

Continued on next page

Table 5.8 – continued from previous page

Merge Method	**Attributes**	**Avg. Accuracy**	**False Positive Rate**
Already Merged	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	85%	2.68%

Table 5.8: Results for SLE Dataset using Random Forest Classifier. For SLE dataset, which is already a combined mix of diary and actigraphy, sleep columns works best.

5.4 Synthetic Data Generation Results for SLE

In table 5.9, we go over multiple synthetic generation algorithms with various parameters. "Columns Generated" are always a superset of "Columns Used" and indicate the columns that were generated during the synthetic data generation process. "Columns Used" indicated the columns that were used to train the Random Forest Classifier to evaluate utility. While all of the experiments generate 1000 rows (instances) of data, we did experiments with generating less and more rows. With less rows we lose utility and with more rows there is no gain in utility, hence we decided to work with 1000 rows. I ran each experiment once and hence the noise that is sampled during SDG could change from one run to the next, which may lead to better or worse results if the experiments were run again. We observe that MWEM PGM 5.9 and MST 5.9 perform similarly when we compare them training on "All" columns and testing on best columns from table 5.8.

MST results on the best columns from table 5.8 can be seen starting at 5.9 over various ϵ levels. In "Over Sample-MST", starting at row 5.9 over various ϵ levels, I used RandomOverSampler [2] to ensure balance the number of positive and negative WakeUpPainDelta

[2]https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.RandomOverSampler.html

and then applied MST on this data. However the test data was still unbalanced so it resulted in a lower accuracy.

We also go over some algorithms which do not provide formal privacy guarantees like CTGAN 5.9 and LLM 5.9 to see the maximum utility possible. These methods give the best accuracy of 81% which is equivalent to $\epsilon = 10$ from MWEM PGM. The training discriminator and generator loss for CTGAN are plotted in figure 5.4 for one experiment as an example. The graph shows two lines tracking the performance of a training process over time, labeled as "Generator Loss" and "Discriminator Loss." The Generator Loss decreases over time, indicating that the generator's performance at its task is improving. The Discriminator Loss initially decreases but then trends upward, which may suggest that it's getting more difficult for the discriminator to distinguish between real and generated data as the generator improves. Both lines fluctuate, which is typical for any GAN training process.

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
MWEM PGM	$\epsilon = 100$, Number of columns generated = 1000	All	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUp-Pain_MA3, pain_avg	81%	0%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
MWEM PGM	$\epsilon = 10$, Number of columns generated = 1000	All	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	81%	0%
MWEM PGM	$\epsilon = 5$, Number of columns generated = 1000	All	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	80%	1.19%
MWEM PGM	$\epsilon = 1.0$, Number of columns generated = 1000	All	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	78%	3.57%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
MWEM PGM	$\epsilon = 0.1$, Number of columns generated = 1000	All	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	76%	0%
MST	$\epsilon = 100$, Number of columns generated = 1000, Iterations = 250, Cliques = [('WakeUpPainDelta', 'pain_avg')]	All	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	81%	0%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
MST	ϵ = 10, Number of columns generated = 1000, Iterations = 250, Cliques = [('WakeUpPainDelta', 'pain_avg')]	All	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	80%	1%
MST	ϵ = 5, Number of columns generated = 1000, Iterations = 250, Cliques = [('WakeUpPainDelta', 'pain_avg')]	All	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	79%	2.2%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
MST	ϵ = 1.0, Number of columns generated = 1000, Iterations = 250, Cliques = [('WakeUpPainDelta', 'pain_avg')]	All	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	78%	5%
MST	ϵ = 0.1, Number of columns generated = 1000, Iterations = 250, Cliques = [('WakeUpPainDelta', 'pain_avg')]	All	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	63%	34%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
MST	ϵ = 1.0, Number of columns generated = 1000, Iterations = 250, Cliques = [('WakeUpPainDelta', 'pain_avg')]	All	All	71%	21.84%
MST	ϵ = 1.0, Number of columns generated = 1000, Iterations = 250, Cliques = [('WakeUpPainDelta', 'pain_avg')]	All	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, pain_avg	78%	4.37%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
MST	ϵ = 0.1, Number of columns generated = 1000, Iterations = 250, Cliques = [('WakeUpPainDelta', 'pain_avg'), ('WakeUpPainDelta', 'WakeUpPain_MA3')]	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	63%	33.63%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
MST	ϵ = 0.5, Number of columns generated = 1000, Iterations = 250, Cliques = [('WakeUpPainDelta', 'pain_avg'), ('WakeUpPainDelta', 'WakeUpPain_MA3')]	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	73%	15.67%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
MST	ϵ = 1.0, Number of columns generated = 1000, Iterations = 250, Cliques = [('WakeUpPainDelta', 'pain_avg'), ('WakeUpPainDelta', 'WakeUpPain_MA3')]	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	78%	5.06%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
MST	ϵ = 10.0, Number of columns generated = 1000, Iterations = 250, Cliques = [('WakeUpPainDelta', 'pain_avg'), ('WakeUpPainDelta', 'WakeUpPain_MA3')]	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	80%	1.91%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
MST	ϵ = 100.0, Number of columns generated = 1000, Iterations = 250, Cliques = [('WakeUpPainDelta', 'pain_avg'), ('WakeUpPainDelta', 'WakeUpPain_MA3')]	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	81%	0%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
Over Sample - MST	ϵ = 0.1, Number of columns generated = 1000, Iterations = 500, Cliques = [('WakeUpPainDelta', 'pain_avg'), ('WakeUpPainDelta', 'WakeUpPain_MA3')]	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	37%	70.99%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
Over Sample - MST	ϵ = 0.5, Number of columns generated = 1000, Iterations = 500, Cliques = [('WakeUpPainDelta', 'pain_avg'), ('WakeUpPainDelta', 'WakeUpPain_MA3')]	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	47%	58.69%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
Over Sample - MST	ϵ = 1.0, Number of columns generated = 1000, Iterations = 500, Cliques = [('WakeUpPainDelta', 'pain_avg'), ('WakeUpPainDelta', 'WakeUpPain_MA3')]	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	60%	34.82%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
Over Sample - MST	ϵ = 10.0, Number of columns generated = 1000, Iterations = 500, Cliques = [('WakeUpPainDelta', 'pain_avg'), ('WakeUpPainDelta', 'WakeUpPain_MA3')]	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	67%	28.66%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
Over Sample - MST	ϵ = 100.0, Number of columns generated = 1000, Iterations = 500, Cliques = [('WakeUpPainDelta', 'pain_avg'), ('WakeUpPainDelta', 'WakeUpPain_MA3')]	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg	73%	17.96%
CTGAN	Number of columns generated = 1000, Epochs = 100, discrete_columns = ['WakeUpPainDelta']	All	All	75%	12.57%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
CTGAN	Number of columns generated = 1000, Epochs = 100, discrete_columns = ['WakeUpPainDelta']	All	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, pain_avg	81% (Train 62%)	0%
LLM	Model = distilgpt2, Number of columns generated = 1000, Epochs = 25	All	All	81%	0%
LLM	Model = distilgpt2, Number of columns generated = 1000, Epochs = 25	All	WakeUpPainDelta, fall_asleep_sd_minute, bedtime_sd_minute, pain_avg	79%	1.91%
LLM	Model = distilgpt2, Number of columns generated = 1000, Epochs = 50	All	All	79% (Train 83%)	5.05%

Continued on next page

Table 5.9 – continued from previous page

Method	**Parameters**	**Columns Generated**	**Columns Used**	**Avg. Accuracy**	**False Positive Rate**
LLM	Model = distil-gpt2, Number of columns generated = 1000, Epochs = 50	All	WakeUpPainDelta, fall_asleep_sd_minute, bed-time_sd_minute, pain_avg	81% (Train 82%)	0%

Table 5.9: Results for SLE Dataset using Synthetic Data. Given privacy preserving algorithms, MWEM PGM and MST perform similarly at similar ϵ levels.

5.5 Combined Dataset Results

The baseline model for the combined dataset, which always predicts 0, has an accuracy of 86% and an FPR of 0%. As seen in table 5.10, we get an overall accuracy of 89% however the combined model using Random Forest Classifier finds it difficult to generalize on both datasets and gets a bad accuracy on the Todd dataset.

[3] https://github.com/sdv-dev/SDV/discussions/980

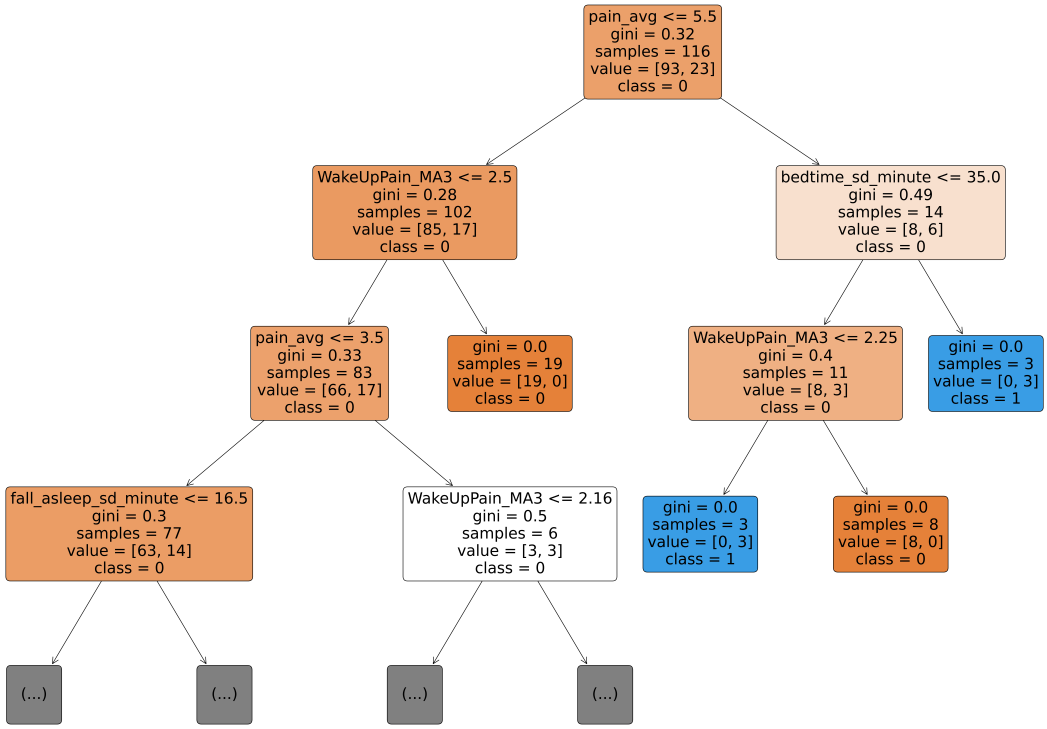


Figure 5.3: Limited Depth Decision Tree Visualization for the SLE experiments trained on "fall_asleep_sd_minute, bedtime_sd_minute, WakeUpPain_MA3, pain_avg" columns showing important columns and thresholds.

Merge Method	**Attributes**	**Avg. Accuracy SIPA**	**False Positive Rate SIPA**	**Avg. Accuracy Todd**	**False Positive Rate Todd**	**Avg. Accuracy Combined**	**False Positive Rate Combined**
Merge Method	**Attributes**	**Avg. Accuracy SIPA**	**False Positive Rate SIPA**	**Avg. Accuracy Todd**	**False Positive Rate Todd**	**Avg. Accuracy Combined**	**False Positive Rate Combined**
SIPA Diary + Todd	WakeUpPainDelta, PreviousDayWakeUpPain, WakeUpPain_MA2, WakeUpPain_MA3, WakeUpPain_MA4, WakeUpPain_MA5	97%	1.05%	79%	6.51%	89%	3.09%

Table 5.10: Results for Combined Datasets. Training is always done on combined data and the results indicate the results on the specified datasets. Combining datasets reduces the accuracy on both datasets.

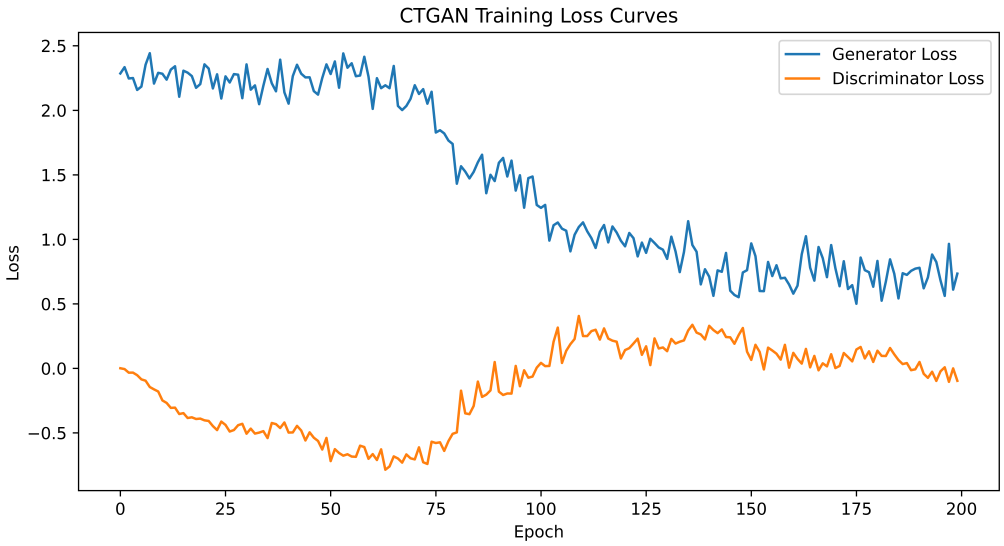


Figure 5.4: Loss Graph for CT GAN for SLE dataset [3]

Chapter 6

CONCLUSION & FUTURE WORK

In this thesis we have designed and trained machine learning models for next day pain prediction in patients with Juvenile Idiopathic Arthritis (JIA) and patients with Systemic Lupus Erythematosus (SLE). We have found that it is possible to train Random Forest classifiers that achieve high accuracy (up to 98% for JIA and 85% for SLE) with a very small false positive rate. Our classifiers are built on features that we extracted and constructed from diary and actigraphy data. The best feature sets that we identified differ between JIA and SLE, and even among two different JIA studies used in this thesis. Training a classifier on the combined data from both JIA studies did yield better results than the classifiers that we trained on the data from each study separately.

We have furthermore explored the potential of synthetic data generation for next day pain prediction. The motivation for this is that patient data is sensitive and cannot be broadly shared with researchers in data science who have the skills to train machine learning models over the data. Generating synthetic data based on the real data, and then sharing only the synthetic data, can be a good solution to make data more broadly available while mitigating privacy concerns. We manage to create synthetic data for SLE with high utility at 81% accuracy and no formal privacy guarantee with CTGAN and LLMs. We also manage to generate synthetic data with differential privacy guarantee with similar accuracy of 81% with $epison = 10$ and with accuracy of 78% but more privacy at $epsilon = 1$, which might be more appropriate for healthcare application, using MST and MEWM PGM.

For future work, it would be nice to work on doing more feature engineering, mainly on finding better ways to summarize actigraphy day data. It would be good to compare additional synthetic data generation methods. Another great direction would be towards

creating synthetic raw actigraphy readings for JIA and SLE patients. We believe this would be a great step towards solving healthcare data availability issues.

BIBLIOGRAPHY

[1] Sachin Ananth. Sleep apps: current limitations and challenges. *Sleep Science*, 14(1):83, 2021.

[2] Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. In *The Eleventh International Conference on Learning Representations*, 2023.

[3] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[4] Germain Forestier, François Petitjean, Hoang Anh Dau, Geoffrey I Webb, and Eamonn Keogh. Generating synthetic time series to augment sparse datasets. In *IEEE International Conference on Data Mining (ICDM)*, pages 865–870, 2017.

[5] Mauro Giuffrè and Dennis L Shung. Harnessing the power of synthetic data in healthcare: innovation, application, and privacy. *NPJ Digital Medicine*, 6(1):186, 2023.

[6] Micah Hartman, Anne B Martin, Benjamin Washington, Aaron Catlin, National Health Expenditure Accounts Team, et al. National health care spending in 2020: Growth driven by federal spending in response to the covid-19 pandemic: National health expenditures study examines us health care spending in 2020. *Health Affairs*, 41(1):13–25, 2022.

[7] Sharon Ho, I Elamvazuthi, and CK Lu. Classification of rheumatoid arthritis using machine learning algorithms. In *2018 IEEE 4th International Symposium in Robotics and Manufacturing Automation (ROMA)*, pages 1–6. IEEE, 2018.

[8] Aria Khademi, Yasser El-Manzalawy, Lindsay Master, Orfeu M Buxton, and Vasant G Honavar. Personalized sleep parameters estimation from actigraphy: a machine learning approach. *Nature and Science of Sleep*, pages 387–399, 2019.

[9] Joern Loetsch, Lars Alfredsson, and Jon Lampa. Machine-learning–based knowledge discovery in rheumatoid arthritis–related registry data to identify predictors of persistent pain. *Pain*, 161(1):114–126, 2020.

[10] Maria Matsangidou, Andreas Liampas, Melpo Pittara, Constantinos S Pattichi, and Panagiotis Zis. Machine learning in pain medicine: an up-to-date systematic review. *Pain and Therapy*, pages 1–18, 2021.

[11] Daniel McDuff, Theodore Curran, and Achuta Kadambi. Synthetic data in healthcare. *arXiv preprint arXiv:2304.03243*, 2023.

[12] Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the NIST contest: A scalable and general approach to differentially private synthetic data. *arXiv preprint arXiv:2108.04978*, 2021.

[13] Ryan McKenna, Daniel Sheldon, and Gerome Miklau. Graphical-model based estimation and inference for differential privacy. In *International Conference on Machine Learning*, pages 4435–4444. PMLR, 2019.

[14] Sergey I Nikolenko. *Synthetic data for deep learning*, volume 174. Springer, 2021.

[15] Aarti Sathyanarayana, Shafiq Joty, Luis Fernandez-Luque, Ferda Ofli, Jaideep Srivastava, Ahmed Elmagarmid, Teresa Arora, Shahrad Taheri, et al. Sleep quality prediction from wearable data using deep learning. *JMIR mHealth and uHealth*, 4(4):e6562, 2016.

[16] Dahee Wi, Tonya M Palermo, Elaine Walsh, and Teresa M Ward. Temporal daily relationships between sleep and pain in adolescents with systemic lupus erythematosus. *Journal of Pediatric Health Care*, 2023.

[17] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In *Advances in Neural Information Processing Systems*, 2019.

[18] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. Opacus: User-friendly differential privacy library in pytorch. *arXiv preprint arXiv:2109.12298*, 2021.

[19] Weichao Yuwen, Maida Lynn Chen, Kevin C Cain, Sarah Ringold, Carol A Wallace, and Teresa M Ward. Daily sleep patterns, sleep quality, and sleep hygiene among parent–child dyads of young children newly diagnosed with juvenile idiopathic arthritis and typically developing children. *Journal of Pediatric Psychology*, 41(6):651–660, 2016.