Entity-Centric Machine Learning: Leveraging Entity Neighbourhoods for Personalised Predictors

DISSERTATION

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von M. Sc. Vishnu Mazhuvancherry Unnikrishnan

geb. am 06.07.1989 in Manipal, Indien

Gutachterinnen/Gutachter

Prof. Dr. Myra Spiliopoulou
Prof. Dr. Ruediger Pryss
Prof. Dr. Panagiotis Papapetrou

Magdeburg, den 13.06.2024

Abstract

Recent times have seen an increase in both the rates at which data are generated, as well as the technology developed to process datasets generated at an ever increasing pace. However, most machine learning methods still apply a one-size-fits-all approach, with the models being tailored to be applied out-of-the-box on the entire dataset, and model complexity focusing on generalising optimally to the patterns without overfitting. Additionally, it is also worth noting that datasets are not monolithic - they are often comprised of repeated observations of a smaller set of objects or 'entities' over time. These entities have 'static' unchanging properties, and act as data generators to create the 'dynamic' data that is observed over time. Many current methods, however, train models over the dynamic data alone, and do not adequately exploit the static data for learning. In this work, we study ways in which machine learning methods can be 'personalised' so that the data of each 'entity' gets its own model, which incorporates the similarity of the 'static' and 'dynamic' parts of the entity. The benefits of personalisation are obvious for some fields like medicine and user generated content, and our solutions are designed for the medical domain where some of the disadvantages of entity-centred datasets express most strongly - each entity has too little data available for learning, each entity's data arrives irregularly, and each entity's data is generated at a different time than the others. Our approach towards personalisation is that each entity in the dataset gets it's own model, and we combat the sparsity of the data (each individual entity has too little data!) by augmenting the data of each entity with the data of other entities that are deemed 'similar'. In our work, we explore three main approaches to training personalised models for medical datasets.

The first part of this work explores the various ways for dealing with data sparsity, irregularity, and dealing with timestamps during training of personalised models. We explore augmenting the dynamic data of of each entity with the dynamic data of its neighbours as defined by the static data. We investigate the various ways to train the neighbourhood-augmented model, deal with timestamps, and the effect of the neighbourhood size. Our findings show that training a model on the combined dynamic data of a small number of neighbours and preserving timestamps yields the best results. We extended this method to allow the similarity to be guided using expert knowledge, and found that grouping users based on medical intuition improves the quality of the resultant models for several subgroups. A baseline that selects the neighbourhood of an entity randomly was found to be very competitive, suggesting that even though the entity-centred models exceeded the global model's performance with less data, the neighbourhood computation can be improved.

Our second approaches investigate the degree to which the dynamic data from the entities can be used to train personalised predictors. Towards this end, we test two types of approaches, one that summarises the time series so that a similarity function may be applied that can discover other similar entities, and another that groups users based on the length of their dynamic data sequences. We saw that summaries of

the dynamic data helped achieve competitive performance to the global model while exploiting <10% of the users, and that predictions can be made and personalised towards users with very short sequences on the basis of other users whose sequences are longer. Since the notion of similarity is difficult to define, we also propose an iterative neighbourhood similarity method that discovers the ideal set of users to learn a personalised model for users with short sequences.

Drawing inspiration from this result, the third part of our work focuses on discovering the optimal neighbourhood for each entity in a supervised way using validation error of the personalised models. We propose one method that searches for the optimal neighbourhood greedily in decreasing order of similarity, and found that the global model is beaten by $\approx 13\% - 15\%$ by a personalised model with our discovered neighbourhood. An analysis of the neighbourhoods themselves show that there are 'celebrity' users whose data is used by almost all others, and 'ostracised' users whose data contributes negatively to other users. Our second proposed method that removes the effect of sorting the users by similarity, however, discovers much smaller neighbourhoods, and also performs worse than the first (although better than the global model). A full comparison of the neighbourhoods and their relative quality, however, needs the help of a clinical expert. We consider the entity-neighbourhoods a part of our output, since it enables further investigations, especially in cases where the underlying similarity function is not known.

Zusammenfassung

In jüngster Vergangenheit ist ein bemerkenswerter Anstieg sowohl in der Häufigkeit der Datenproduktion als auch in der Entwicklung von Technologien zur Verarbeitung von Datensätzen zu verzeichnen. Trotz dieser Fortschritte verfolgen jedoch die meisten Methoden des maschinellen Lernens nach wie vor einen konservativen Ansatz, der darauf abzielt, ein Modell auf dem gesamten Datensatz zu trainieren und anzuwenden. Hier ist der Schwerpunkt, allgemeine Muster zu identifizieren und Overfitting zu vermeiden. Es ist jedoch wichtig anzumerken, dass Datensätze keine homogenen Gebilde darstellen; vielmehr bestehen sie oft aus Beobachtungen, welche von einer begrenzten Anzahl an Entitäten produziert werden. Diese Entitäten verfügen oft über statische, unveränderliche Eigenschaften und fungieren als Datenquellen für die Generierung der im Zeitverlauf beobachteten "dynamischen" Daten. Viele aktuelle Methoden zur Vorhersage von dynamischen Daten ignorieren jedoch statische Eigenschaften der Entitäten beim Training der Modelle. In dieser Studie untersuchen wir daher Ansätze, wie Methoden des maschinellen Lernens "personalisiert" werden können, sodass die Daten jeder "Entität" ihr eigenes Modell erhalten. Dabei wird die Ähnlichkeit der "statischen" und "dynamischen" Teile der Entität berücksichtigt. Die Vorteile der Personalisierung sind insbesondere in Bereichen wie der Medizin und für Anwendungen mit nutzererstellten Inhalten offensichtlich. Unsere Lösungen sind speziell für den medizinischen Bereich konzipiert, in dem einige der Herausforderungen an entitätszentriertes Lernen am deutlichsten zum Ausdruck kommen. Beispielsweise verfügen viele Entitäten über zu wenige Beobachtungen zum Lernen; die Daten jeder Entität treffen unregelmäßig ein; und die Daten jeder Entität werden im Vergleich zu den anderen zu asynchronen Zeitpunkten generiert. Unser Ansatz zur Personalisierung sieht vor, dass jede Entität im Datensatz ihr eigenes Modell erhält. Wir adressieren die potenzielle Datenknappheit von Entitäten, indem wir deren Daten mit den Beobachtungen "ähnlicher" Entitäten ergänzen. In dieser Arbeit untersuchen wir drei Hauptansätze zum Training personalisierter Modelle für medizinische Datensätze.

Der erste Teil der Arbeit befasst sich mit verschiedenen Möglichkeiten zur Bewältigung von Datenknappheit und zeitlichen Aspekten beim Training von personalisierten Modellen. Wir untersuchen die Erweiterung der dynamischen Daten jeder Entität mit den dynamischen Daten ihrer nächsten Nachbarn, die durch statische Daten definiert sind. Wir untersuchen die verschiedenen Möglichkeiten zum Trainieren des nachbarschaftserweiterten Modells, den Umgang mit Zeitstempeln und die Auswirkungen der Nachbarschaftsgröße. Unsere Ergebnisse zeigen, dass das Training eines Modells auf den kombinierten dynamischen Daten einer kleinen Anzahl von Nachbarn und unter Beibehaltung von Zeitstempeln die besten Ergebnisse liefert. Wir haben diese Methode so erweitert, dass die Ähnlichkeit mit Hilfe von Expertenwissen gesteuert werden kann, und haben festgestellt, dass die Gruppierung von Nutzern auf der Grundlage medizinischer Intuition die Qualität der resultierenden Modelle für mehrere Untergruppen verbessert. Eine Vergleichsmethode, welche die Nachbarschaft einer Entität zufällig auswählt, erwies sich als sehr konkurrenzfähig. Obwohl die entitätszentrierten Modelle die Leistung des globalen Modells mit weniger Daten übertreffen, deutet dieser Umstand darauf hin, dass die Berechnung der Nachbarschaft verbessert werden kann.

Der zweite Teil untersucht, inwieweit die dynamischen Daten der Entitäten zum Training personalisierter Prädiktoren verwendet werden können. Zu diesem Zweck testen wir zwei Arten von Ansätzen. Einen, der die Zeitreihen zusammenfasst, sodass eine Ähnlichkeitsfunktion zur Identifikation ähnlicher Entitäten angewendet werden kann, und einen anderen, der Nutzer auf der Grundlage der Länge ihrer dynamischen Datenfolgen gruppiert. Es wurde deutlich, dass die Zusammenfassungen der dynamischen Daten dazu beigetragen haben, eine mit dem globalen Modell vergleichbare Leistung zu erzielen, wobei <10% der Nutzer verwendet wurden. Des Weiteren hat sich gezeigt, dass Vorhersagen für Nutzer mit sehr kurzen Sequenzen auf der Grundlage anderer Nutzer mit längeren Sequenzen gemacht und personalisiert werden können. Da der Begriff der Ähnlichkeit schwer zu definieren ist, schlagen wir auch eine iterative Methode der Nachbarschaftsähnlichkeit vor, welche die ideale Menge von Nutzern entdeckt, um ein personalisiertes Modell für Nutzer mit kurzen Sequenzen zu lernen.

Ausgehend von diesem Ergebnis konzentriert sich der dritte Teil dieser Arbeit auf die Entdeckung der optimalen Nachbarschaft für jede Entität auf überwachte Weise unter Verwendung des Validierungsfehlers der personalisierten Modelle. Wir schlagen eine Methode vor, die 'greedy' nach der optimalen Nachbarschaft in abnehmender Reihenfolge der Ähnlichkeit sucht, und haben festgestellt, dass das globale Modell von einem personalisierten Modell mit der von uns entdeckten Nachbarschaft um $\approx 13\% - 15\%$ geschlagen wird. Eine Analyse der Nachbarschaften selbst zeigt, dass es "prominente" Nutzer gibt, deren Daten von fast allen anderen genutzt werden, und "diskriminierte" Nutzer, deren Daten einen negativen Beitrag für andere Nutzer leisten. Unsere zweite vorgeschlagene Methode, die den Effekt der Sortierung der Nutzer nach Ähnlichkeit beseitigt, entdeckt jedoch viel kleinere Nachbarschaften und schneidet auch schlechter ab als die erste (wenn auch besser als das globale Modell). Für einen vollständigen Vergleich der Nachbarschaften und ihrer relativen Qualität ist jedoch die Hilfe eines klinischen Experten erforderlich. Wir betrachten die Nachbarschaften einer Entität als Teil unseres Ergebnisses, da sie weitere Untersuchungen ermöglichen, insbesondere in Fällen, in denen die zugrunde liegende Ähnlichkeitsfunktion nicht bekannt ist.

Contents

List of Figures

List of Tables

List of Acronyms

AI	artificial intelligence
AIC	Akaike Information Criterion
AR	Autoregression
ARMA	Autoregressive-Moving-Average
BIC	Bayesian Information Criterion
CBT	Cognitive Behavioural Therapy
DTW	Dynamic Time Warping
EMA	Ecological Momentary Assessment
EMI	Ecological Momentary Intervention
HMM	Hidden Markov Model
kRE	k-Random Entities
MA	Moving Average
MCS	Mobile Crowd Sensing
mHealth	mobile health
Mini-TQ	Mini-Tinnitus-Questionnaire
PAA	Piecewise Aggregate Approximation
RCT	Randomised Controlled Trial
RQ	Research Question
SAX	Symbolic Aggregate Approximation
THI	Tinnitus Handicap Inventory
TSCHQ	Tinnitus Sample Case History Questionnaire
TYD	TrackYourDiabetes
TYT	TrackYourTinnitus
UNITI	Unification of Treatments and Interventions for Tinnitus Patients
WHO	World Health Organisation

1. Introduction

Machine learning and artificial intelligence (AI) have touched almost every industry ranging from self driving cars to automated trading. Advances in storage and processing capabilities have allowed computers to tackle ever larger datasets, and support the creation of more and more complex models. However, as the models and methods become more sophisticated, the basic approach has remained "train the best possible model from all available data".

Many datasets, however, contain additional information along with the instances that reflect innate properties of the instances. For example, reviews of products, physiological measurements from humans, recordings from multiple sensors etc. (i.e., 'entities') will all have the data as well as additional metadata that are of relevance to the prediction problem. The most obvious use of such metadata is that multiple measurements from a 'product' or 'human' over time give you more information about the future of that product or person, than of all products / people. Training models that take such dependencies into account is only recently gaining popularity [79, 64, 93].

The fact that pursuing of predictive performance alone is insufficient is indicated by the recent rise in interest towards model explainability. Search interest for "Explainable AI" (XAI) from Google Trends over the last 10 years is shown in Figure 1.1, where it can be seen that there is a very strong increase in interest for explainable models from 2018[1]. The trend in scientific publications is broadly similar, with many conferences introducing separate tracks for XAI.

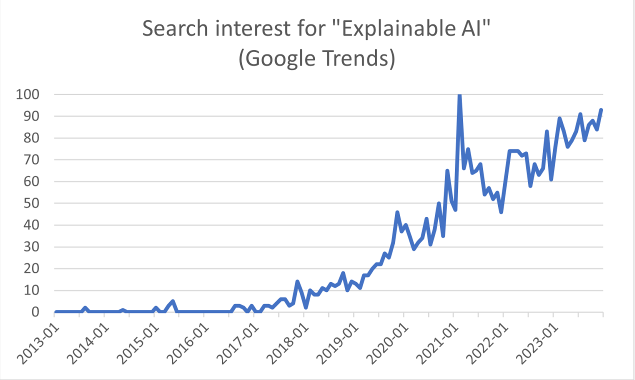


Figure 1.1.: Search interest for the term "Explainable AI" from Google Trends for the last 10 years

A recent survey conducted by the Federal Ministry for Economic Affairs and Climate Action [53] also found that the practitioners from different industries place different levels of emphasis on explainable models. Respondents from healthcare, finance, and

[1] Data pulled on 02.01.2024

manufacturing rated local explainability as the most important, with the majority of participants from the healthcare industry even suggesting that explainability should be mandatory. However, XAI focuses primarily on explaining predictions for individual instances in the data as predicted by a model that was trained on all available data.

Data sources with longitudinal observations for collections of entities are not uncommon. However, most machine learning approaches do not explicitly model these entities, choosing rather to use all available data and rely on the larger dataset to achieve generalisation. It is perhaps unsurprising, however, that the most convincing argument for explicitly modelling the entities separately comes from the field of medicine as 'personalised health' [65], where it is said that given the significant inter-individual variability in the area of health, "when estimating symptoms, responses to medication, or heart-rate profiles, it would be impossible to make useful predictions without personalization". Fields like e-commerce and social media also stand to gain from modelling and personalising towards users of their platforms, but it is necessary to first call out an important deficiency of the keyword "personalised", and how this work uses it in a way that is more narrowly defined than some other works in the literature.

While the goal of a personalised model in this work aligns with that presented by McAuley in the book [65]. The book highlights the fact that the idea of personalisation is not yet mature enough for all researchers to have converged upon the same definition of what it means. This is clear from the taxonomy they present (see [65], Section 1.7.2) that even a recommender system that recommends products to a user (because similar users might have bought another product) arguably makes personalised predictions. i.e., the system uses the user data, but does not include explicit parameters or a model towards just that user. This is called *contextual personalisation*, while our approach aims to train personalised models by actually training separate models for each individual (or 'entity') in the dataset. This approach is called 'model-based' personalisation in [65]. In our work, the phrase 'personalised model' refers exclusively to approaches for model-based personalisation. This is because our methods target not only generating accurate personalised predictions, but also on delivering a 'neighbourhood' for each personalised model. While not discussed [65], we believe that delivering the neighbourhoods along with personalised models allow for closer inspection of the models, and also enables further analyses of the neighbourhoods themselves, which might reveal hitherto unknown relationships between the entities (or in the case of personalised medicine, patients) that comprise them. We expect that such approaches are especially relevant for diseases where the underlying similarity between patients is not yet fully understood.

Although still an open research question, the need for personalised solutions is being increasingly acknowledged in the world of mobile health (mHealth) apps [96]. mHealth apps are used by users (who may be diagnosed patients) to help self-manage their lifestyle, or a variety of diseases. Popular areas where mHealth apps are common are: apps that promote healthy lifestyles, apps for endocrine and nutritional disease, apps for psychological and behavioural issues. The use of these apps helps patients learn about and manage their symptoms[51], while also helping doctors keep track of their patients remotely, and to be alerted in case interventions are required [98]. The use of mHealth apps also helps doctors monitor patients outside of a hospital, and therefore get an increased insight into the dynamic presentation of diseases. This point in particular can be valuable for clinicians, since personalised models and a data-driven understanding of patient similarity might help the clinicians understand the disease dynamics better.

Towards this goal, we aim to develop personalised modelling methods for datasets comprising multiple observations from multiple ‘entities’ (primarily patients, but also sensors, products, weather stations, etc.), where the personalised model of each entity is chosen so as to:

- maximise predictive quality of the entity-specific model
- deliver a set of entities over whose data the entity specific model should be trained.

1.1. Challenges in developing personalised predictors

This work focuses on developing personalised predictors assuming a panel-data-like data source where multiple entities are tracked over time. Although technically accurate, we do not use the term ‘panel data’ to describe our dataset because in addition to observing time-changing or ‘dynamic’ properties of each entity over time, we also additionally assume that each entity is also described by some ‘static’ unchanging properties. To summarise, an entity within the dataset is described in two modalities, or data spaces: one that is relatively ‘static’ and unchanging (like the gender and date of birth of a patient), and the other which is ‘dynamic’ (like daily blood sugar levels, weight, etc.). Although the methods we study are applicable to datasets from several domains, the focus is on working with medical data because developing personalised predictors is of high relevance in the field of medicine. In a medical context, the ‘static’ data might come from a (relatively infrequent) hospital visit, and the ‘dynamic data’ could be generated by remote monitoring of the patient using mHealth applications.

Personalised predictors are especially interesting in the field of medicine partly because the idea of personalised medicines are rooted in ‘biological realities’ like unique molecular, physiological, environmental and behavioural fingerprints [30], but also because healthcare data is increasingly digital, and being made accessible to interdisciplinary researchers. In addition to interventions, therapies and prophylactics that are tailored to individuals, it is also reasonable to expect that methods that generate predictions for each individual separately may reveal hitherto unknown similarities between individuals with similar predictions. This would of great benefit to healthcare professionals for gaining a deeper understanding of the disease, through individuals that experience it similarly.

However, the development of personalised predictors involves selecting the best subsets of data to learn from, and given the heterogeneity of medical data, the formulation of a similarity function is critical. This has been acknowledged to be challenging, since the concept of similarity may be context dependent, with different definitions based on whether goal is to classify, cluster, or detect outliers [94]. Many define similarity for patients using atemporal data with similarity measures like pearson correlation, cosine similarity, eucliedean distance, etc., while similar temporal data can be found using methods like DTW [61]. Some works like [39] also explore combining unstructured data like text clinical notes with demographic and temporal data using dimensionlity reduction methods that extract relationships between the multiple data sources while excluding noise.

This work aims to tackle the question of developing personalised predictors in the panel-data-like scenario, where multiple 'entities' are observed over time. While the term 'panel data' is the closest fit, it is not perfect . The key characteristics of our data are described below:

- each sequence / time series in the data are generated by one of a collection of 'entities'. We call this 'dynamic data'.
- apart from the multivariate information available over time, the entities are also described by static covariates - we refer to this as 'static data'. In extreme cases, different entities may be described by overlapping, but possibly different feature spaces.
- not every entity is observed for the same length of time,
- the range of variability among the shortest to the longest entity-level sequences is very large (up to 3 orders of magnitude),
- the time series / sequence of observations generated by each entity are not regularly spaced within an entity and across entities, and
- not only are all entities not observed over the same length of time, the first and last dates of entities within the data might be non overlapping.

Several industries create datasets like the one described above, typically wherever the data is generated by voluntary user interactions with a system - for example, e-commerce, health, and mobile health usage data. Given that a small proportion of entities might contribute a disproportionately large part of the dataset, it is necessary to design methods that accommodate the fact that most entities in the dataset have too little data. This necessitates that the personalised models of entities that contribute little data are augmented by the data of other entities that are similar. Since the majority of this work focuses on personalised predictors in datasets comprising actual humans interacting with an mHealth application, we will use the words 'entities' and 'users' interchangeably.

However, in addition to the personalised models themselves, our methods also deliver the list of entities that contribute to each model. Apart from making it easier to service data deletion requests from users, it is also hoped that the neighbourhoods would themselves help a clinician develop hypotheses on what might make patients similar, or get a deeper insight into the disease dynamics.

1.2. Research Questions

As described in Section 1.1 above, the main goal of this work is to develop methods for datasets with multiple observations for entities over time, where each entity is described by two data spaces - one relatively fixed or 'static', and the 'dynamic', which samples more volatile properties of the entity over time. The core question of our work in this context is:

> How to select the most relevant neighbours of an entity when developing personalised models?

We break down this question into the following three Research Questions (RQs):

RQ1 To what extent can static data similarity be exploited when training personalised models? To what extent does expert knowledge contribute towards improving neighbourhoods?

RQ2 To what extent does similarity in the dynamic domain guide the neighbourhood selection process?

RQ3 To what extent can the notion of similarity be supervised? To what extent does a neighbourhood based on supervised similarity improve personalised predictors?

1.3. Summary of Scientific Contributions

Our main contributions are methods that train personalised models for an entity in an environment of strongly varying amounts of data per entity. The list of contributions towards each of the research questions are listed below:

RQ1 How to exploit entity similarity from the 'static' domain when training personalised models?

- We show that personalised predictors that train on a subset of entities can outperform global model trained on data of all entities. We also show that similarity in static data can serve as a guide to the neighbourhood selection process.
- We show that the neighbourhoods discovered by methods exploiting static data can be further tuned with expert knowledge (when available) to yield subgroups with different prediction accuracy. This is demonstrated on the tinnitus dataset, where the results obtained reinforce the established knowledge that known subgroups of patients with anomalous behaviours should not be used to inform each others' models.

RQ2 How to include data from dynamic / time series domain to guide the neighbourhood selection process?

- We show that the dynamic data can be summarised using hidden markov models and granger causalities, which can then be used to 'summarise' patients as fixed-length vectors. Our results show that these fixed-length representations can be used to construct neighbourhoods that match the performance of the global model while relying on a fraction of the total number of users in the dataset.
- We show that behavioural information from the dynamic space can help develop personalised models. Users with little data ("short" users) can be predicted by those that have more ones ("long" users), and also that as the short users get longer, the personalised models can progressively incorporate their data to improve the personalised models.

- Different short users might need different neighbourhoods, but also different neighbourhoods of different neighbourhood sizes. These neighbourhoods can be incrementally discovered guided by static data similarity.

RQ3 Can the notion of similarity be "learned" in a supervised manner?

- Since both static and dynamic data might be insufficient for discovering neighbourhoods that improve predictive performance, we show that a supervised notion of similarity that chooses neighbourhoods maximising predictive performance can outperform global models, while delivering personalised models with personalised neighbourhood sizes.
- Our method delivers a small ($\approx$ 10%) but significant improvement to about 85% of the users.
- The neighbourhood selected by the supervised similarity confers a small but significant improvement in predictions for most more than 85% of the users.
- Apart form the improvements in predictive performance, the supervised similarity also reveals that there is a small set of users who do not contribute positively to most users' neighbourhoods. This unexpected result could serve as a starting point for clinical investigations.

1.4. Outline of the Thesis

The following chapters of this thesis begin with a discussion of some necessary background to better understand the context in which this work is placed, followed by three chapters that explore the three research questions listed in Section 1.2.

- Chapter 2 discusses some necessary medical background, followed by an overview of the main datasets used in this thesis, and then concludes with some definitions of frequently used terms.
- Chapter 3 is the first part of this thesis, and introduces our main approach to fitting personalised models in the context of panel-like data where the amount of data per entity varies strongly. We explore several ways in which personalised models can be trained and compare the results of each to discover the best performing one. The results collected from our investigations in Chapter 3 serve as the foundation for the investigations and design decisions of Chapters 4 and 5.
- Chapter 4 discusses the next part of this thesis, where we explore two main methods to exploit the dynamic data to build patient neighbourhoods - the first uses HMMs and Granger causalities to summarise the sequences of unequal lengths, and the degree to which these user summaries can be used to discover neighbourhoods. The second set of methods explores grouping users on the basis of the length of their interaction to learn models for users with short sequences using the data of those who have long ones.
- Chapter 5 builds upon the results of Chapter 4 to propose methods that exploit the dynamic data as a validation set and discovers a neighbourhood for each user on the basis of a supervised notion of similarity ($\equiv$ reduction in error).

- Chapter 6 presents a summary of the results from each of our three main approaches, and discusses several possible avenues for future explorations.
- The appendices in A.1, A.2, and A.3 contain some supporting information for our experiments from Chapters 3, 4, and 5. The supporting information are not required reading to understand the core contributions of this thesis. The appendices are followed by the bibliography.

2. Underpinnings

Our collaborations with medical experts influenced the design of many of our proposed solutions, so some of the more salient points of the diseases involved are covered in the sections below. They include an overview of tinnitus and diabetes, followed by the concept of the Ecological Momentary Assessments (EMAs). We conclude in 2.3 by formalising some definitions that are used often in this thesis.

2.1. Medical underpinnings

Although not strictly relevant to a technical reader of this manuscript, a rather detailed overview of tinnitus, its causes, and treatments is provided below. This is primarily to impress upon the reader the high degree variability in symptom presentation, causes, and the need for varied (and personalised) therapeutic approaches that accommodate for the various ways in which tinnitus can be caused and expressed. The executive summary at the beginning of Section 2.1.1 should highlight the main takeaways.

2.1.1. Tinnitus and its treatments

- Tinnitus is a psychoacoustic disorder that affects 10-15% of the population, and about 1-2% claim that tinnitus has a moderate to severe affect on their quality of life.
- Tinnitus is a complex disorder with many causes ranging from hearing loss to trauma and infections. The perception of tinnitus is also highly heterogeneous, and this has hampered the development and testing treatments.
- Several known treatments exist, but treatments that try to alleviate symptoms are less effective than those that teach a patient to manage their symptoms and its effect on their psychological outlook.
- The high degree of variability in symptom presentation, tinnitus distress, and treatment success suggest that treatments need to be personalised to the individual case.
- Since tinnitus distress is sometimes independent of (and easier to treat than) tinnitus loudness, physicians may be more interested in studying tinnitus distress.

What is tinnitus?: Tinnitus, derived from the Latin word *tinnire*, meaning "to ring" is a neuropsychiatric disorder characterised by the phantom perception of sound, i.e., a perception of sound in the absence of an external stimulus. The perceived sound my be confined to one or both ears, and the location of the perceived sounds may also be perceived accordingly, including being described as "inside the head".

The nature of the sound is also varied, with patients describing their tinnitus as ringing, buzzing, hissing, roaring, clicking, etc. [17]. The sound may also be constant or rhythmic/pulsing, with the latter case being suspected of patients for whom the tinnitus has an origin in the vasculature [4].

Prevalence, impact, and risk factors: Prevalence of tinnitus is mostly best studied in the USA and Europe, and studies very in their estimates of tinnitus prevalence, but most studies estimate the prevaence of tinnitus at 10-15% of the population . About 1.6% of tinnitus patients rate their tinnitus as 'very annoying', with a further 0.5% claiming debilitating effects on quality of life. Tinnitus patients may suffer from a variety of psychological problems ranging from hyperacousis to insomnia, trouble concentrating, frustration and depression[55]. The prevalence of tinnitus has been found to be similar for men and women, with increasing prevalence of troublesome with increasing age [4][55].

The risk factors associated with tinnitus are also several and varied. Although hearing loss is the primary risk factor, many patients with hearing loss do not report tinnitus, and not all patients with tinnitus have hearing loss[4]. An epidemiological cohort study in ageing identified hearing loss, noise exposure, head injury, depressive symptoms, arthritis, and use of certain medications as the risk factors[77]. However, other factors like smoking, alcohol consumption, hypertension, etc. are also listed as risk factors [4]. A comprehensive list of risk factors is presented in Table 2.1 (reproduced from [4]).

Category	Risk Factor / Co-morbidity
Otological (infectious)	Otitis media, labyrinthitis, mastoiditis
Otological (neoplastic)	Vestibular schwannoma, meningioma
Otological (labyrinthine)	Sensorineural hearing loss, Ménière's disease, vestibular vertigo
Otological (other)	Impacted cerumen, otosclerosis, presbyacusis, noise exposure
Neurological	Meningitis, migraine, multiple sclerosis, epilepsy
Traumatic	Head or neck injury, loss of consciousness
Orofacial	Temporomandibular joint disorder
Cardiovascular	Hypertension
Rheumatological	Rheumatoid arthritis
Immune-mediated	Systemic lupus erythematosus, systemic sclerosis
Endocrine & metabolic	Diabetes mellitus, hyperinsulinaemia, hypothyroidism, hormonal changes during pregnancy
Psychological	Anxiety, depression, emotional trauma
Ototoxic medications	Analgesics, antibiotics, antineoplastic drugs, corticosteroids, diuretics, immunosuppressive drugs, non-steroidal anti-inflammatory drugs, steroidal anti-inflammatory drugs

Table 2.1.: Known tinnitus risk factors and comorbidities, reproduced from [4]

Treatments: When the tinnitus has a known cause such as trauma or infection, the primary treatment is to treat the cause, but tinnitus is known to sometimes persist even after the underlying cause has been treated. Given the wide variation in presentation and risk factors, and lack of a standard outcome measure have also affected the quality of evidence for tinnitus treatments and therapies [4].

Once the underlying pathology has been treated, the standard treatments for tinnitus are patient education[33], sound therapy, counselling, Cognitive Behavioural Therapy (CBT)[16] and combinations thereof [4]. However, several emerging treatments are also under investigation: tinnitus maskers and hearing aids [102], brain stimulation[21], and even surgery or cochlear implants, when applicable[5]. Medications have met with more limited success, except for dental anaesthetics, which have limited use because of high risk and limited modes of delivery[4]. A brief description of the main auditory and psychological treatments is given below:

- Auditory Treatments:
  - Hearing aids and cochlear implants: Hearing aids can be used to treat patients who have tinnitus as well as hearing loss. However, the efficacy of hearing aids may depend on the tinnitus frequency and on other otological conditions. [55]. For patients who have profound bilateral sensorineural hearing loss, cochlear implants have been shown to suppress tinnitus [5]. This also works for cases where the hearing loss is severe but only in one ear. It is expected that the restoration of input to the central auditory system is the reason for tinnitus suppression.
  - Environmental Sound generators and tinnitus maskers: Environmental sound generators and maskers are used to create sounds of the sea, rain, white noise, etc., and the primary purpose is to play a relaxing sound that masks sound of tinnitus. These devices are designed to fit behind the ear, and typically allow for the frequency and loudness to be adjusted to the tinnitus. These features may also be integrated into a hearing aid. Although popular, such devices are shown to bring only limited benefits [37].
  - Individualised sound stimulation: Personalised sound stimulation follows three approaches derived from complementary notions of the types of sound that reduce tinnitus volume. The first is based on the notion that tinnitus fills in the areas of the audio spectrum that are challenged by hearing loss. Towards this end, auditory stimulation consisting of music adapted to the frequencies of hearing loss were played back to the patients along with their regular counselling sessions. These patients improved more than similar patients treated with non-personalised noise stimulation[20], but these results were later refuted by a controlled study[130]. The second approach creates music that has frequency ranges around the tinnitus frequency suppressed, and this was shown to achieve tinnitus volume reduction as well as decreases in activity levels of the auditory cortex[78], but the sample sizes in the study are small[55]. The third approach suppresses the tinnitus frequencies but enhances frequencies just above and just below the tinnitus frequencies, with the goal of renormalising tinnitus related auditory neuronal synchrony[119]. However, sample sizes are still small.
- Psychological Treatments:
  - Counselling and psychoeducation: Since there is no universal reliable

cure for tinnitus, psychoeducation techniques aim rather to habituate the patient to the phantom sound. Counselling aims to arm the patient with information and advice on how to achieve habituation, and how to better cope with the psychological consequences of tinnitus in patients' personal, social and occupational lives. Counselling aims to help individuals demystify their tinnitus, and also helps ensure continued compliance with other treatment strategies. However estimating the effect of counselling has been hard [55].

- Tinnitus retraining therapy (TRT): TRT aims to habituate the patient to tinnitus by teaching the patient to reclassify tinnitus as a neutral stimulus, while using sound therapy to reduce its intensity. This is based on the assumption that tinnitus is caused by abnormal neurophysical activity and connectivity in the auditory and non-auditory central nervous circuits [43]. However, the efficacy of TRT is disputed [82].
- Cognitive Behavioural Therapy: Rather than treat the tinnitus symptoms, CBT aims to reduce tinnitus distress by changing the emotional and behavioural responses to the tinnitus symptoms. CBT involves education, relaxation training, and exposure, which are then used to modify patients' responses to symptoms. CBT has been shown to reduce depression and improve quality of life, even when it did not reduce tinnitus volume [16].

As is clear from the brief summary above, tinnitus is a disease with a high degree of heterogeneity in both patient presentation as well as treatment success. As a psychoacoustic disorder, treating the symptom alone is sometimes not sufficient, and patients stand to benefit from a more personalised approach. The degree of distress from tinnitus can often be managed even when mitigating the severity of symptoms may be impossible, therefore, tinnitus distress is used as the variable of prediction interest rather than tinnitus loudness.

2.1.2. Diabetes

The discussion of diabetes in this work is more cursory, since the disease is much more well known and the average reader is already familiar with the disease. Although different from tinnitus in terms of the psychological distress, diabetics are not exempt from the psychological affect of the disease, since it restricts many facets of peoples lives, including diet, need for exercise, etc. It is also obvious that personalisation can also help in this domain, because like the psychological affect of diabetes, people are also highly individual in the degree to which they maintain good lifestyle habits (diet, exercise, etc.), and two people may achieve highly similar trajectories in the disease with totally lifestyles (for example, a severe diabetic who works very hard to keep in good health, vs a mild diabetic who is more casual in managing his disease). As a chronic lifelong disease relying primarily on lifestyle changes, technology enabling monitoring of a patient remotely will help doctors and patients themselves get a better understanding of the disease. The impact of these technologies can be expected to increase as the world population ages, increasing the incidence of diabetes, and the stress that is expected to be added to an already stressed healthcare system.

A summary of the main aspects of the disease, its prevelance, risk factors, and treatment are summarised below. All information is summarised from the World Health Organisation (WHO) global report on diabetes[97].

What is diabetes?: As described by the WHO global report[97], diabetes is a chronic metabolic disorder characterised by the insufficient production of insulin. It has two main types, Type 1 and Type 2. Type 1 diabetes occurs in individuals where the pancreas do not produce enough insulin, and type 2 occurs in people whose body has developed resistance to insulin in the blood.

Prevalence, impact, and risk factors: It is estimated that around 1 in 10 adults have diabetes, and the number is expected to continue growing as the world population ages. The number of diagnosed diabetics has risen from 180 million to more than 420 million from 1980 to 2014. Age standardised prevalence has also doubled, and the majority of cases are expected to be type 2 diabetics. The prevalence is also expected to increase with decreasing poverty around the world.

Type 1 diabetes is most common among children and young adults, but no clear cause is yet known, apart from genetic susceptibility. Type 2 diabetes, on the other hand, is a lifestyle disease, and has several risk factors - genetics, being overweight or obese, low levels of physical activity.

With 1.5 million deaths with diabetes as the direct cause in 2012, shorter life expectancy and disability is a high risk from poor management of diabetes. Even when not serious enough to cause death, uncontrolled diabetes can damage blood vessels and nerves, cause loss of vission and kidney function, cause heart attacks, strokes, and also higher risk of amputations in lower limbs.

Treatments: As a lifelong illness, diabetes care needs to be continuous. Patient education on better diet, physical activity and monitoring are the main recommendations. Medications exist for glucose management (including injecting insuling), cardiovascular diseases, etc., and periodic examinations to detect vascular and complications in the nervous system are recommended.

2.2. Mobile Health Solutions & Ecological Momentary Assessments

2.2.1. mHealth

The Global Observatory for eHealth from the WHO defines mHealth as a “medical and public health practice supported by mobile devices”[80]. With more than 2.5 billion people worldwide estimated to have a cellphone, the percentage of population with access to cellphones grows faster than the healthcare services. This proliferation of devices that provide cheap and convenient access to specialist clinical diagnosis and treatment advice stresses the potential of mHealth solutions. A recent study [98] that investigated the clinical value of mHealth for patients has said that although no clear clinical guidelines on how to use mHealth to add value to care delivery, the various mHealth apps can be summarised to belong to one of the following categories based on their value proposition:

- apps that improve accuracy and accuracy of diagnostics
- apps that deliver personalised treatment regimes
- apps that provide behavioural change advice

- apps that improve access to therapies (like CBT)

Although not very effective at diagnostics [112], mHealth apps for some specific tasks like screening melanomas have been shown to have up to 80% diagnostic sensitivity[14]. Apps that are designed to elicit behaviour change have been when to improve both weight loss and physical activity outcomes[24]. Apps have also helped type 2 diabetics adhere to treatment protocols better, improving their glycaemic control[135]. Although not scientifically quantified yet, anecdotal claims of improved communication between doctor and patients are also a possible additional benefit[98].

The WHO report [80] mentions mHealth apps that support mny more functionalities, including apps that send out reminders to patients to improve adherence to medication prescriptions. One such other category of mHealth apps relevant to this work is those that enable patient monitoring of a patient's symptoms.

2.2.2. Ecological Momentary Assessments (EMAs)

Mobile crowdsensing is a new paradigm which, analogously to crowdsourcing, gathers data quickly from a large pool of available human 'sensors' (crowdsensing participants). Within this growing field, mobile crowdsensing for healthcare applications is emerging as a new way to collect useful healthcare data for research, while also enabling the patient towards better self management.

The taxonomy presented in [88] lists traditional crowdsensing application as belonging to one of three categories: participatory sensing, opportunistic sensing, and other works that present a crowdsensing infrastructure approach without falling into either category.

Participatory sensing advertises sensing tasks (eg, measure noise levels or temperature at a particular location) on a crowdsensing platform, and users of the platform may opt in to provide an answer. The answers from different users need to be combined in the best possible way, and the best users to accomplish a task must be found under constraints of time. Users are typically incentivised by monetary rewards, a societal commitment towards the outcome, etc. [88]. Opportunistic sensing, on the other hand, eliminates the opt-in nature of the sensing task - the users simply install an app on their phone, and the app collects the information it needs in the background. It is clear that the range of tasks that can be accomplished opportunistically are more limited, i.e., it might be easy to use the microphone on a cellphone to measure noise levels, but measuring temperature might be impossible due to hardware limitations. The overarching goals, however, remain the same as participatory sensing, in that the best user for the task, as well as the best way to combine many responses of multiple users still remains.

While traditional crowdsensing applications are geared more towards the accomplishment of a certain task, it is said that mobile crowd sensing for healthcare is different [88], since the focus is taken away from the task, and placed on the user instead. i.e., unlike in traditional crowdsensing where it does not matter which exact user accomplishes a task, mobile crowdsensing for healthcare is about collecting patient-specific information. This also has the consequence that crowdsensing for healthcare benefits from incentivising user longevity and loyalty, while traditional crowdsensing rewards users' interaction intensities. Study of user adherence in this context is not only important, but also challenging [105, 106, 107].

Ecological Momentary Assessments (EMAs) are an example of mobile crowdsensing tools that can be used by patients to keep track of symptoms and to better understand their disease, while allowing clinicians and researchers to gather additional data about the disease[88]. mHealth apps that use EMAs allow for data to be gathered quickly and easily, and then compare this data to other users in the "crowd". Such tools are valuable in the management and treatment of chronic diseases, since the researchers gain an additional insight into the data, while the patient is better informed about self management.

Initially called "experience sampling" [18], EMAs are the more recent keyword that capture a concept known also as "ambulatory assessments". Unlike more traditional forms of assessment, EMAs measure experiencesas the subject goes about their daily life, making them particularly well suited to studying human perception, emotions, and cognition. This is because the self-reported data is unlikely to be affected by biases like retroactive recall, memory decay, and mental reconstruction [25, 104].

It is further reported that using EMAs to measure tinnitus over two [34] or four weeks [103] were not found to significantly alter the perceived loudness or distress from experiencing the disease. It is argued in [104] that this is an important prerequisite to studying tinnitus, since it suggests that using the process of using EMAs neither alters the symptom severity, nor introduces a systematic measurement bias, as would be the case if patients using the app for longer would be differently affected than users using the app for a shorter period of time. Additionally, a majority of the users acknowledged that EMAs are a good tool for measuring the tinnitus variability in and across individuals.

EMAs collected through mHealth applications have already helped understand tinnitus better. It has been found that emotional arousal and valence mediate tinnitus [85], and that people with high emotional variability (both in term of how their emotions change in intensity, and how quickly they change from negative to positive emotions) experience more tinnitus distress [84]. Tinnitus was also shown to be affected by time of day, with higher levels of distress experienced at night[86].

2.2.3. EMA apps used in this work

In this work, we use several mHealth applications targeted at tinnitus patients (we use the term patients and users interchangeably, although it is important to note that not all users of the app may be diagnosed with tinnitus by a physician), and one app targeted at patients of diabetes. We give a short description of each below, highlighting only the main points. The full description of the datasets is given in the following chapters where they are used, since each work may use data pulled at different points in time, and therefore have more users and/or more data from existing users.

TrackYourTinnitus

As described in Section 2.2.2, TrackYourTinnitus (TYT) is an mHealth platform targeted at people suffering from tinnitus. It is necessary to call out that not all users may have been medically diagnosed with tinnitus, either because their tinnitus might not be severe enough to warrant seeking medical treatment, or because the users do not have easy access to medical care for tinnitus (for example, because of decreased mobility due to age). We assume that all users of the system do indeed suffer from tinnitus, and therefore use the words 'user' and 'patient' interchangeably. The goal of the system [89] is to make it easier to collect longitudinal datasets for tinnitus using EMAs from a large number of patients.

TYT is accessible to users through downloading and installing a smartphone application. After installing the app, the users go through a registration process where they fill in some questionnaires that provide some questionnaires that provide a 'static' assessment of their symptom severity - i.e., you have symptom severity as assessed by some clinically validated questionnaires at the beginning of the users' interactions. Registering for TYT is also possible through a website. However, the EMA functionality is only available through the app.

The EMAs are collected through a notification generated by the app. Upon receiving the notification, the user is prompted to answer the EMA questionnaires that collects the assessment of the momentary severity of tinnitus. Once the user has filled in the static questionnaires, the user is guided through a setup process where they provide the time ranges and the days of the week when they are willing to receive notifications. The user can also set up the maximum number of notifications per day (up to 12). Once all this information has been collected, the TYT ensures that the user is prompted at random timepoints (within the predefined allowed periods) to answer the EMA questionnaire. The randomisation ensures that the assessment is indeed 'ecological', since the patient cannot predict when they will be alerted, and can therefore not prepare for it (by moving to a quiet space, for example).

The questions that are part of the EMA questionnaire in TYT are listed in Table 2.2 (translated from German, as listed in [47]).

Question	**Type**
Do you perceive tinnitus right now?	Binary (Yes/No)
How loud is your tinnitus right now?	Slider [0–1]
How distressing is your tinnitus right now?	Slider [0–1]
How is your mood right now?	Slider [0–1]
How aroused are you right now?	Slider [0–1]
How stressed are you right now?	Slider [0–1]
How much are you concentrating on things right now?	Slider [0–1]
Do you feel unstable at the moment?	Binary (Yes/No)

Table 2.2.: The EMA questionnaire used in the TrackYourTinnitus app.

All questions are answered with a slider, except for the first, which is binary with Yes/No. The app also collects additional information in the form of the ambient noise level as measured by the built-in device microphone, so that the effect of ambient noise of tinnitus can be investigated.

An overview of the full TYT platform is shown in 2.1 (image reproduced from [89]).

TinnitusTips

The TinnitusTips app is part of the TinnitusCare mHealth framework, and is an extension of the TYT app that introduces a psychoeducation module. Similarly to the TYT app, the TinnitusTips app also collects an initial 'static' assessment of symptom

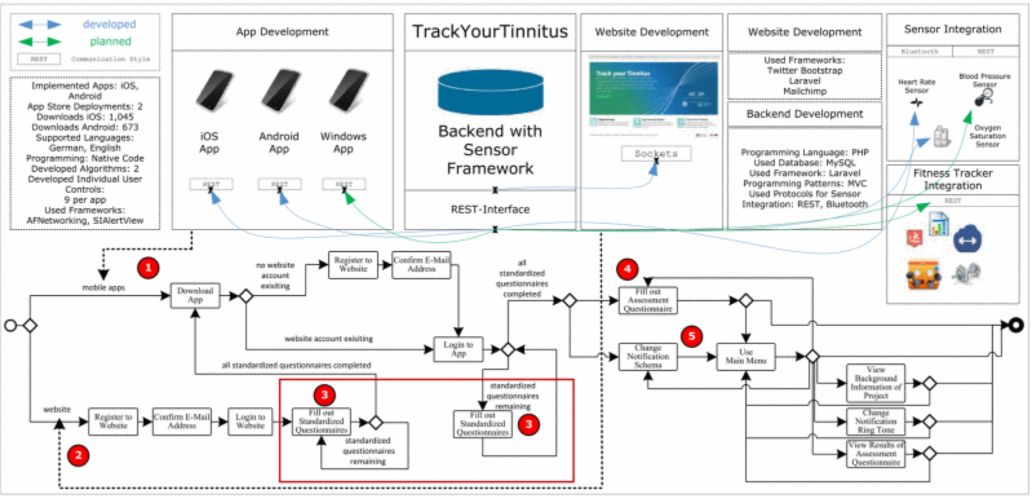

developed
planned
REST
Communication Style
Implemented Apps: iOS, Android
App Store Deployments: 2
Downloads iOS: 1,045
Downloads Android: 673
Supported Languages: German, English
Programming: Native Code
Developed Algorithms: 2
Developed Individual User Controls: 9 per app
Used Frameworks: AFNetworking, SIAlertView
App Development
iOS App
Android App
Windows App
REST
TrackYourTinnitus
Backend with Sensor Framework
REST-Interface
Website Development
Sockets
Website Development
Used Frameworks: Twitter Bootstrap Laravel Mailchimp
Backend Development
Programming Language: PHP
Used Database: MySQL
Used Framework: Laravel
Programming Patterns: MVC
Used Protocols for Sensor Integration: REST, Bluetooth
Sensor Integration
Bluetooth
REST
Heart Rate Sensor
Blood Pressure Sensor
Oxygen Saturation Sensor
Fitness Tracker Integration
REST
1
mobile apps
Download App
no website account exisiting
Register to Website
Confirm E-Mail Address
website account exisiting
Login to App
all standardized questionnaires completed
all standardized questionnaires completed
standardized questionnaires remaining
Fill out Standardized Questionnaires
3
website
Register to Website
Confirm E-Mail Address
Login to Website
3
Fill out Standardized Questionnaires
standardized questionnaires remaining
2
4
Fill out Assessment Questionnaire
5
Change Notification Schema
Use Main Menu
View Background Information of Project
Change Notification Ring Tone
View Results of Assessment Questionnaire

Figure 2.1.: An overview of the TYT platform (image from [89]), with apps compatible with 3 mobile operating systems, the website, backend, registration process (top), and the user interaction flowchart (bottom)

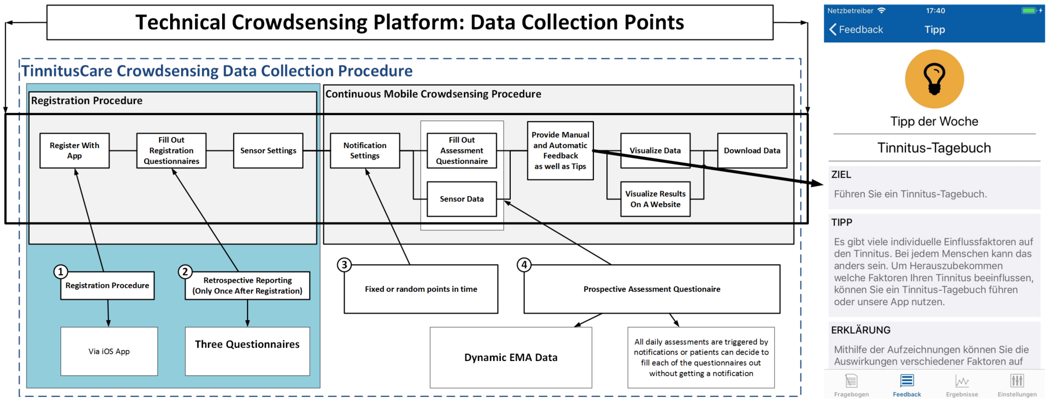
Question	Type
Do you perceive the tinnitus right now?	Binary (Yes/No)
How loud is your tinnitus right now?	Slider (0–100)
How distressed are you by your tinnitus right now?	Slider (0–100)
How well do you hear right now?	Slider (0–100)
How much are you limited by your hearing right now?	Slider (0–100)
How stressed do you feel right now?	Slider (0–100)
How exhausted do you feel right now?	Slider (0–100)
Are you wearing a hearing aid right now?	Binary (Yes/No)

Table 2.3.: The EMA questionnaire used in the TinnitusTips app.

severity at registration time, followed by EMAs collected through notifications. The static data collected are the Mini-TQ questionnaire [35], the Tinnitus Sample Case History Questionnaire (TSCHQ) questionnaire [54], and the worst symptom questionnaire (1 question that asks about the patient's worst tinnitus-associated symptom). The EMA questionnaire is shown in Table 2.3.

The psychoeducation module of TinnitusTips is in the form of "tips" that are presented to the user after submitting an EMA. The tips were structured so that they defined a goal (falling asleep easier, for example), a specific tip on how to achieve that goal (listen to music), and additional information on why / how the tip helps to achieve the goal (because music will mask the tinnitus sounds).

An overview of the TinnitusTips app is shown in Figure 2.2 (image reproduced from [127]). It was also seen in the study that the psychoeducation module may improve user adherence, with short-term changes to tinnitus severity measured according to the Tinnitus Handicap Inventory (THI) questionnaire (which were however not found to persist).


Technical Crowdsensing Platform: Data Collection Points
TinnitusCare Crowdsensing Data Collection Procedure
Registration Procedure
Register With App
Fill Out Registration Questionnaires
Sensor Settings
Continuous Mobile Crowdsensing Procedure
Notification Settings
Fill Out Assessment Questionnaire
Sensor Data
Provide Manual and Automatic Feedback as well as Tips
Visualize Data
Download Data
Visualize Results On A Website
1
Registration Procedure
2
Retrospective Reporting (Only Once After Registration)
Via iOS App
Three Questionnaires
3
Fixed or random points in time
4
Prospective Assessment Questionaire
Dynamic EMA Data
All daily assessments are triggered by notifications or patients can decide to fill each of the questionnaires out without getting a notification
Netzbetreiber
17:40
Feedback
Tipp
Fragebogen
Feedback
Ergebnisse
Einstellungen

Tipp der Woche

Tinnitus-Tagebuch

ZIEL

Führen Sie ein Tinnitus-Tagebuch.

TIPP

Es gibt viele individuelle Einflussfaktoren auf den Tinnitus. Bei jedem Menschen kann das anders sein. Um Herauszubekommen welche Faktoren Ihren Tinnitus beeinflussen, können Sie ein Tinnitus-Tagebuch führen oder unsere App nutzen.

ERKLÄRUNG

Mithilfe der Aufzeichnungen können Sie die Auswirkungen verschiedener Faktoren auf

Figure 2.2.: An overview of the TinnitusTips platform (image from [127]), with registration, crowdsensing, and psychoeducation modules.

UNITI mobile

The UNITI Mobile mHealth application was developed as part of the Unification of Treatments and Interventions for Tinnitus Patients (UNITI) Randomised Controlled Trial (RCT) [109] , a large multi-center randomised controlled trial that aims to investigate whether combinations of common tinnitus therapies can be more effective than individual therapies. Some therapies are delivered in a hospital by a physician, while the UNITI mobile app is designed partly as an EMA accompaniment to the trial, as well as an option to deliver Ecological Momentary Interventions (EMIs) [131]. Two types of EMIs were added, one that delivers momentary interventions with sound stimulation, and another that contains a psychoeducation module. The full scope of the project is complex, and this work focuses on only the EMA data generated by the UNITI Mobile app, which is the component shown in the bottom-left of Figure 2.3. This component of the app is accessible to all users of the platform, while the EMI components are only available to users whose RCT randomisation allows them to be exposed to that component. However, all treatments, whether delivered via EMI or in a clinic, may directly or indirectly affect the patients' individual perception of tinnitus, and therefore, their EMAs as well.

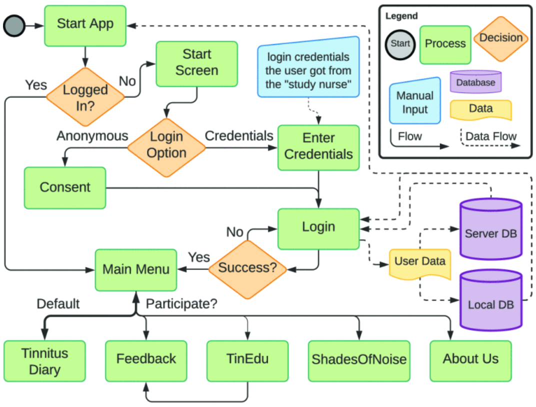


Figure 2.3.: An overview of the UNITI platform (image from [131]), with a flow chart of the user path and data flows.

The EMA component of the UNITI mobile app is the "Tinnitus Diary", as shown in the bottom left of 2.3. This component is enabled for all patients who are participating in the UNITI RCT, and also users who download the app independently from the Apple and Android app stores. The EMA questionnaire consists of 11 questions: 5 that are about the current moment, 5 that are about the daily experience of tinnitus, and one free-text column where the user may make addition notes.

The questions in the EMA questionnaire are shown in Table 2.4. The second question is the EMA for the tinnitus distress for reasons explained in Section 2.1.1 (tinnitus distress is what treatments try to improve, not symptom severity).

Question	**Type**
How loud is your tinnitus at the moment?	Slider 0 - 100 (inaudible – very loud)
How burdensome do you find your tinnitus at the moment?	Slider 0 - 100 (not burdensome – very burdensome)
How tense does your jaw feel right now?	Slider 0 - 100 (not tense at all - very tense)
How tense does your neck feel right now?	Slider 0 - 100 (not tense at all - very tense)
How often have you thought about tinnitus today?	Slider 0 - 100 (not at all - the whole day)
To what extent did you feel affected by your tinnitus today?	Slider 0 - 100 (not at all - very much)
What was your maximum tinnitus volume today?	Slider 0 - 100 (inaudible - very loud)
How much did you move today?	Slider 0 - 100 (not at all - very much)
How stressed did you feel today?	Slider 0 - 100 (not at all - very stressed)
What emotion would you use to describe today?	Slider 0 - 100 (frown - smile)

Table 2.4.: The EMA questionnaire used in the UNITI mHealth app (TinnitusDiary compoenent)

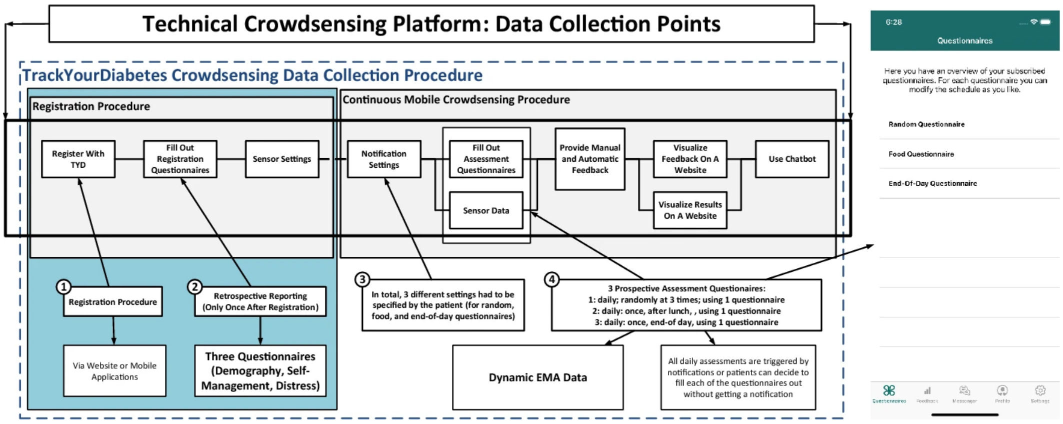
TrackYourDiabetes

The TrackYourDiabetes application was developed as an extension to existing mobile crowdsensing frameworks [87, 52], as part of a pilot study in empowering diabetes patients in Spain and Bulgaria. The framework is broadly similar to what is shown in Figure 2.2, with the important differences beting that there are three momentary assessments questionnaires - the food, random, and end-of-day questionnaires. The food questionnaire was to be filled after every meal (including a photograph of what was eaten), the random questionnaire was presented to the user randomly like in the tinnitus EMA case, and the end-of-day questionnaire was the only questionnaire that was mandatory to be filled out, like the name suggests, at the end of the day. This questionnaire asks the user about their daily heabits, food intake, and most importantly, the degree to which they feel in control of their diabetes. The questions are shown in Table 2.5. In our work, we "use the feeling of control" variable as the

Question	Answer options / Data type
How often do you have measured your sugar level today?	Numeric
For how many minutes have you performed physical activity or sports today?	Numeric
How many bread units have you eaten today?	Numeric
Did you have signs of hyper- or hypo- glycemia today?	No / Don't know / Hyperglycemia / Hypoglycemia / Both
Did you feel in control of your diabetes today?	Slider (0-100)

Table 2.5.: The end-of-day questionnaire in the TrackYourDiabetes App

target variable to stay close to the other mHealth applications' prediction problems, and because it was the variable of interest in the pilot study.


Technical Crowdsensing Platform: Data Collection Points
TrackYourDiabetes Crowdsensing Data Collection Procedure
Registration Procedure
Register With TYD
Fill Out Registration Questionnaires
Sensor Settings
Continuous Mobile Crowdsensing Procedure
Notification Settings
Fill Out Assessment Questionnaires
Sensor Data
Provide Manual and Automatic Feedback
Visualize Feedback On A Website
Visualize Results On A Website
Use Chatbot
1
Registration Procedure
Via Website or Mobile Applications
2
Retrospective Reporting (Only Once After Registration)
Three Questionnaires (Demography, Self-Management, Distress)
3
In total, 3 different settings had to be specified by the patient (for random, food, and end-of-day questionnaires)
4
3 Prospective Assessment Questionaires:
1: daily; randomly at 3 times; using 1 questionnaire
2: daily: once, after lunch, , using 1 questionnaire
3: daily: once, end-of day, using 1 questionnaire
Dynamic EMA Data
All daily assessments are triggered by notifications or patients can decide to fill each of the questionnaires out without getting a notification
6:28
Questionnaires
Random Questionnaire
Food Questionnaire
End-Of-Day Questionnaire
Questionnaires
Feedback
Messenger
Profile
Settings

Here you have an overview of your subscribed questionnaires. For each questionnaire you can modify the schedule as you like.

Figure 2.4.: An overview of the TrackYourDiabetes platform (image from [126]), with a flow chart of the user path and data flows.

2.3. Basic terms and definitions

This section gives a quick overview of some terms that are frequently used in the subsequent sections of this work. Each section introduces them in the context of the datasets, but the abstract overview provided here should help the reader understand the motivation a little better.

Entity

All datasets considered in this work contain multiple observations from what we call 'entities'. Intuitively the entities can be thought of people, the companies whose stock market data are tracked, the sensors that are observed, the products that are being reviewed online, etc. In the subsequent chapters, the word 'entity' may be used interchangeably with 'users' or 'products', depending on the dataset. In our work, we work primarily under the assumption that not all entities contain equal amounts of data, and that most entities might lack sufficient data to train personalised predictive models. This has the consequence that a personalised model for an entity needs to be augmented with the data of other entities. Discovering the best set of entities to learn a personalised entity-level model is the central goal of this work.

More formally, the dataset $\mathcal{D}$ is assumed to be made up of a set of entities $E = \{e_1, \ldots e_{\mathcal{N}}\}$, where $|E| = \mathcal{N}$ and each entity e_i is observed over time.

The term 'personalised model' or 'entity-level' model for entity e_i refers to a model ξ_n which is used only to predict the data of entity e_i. The output of the personalised modelling framework is a set of models Ξ trained on the data of all entities in E, such that $\Xi = \{x_1 \ldots x_{\mathcal{N}}\}$.

'Static' and 'Dynamic' data Each entity e_i is described by two vectors D_i and S_i, which describe the properties of the entity that do and do not vary over time. S_i is a vector of dimensionality E_s, and the size of this vector $|S_i| = E_s$ is fixed for all entities. $D_i = \{o_1 \ldots o_{T_i}\}$ is a sequence of timestamped observations generated by the entity e_i, and $timestamp(o_x) < timestamp(o_y) \forall x < y$. The number of observations from entity $|D_i| = t_i$, and this number may be different for each entity. Each observation is a d-dimensional vector $o_i = \{x_1 \ldots x_d\}$.

Intuitively, the set of entities E can be thought of as a set of users / patients / products, etc. in a dataset, where each entity e_i is described in two data spaces, the 'static' and 'dynamic'. In case of a patient, the 'static' data are properties of the patient that do not change over time (for example, genetic factors, family history, gender, etc.), and the 'dynamic' data captures properties that are time-changing (blood pressure, blood sugar levels, tinnitus distress, etc.) and are more often of prediction interest.

Figure 2.5 shows a high level overview of how the set of entities E in the dataset $\mathcal{D}$ contains entities e_i that can be described in both the static domain S_i and the dynamic domain D_i.

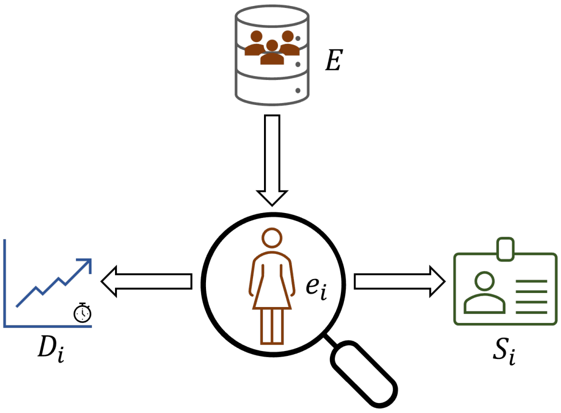


Figure 2.5.: An overview of the entities, with the dynamic (left) and static (right) properties describing each entity e_i in the set of all entities E.

3. Towards RQ1: Personalised predictors using neighbourhoods on static data

This chapter is based on the outputs from the following papers:

[123] Unnikrishnan, V. et al. “Entity-Level Stream Classification: Exploiting Entity Similarity to Label the Future Observations Referring to an Entity”. In: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA). 2018, pp. 246–255.

[125] Unnikrishnan, Vishnu et al. “Entity-level stream classification: exploiting entity similarity to label the future observations referring to an entity”. In: International Journal of Data Science and Analytics 9.1 (2020), pp. 1–15.

This chapter introduces the first of three main conceptual parts investigated as the three main research questions introduced in Section 1.2, namely:

RQ1 To what extent can entity-similarity be exploited when training personalised models? To what extent does similarity in static data inform about similarity in dynamic data?

In order to tackle this research question, it is necessary first to explore how personalised predictors can be trained, given the type of data described in Section 2.3. This breaks down RQ1 into two parts - finding the best entities to learn from, and combining their data effectively.

3.1. Motivation and comparison to related work

As explained in the introduction to this work, the idea of personalised predictors has found increasing attention in recent times, thanks to increases in data availability as well as higher interest in interpretable and explainable models [65]. Of the two approaches toward personalisation described in the book, we follow the approach of ‘model-based personalisation’, where the model is explicitly parametrised with the entity that it is being personalised to.

An early work that highlights the importance of personalisation in the context of recommender systems comes from the industry, where [60] discusses the approach followed by Amazon in developing personalised recommendation models. They liken personalisation for Amazon with a physical store where each customer has their own version with different shelves stocked with different products. Although they found user-based methods intractable because of large dataset sizes and the performance and computation disadvantage of clustering-based methods in grouping users, they achieve personalisation by flipping the problem and performing the expensive offline computations on the item-item level instead. Throughout the work, there are multiple references to the fact that data that is intractably large becomes more manageable when focusing on reasonable subsets - for example, all items bought by a user is often far smaller than all items in the dataset. The work in [60] shows that even for very early work in personalisation, it was seen that relatively simple methods applied locally on well represented knowledge can achieve competitive, and more importantly for the industry, scalable performance. The authors have subsequently expanded on their work in [117], where they explore refinements to handle the fact that Amazon sells more than only books, as was the case in 2003. The work highlights the importance of learning from the correct data and the fact that methods that use all users' data indiscriminately tend to get dominated by a few heavily active users. The similarity measures they use, however, while acknowledged to be important, are unfortunately tailored to the properties expected in an e-commerce dataset. Another important point they highlight is the role of time in datasets where different users join, interact with, and leave the systems at different points in time. User preferences are also learnable to different levels from interactions over different time ranges (for example, a book says more about the person for a longer time frame than clothes). Further approaches have also investigated personalisation in the context of recommender systems by explicitly modelling user behaviour as a linear model of weighted past interactions[76], and as an averaged latent factor model of item representations [46] respectively. While they are instructive in the fact that similarity is complex, temporal, and difficult to capture for all users, the training data (and more importantly, the size of the training data) prohibit using these works as anything but inspiration.

Closer to our work is the data of [138], which works with web log sequences that have fixed beginning and end conditions. The data are logs from the game "Wikispeedia", where users race to find the shortest path between a source and destination article, using only in-page Wikipedia links. [138] proposes methods that deal with several issues that are very relevant to our problem, like:

- capturing different levels of change
- dealing with each entity having its own clock
- the need to explicitly model whether all entities evolve according to the global clock, or whether each entity is given their own start time
- whether the entities will follow the same path, or progress through the same stages, and
- if yes, whether they will do so at the same rate

They also show that the personalised modelling approach can fit to data from very different domains - for example, they include a medical event sequence dataset as well as web log data (Wikispeedia). In their contextual personalisation approach, the authors investigate time-evolving event sequences from multiple domains to extract the evolution of temporal patterns in event sequences. They propose a dynamic programming framework that discovers the 'states' that the sequences travel through. The states are analogous to Hidden Markov Model (HMM) states, but the proposed solution constrains the maximum likelihood estimate by limiting the permitted transitions between the states - once a state S_i has been reached, subsequent observations from the sequence are limited to belong to state S_i or higher (i.e., for all subsequent states S_j, $j \geq i$). While this works for some medical datasets (patients are unlikely to reach Stage-III cancer after they reach Stage-IV), it is not our case since diseases like tinnitus have no known 'states' that a patient progresses through, and the states are not ordered to exclude some states once another has been reached.

The idea of personalisation has also been studied by Ng, et al. in [73], where the authors refer to the problem of predicting for multiple entities as "heteroscedasticity". While the keyword does capture the central point that variances across individual surgical procedures are different, we would like to call out that it does not sufficiently distinguish against the case of the variance within the same entity/patient changing over time. Ng, et al. investigate the case of estimating the surgery duration, where they propose a way to overcome the drawback of neural networks in dealing with heteroscedasticity by explicitly parametrising a gamma distribution and a Laplace distribution. The use of the gamma distribution is motivated by the fact that its long tail is restricted to positive values, and the Laplace distribution was found to better fit the domain data.

Another study that is thematically closer is [74]. Ni, et. al. investigate the case of making personalised recommendations for fitness tracker data, which is closer to the mHeath scenario, although it is important to note that the availability of data collected using a sensor is less noisy and more regular than EMAs where a patient needs to report their subjective experience by filling out a questionnaire. This is also reflected in the size of their dataset, which has >250,000 workouts, but the authors also call out the fact that the amount of data available per user is low. Another similarity is the use of the keywords 'static' and 'temporal' features, although the authors use the keyword static feature as the exercise profile that the same user has exhibited in his past workouts (i.e., the temporal profile one would expect given a user and workout type, given that same user's workout history.), along with some user-specific attributes. The authors find that an LSTM augmented with static and temporal features is able to better predict personalised recommendations to the user based on forecasting the short-term temporal data ("take a 5 minute break to keep your heart rate in the optimal range").

Another approach to personalisation is the two-tier architecture followed in [6], which is evaluated in the context of financial time series. In the first step, they identify the other most relevant time series to learn from, and in the next, exploit the information from those time series using a multivariate k-Nearest Neighbours (kNN) regression. The discovery of the most relevant time series that can be exploited to create predictions for the current time series is discovered through a correlation-based analysis of the historical data of the past 9 years. The highest correlated stocks from other companies are chosen by the kNN, and the predictions of the subsequent kNN is trained on an interval-based transformation of the stock price. They show although not all stocks are better predicted by the neighbourhood-augmented kNN, the augmentation process improves the overall accuracy by 50-55% over a custom error measure that computes the RMSE over N consecutive days (improvement is $\sim$ 30% for MAPE). This workflow can work when the input time series are aligned and when they are observed regularly over time, but these prerequisites are not met by the datasets we aim to study in this work. The authors also exploit the additional information about the S&P500 sector by limiting the kNN step to searching only among other time series belonging to the same sector. This information might not always be available.

Numerous approaches attempt a similar approach to [6] to forecast electricity pricing [122, 121, 62, 90]. kNN predictors are used to forecast electricity price development by chopping the electricity demand data into 24-hour chunks, and computing the similarity of the current day to the most similar historical data. Improvements have been made by manually handling exceptional days like holidays with dummy variables, including temperature, cloud cover, etc.

The term 'subpopulation model' has also been used to describe personalisation. Liu, et al. investigate the training of 'instance-specific' subpopulation models [61]. They acknowledge several of the difficulties in training patient-specific models, like the difficulty of defining similarity based on 'atemporal patient data', and the difficulty of relying on sequence/subsequence similarity measures when some patients may be too short. Additionally, they also want to avoid the problem that a subpopulation gets so large that patient-specific variability is lost. Their proposed solution, therefore, trains a global model on all available data and then trains an additional patient-specific model that is trained on the residual of the patient data as predicted by the global model. It is expected that the 'adaptation model' trained on the residual captures the patient-variability as defined against the whole population. Each of the models is trained separately and the lowest error model is switched in / the predictions are weighted based on the errors that each model makes for each patient.

Previous work in our group [10] has already investigated entity-level models (what we call 'personalised', in this work), in the context of textual review data in the "Tools and Home Improvement" category of the Amazon dataset. The degree to which entity-centric models can predict future arriving labels based purely on the timestamp and the product ID was investigated using a variety of models. It was found that even very simplistic entity-specific models like those that predict the next rating as a simple moving average over the past ratings can achieve an error of 1.3 units, while global models trained on text data from all products were only 11.4% better. Quite apart from the fact that the predictors can be 11% worse without reading the text, the simple moving average models took only 0.91% of the time to execute compared to the global multinomial naive Bayes' (0m:7s vs. 12m:48s). This work serves as inspiration for our methods since it suggests that information from entities and their neighbourhood can capture local trajectories, while global models might fail because the large number of local trajectories overwhelm each other.

3.2. Personalised models on static data, and incorporating expert knowledge into them

The following two sections describe our work in developing personalised predictors for a classification scenario. Although the target attribute is numeric, we assume that the data collection process involves collecting the full dynamic feature space for an entity until time t, after which the target variable is no longer available. Intuitively, this can be thought of as the case where patient data is collected regarding lifestyle, diet, etc. while wearing a glucose monitor for a week, but you want to train a model from that data to be able to predict the blood glucose level even after the glucose monitor is no longer being used. It is expected that estimating the current value of the target given the current values of the exogenous variables is an easier problem than forecasting the exogenous variable at a time $t + \delta$ when none of the variables are known (you don't have the glucose monitor data, but also do not have the data regarding patient lifestyle).

3.2.1. RQ1.1: Augmenting data for personalised predictors

In this work, we tackle the main question of how a personalised predictor can be trained in an environment where the dataset consists of multiple entities, each of which is tracked over time. Additionally, each of the entities is described in a 'static' data space, where relatively unchanging properties of the entity are recorded. To reiterate the overview already presented in Section 1.1 with an example, each entity can be thought of as a patient or a user of the apps presented in Section 2.2. Each entity is therefore described by relatively unchanging properties (like age, gender, and severity of symptoms at registration time), as well as more dynamic properties that are of prediction interest. For example, in the case of tinnitus patients, the daily tinnitus distress is of prediction interest to physicians because the everyday/dynamic presentation of diseases like tinnitus is not fully understood. Although our proposed methods will work for many datasets that have panel-data-like properties, we consider the feature that our methods produce not just a personalised model that is potentially more accurate, but also the fact that each entity also gets a list of 'neighbours' as part of the useful output. This additional output is expected to help better understand the model, and to be particularly relevant in the case of medical datasets since it can enable a physician to perform additional analyses on the properties shared by those users that are in each others' neighbourhoods.

In many cases, computing the similarity over the dynamic data is sufficient to discover the best subset of data to learn a personalised model. However, this does not work for all datasets, because datasets like mHealth data, and other datasets that reflect human behaviour (esp. when the behaviour is the voluntary submission of data, like reviewing a product) have some properties that make computing a similarity over dynamic data impossible. The main reasons are that the data is irregularly sampled at the entity level, the submitted data may contain missing values that complicate the similarity computation, and also the fact that the 'short' entities are orders of magnitude shorter than the 'long' entities (the shortest users may have just one day of data, while the longest will have thousands). Although distance functions like Dynamic Time Warping (DTW) would be able to compute a distance between very short sequences and those that are much longer, we have already seen that many works recommend summarising such sequences in order to handle the difference in lengths [138].

Since the dynamic data cannot be used to compute the neighbourhood of an entity, our proposed approach aims to discover the neighbourhood using the static data S_i. Once this neighbourhood defined over the static data has been computed, the dynamic data of those users can be used to train the personalised model. Several options for combining the dynamic data of the various neighbours are investigated: The data can either be pooled to train a single model, or the personalised model can be treated as an ensemble of entity-level models, where the predictions of each neighbour's model (trained on that neighbour's data alone) are aggregated into the final prediction. Since each entity has its own start time, the effect of preserving the timestamps v/s not preserving the timestamp needs to be investigated.

Formalisation

As has been discussed in Section 2.3, this work assumes a set of entities E in the dataset, where for each $e_i \in E$, we have data in two modalities - static data S_i and dynamic data D_i.

The static data is a dimensional vector $S_i = s_i^1 \dots s_i^{|S_i|}$, and the dynamic data consists of a sequence of timestamped observations $D_i = \{o_i^1 \dots o_i^{T_i}\}$. Each entity e_i has an 'entity length' equal to the number of observations in the dynamic data D_i, which in the above case would be $|D_i| = T_i$. Each observation in the dynamic data of entity e_i has a timestamp associated with its time of arrival in the sequence D_i. i.e., $\forall o_i^n, \in D_i, n = 1 \dots T_i$, the function $timestamp(o_i^n)$ returns the time at which the observation arrived.

As described in Section 3.2, the goal of the supervised learning problem is to learn a model that is able to label an observation o_i^n, arriving n^{th} in a sequence given the past observations until $timestamp(o_i^n)$. This is analogous to a situation where an oracle is available to label a stream until the $n - 1^{th}$ instance. We focus on methods for numerical labels, and focus on predicting the labels of all arriving observations after the cutoff where the labels become unavailable.

Discovering the neighbourhood by 'borrowing' the similarity from static data: The first step towards the training of the personalised model is to discover the neighbourhood for each entity e_i. In our work, this is accomplished through kNN algorithm that operates on the static data of the entity, i.e, the vector S_i. The output of the kNN is an ordered set $kNN(e_i) = kNN(S_i, \mathcal{D}) = n_i^1 \dots n_i^k$, where each $n_i^j \in E, j \neq i$, and $similarity(S_i, S_j) \geq similarity(S_i, S_k), \forall k > j$. In our work, we compute kNN using the Euclidean distance (the inverse of which is used as the similarity).

Training the personalised model: Once the neighbours of an entity have been identified, the next step is to train the personalised model. Since the output of $kNN(e_i)$ is a set of neighbours $n_i^1 \dots n_i^k$, the dynamic data of the entity D_i is combined with the dynamic data of each of the neighbours $D_n, \forall n_i \in kNN(e_i)$.

We investigate two ways to combine the data - augmenting the data of similar entities, and augmenting the models of similar entities to create personalised predictors. Given the fact that entity-level models are limited to the amount of data available per entity, we stick to simple models that are not prone to overfitting given limited data. Therefore, the models are linear regressors trained separately for each entity.

- **Model augmentation:** The model augmentation approach trains a model m_i for each entity $e_i \in E$ trained on the dynamic data D_i. Given that the models are linear regressors, the slope and intercept parameters are averaged to create the final 'augmented model' that has the average of the relevant entity-level tendencies for entities in $kNN(e_i)$. Intuitively, this is analogous to saying "The tendency of the entity-level model is the average of the tendencies of its neighbours". The algorithm is presented in Algorithm 3.1.
- **Data augmentation:** Since it is possible that many entities have too little data to train reliable models, we also follow another approach to train entity-level models. In the data augmentation approach, the dynamic data of entity

Algorithm 3.1 Train an entity-centric model for entity e using model augmentation

```
procedure GET_AVERAGED_MODEL($all\_models$)
    $intercept \leftarrow 0$
    $slope \leftarrow 0$
    for $model \in all\_models$ do
        $intercept \leftarrow intercept + model[intercept]$
        $slope \leftarrow slope + model[slope]$
    $intercept \leftarrow intercept/(k+1)$                ▷ k neighbours + 1 entity model
    $slope \leftarrow slope/(k+1)$
    return $linear\_model(intercept, slope)$  ▷ Create model object with params
procedure GET_NEIGHBOURHOOD($e, D, k$)
    $static\_data \leftarrow e[\text{“}static\_data\text{”}]$
    $neighbours \leftarrow get\_kNN(static\_data, D)$
    return $neighbours$
procedure TRAIN_PERSONALISED_MODEL($e, target$)
    $dynamic\_data \leftarrow e[\text{“}dynamic\_data\text{”}]$
    $entity\_model \leftarrow train\_regressor(dynamic\_data, target)$
    return $entity\_model$
Require: Entity $e$, dataset $D$, target, k (neighbourhood size)
$all\_models \leftarrow dict()$                  ▷ initialise empty dictionary of models
$neighbours \leftarrow GET\_NEIGHBOURHOOD(e, D, k)$
$all\_models[e] \leftarrow TRAIN\_PERSONALISED\_MODEL(e, target)$
for $n \in neighbours$ do
    $nb\_model \leftarrow TRAIN\_PERSONALISED\_MODEL(n, target)$
    $all\_models[n] \leftarrow nb\_model$
$personalised\_entity\_model \leftarrow GET\_AVERAGED\_MODEL(all\_models)$
```

D_i as well as the dynamic data of all entities $D_{n_i}, \forall n_i \in kNN(e_i)$ is combined to create a single personalised model m_i. An algorithm for the procedure is presented in Algorithm 3.2

Algorithm 3.2 Train an entity-centric model for entity e using data augmentation

```
procedure GET_NEIGHBOURHOOD($e, D, k$)
    $static\_data \leftarrow e[“static\_data”]$
    $neighbours \leftarrow get\_kNN(static\_data, D)$
    return $neighbours$
procedure GET_COMBINED_DATA($entity\_set$)
    all_data = list()
    for $entity \in entity\_set$ do
        $entity\_dynamic\_data \leftarrow entity[“dynamic\_data''}]$
        $all\_data.append(entity\_dynamic\_data)$
    return $all\_data$
Require: Entity $e$, dataset $D$, target, k (neighbourhood size)
$all\_models \leftarrow dict()$                    ▷ initialise empty dictionary of models
$neighbours \leftarrow GET\_NEIGHBOURHOOD(e, D, k)$
$combined\_data \leftarrow GET\_COMBINED\_DATA(\{e\} \cup neibhbours)$
$personalised\_models[e] \leftarrow train\_regressor(combined\_data, target)$
```

Dealing with time: $kNN(e_i)$ delivers a list of entities n_i that are similar to e_i, however, each entity in the set $e_k \in e_i \cup kNN(e_i)$ has observations in D_k that have their own timestamps. There are two options for dealing with time. One is to use the timestamps as-is, and allow that some entity-level models are informed by the others whose first observations might be in the future. The other method aligns each entity to its own 'local clock' with $local\_timestamp(o_i^t) = timestamp(o_i^t) - timestamp(o_i^1)$, where the first observation of each entity arrives at time 0, and all other timestamps are calculated relative to it.

Choosing the neighbourhood size: The main motivation of our work in training personalised models is that the training data for the model is chosen to be most relevant for each entity in the dataset. The optimal neighbourhood size is expected to be dataset-dependent and needs to be discovered through hyperparameter tuning. Note that a trade-off exists that very small neighbourhoods are likely to yield too little data for training reliable models and that as the neighbourhood size k increases, the model gets less and less personalised. It is expected that a dataset where personalisation is beneficial will show an error curve that is U-shaped, with a minimum at the optimal neighbourhood size. It is also important to remember that while optimising for neighbourhood size, it is assumed that the similarity function used by the kNN is relying on relevant information only. The inclusion of misleading or noise features can artificially inflate/deflate similarities and threaten the result. Careful evaluation is therefore necessary, because of the high number of models trained.

3.2.2. RQ1.2: Incorporating expert knowledge into the personalised modeling process

As explained in Section 3.2.1, our personalised modelling process produces two outputs - one is a personalised model that is potentially more accurate, and the other is the neighbourhood of an entity that helped create those more accurate predictions. Assuming that the neighbourhood size is tuned to best fit the dataset, the neighbourhoods can serve as a starting point for further analyses, especially for a physician. For example, various static properties shared by entities that are in each others' neighbourhoods can be studied, and further analyses can also attempt to connect them to the dynamic data that details the presentation of their disease. This is akin to developing a 'dynamic' phenotype of the disease, which can be very useful for diseases with a psychological component because symptoms can vary across patients at very different rates. If certain static properties are identified to be associated with 'unhealthy' patterns in the EMA data, those patients can be prioritised for preventive care. Since the only expert knowledge that we have access to was for the EMA datasets referring to Tinnitus Symptoms, the methods in this section are developed primarily with the EMA datasets in mind. However, the main concept is sufficiently abstract that it can be applied to any dataset where information like pairwise constraints (used in semi-supervised clustering) is available[91].

The idea of incorporating expert knowledge into clustering has already been thoroughly explored by numerous works [91, 110, 132], which discuss the use of must-link and cannot-link constraints to improve clustering. Other ideas of constraining the k-Means clustering by enforcing a minimum size for clusters, and also the maximum distance from cluster centre are presented in [13] and [7]. As explained in the survey [110], the main ideas for using constraints in clustering come in the form of must- and cannot- link constraints, distance constraints, and constraints that combine the two concepts.

The idea of constraining kNN classifiers is also detailed in [99, 142] and numerous other works, with the main themes being either reducing the dimensionality of the kNN space [142] or limiting the neighbourhood search in a way that prunes distant objects, for e.g., by performing an initial clustering and then limiting the kNN search process to objects within the nearest cluster to the query instance[22]. Apart from classification accuracy, the pruning is also motivated by the fact that restricting the search of the lazy kNN classifier to a smaller set of instances comes with performance improvements, especially for large datasets.

We have established above that the kNN algorithm can be constrained so that two entities can be excluded from each others neighbourhoods based on expert knowledge. However, since we apply the kNN on the static data as a way to select the best subset of dynamic data D for training the personalised model, it is still necessary to show that static data properties can reflect the development in the dynamic / EMA data. Fortunately, the idea that EMAs can reflect the disease dynamics is well studied, and its benefits to better understanding and management of psychological illnesses is obvious. In the case of suicidal ideation, it is said in [12] that direct access to the mood of the patient through EMAs eliminates the problem of having only indirect information about the patient's daily life, and the variability of the symptoms everday. Remote monitoring through EMAs adds the further benefit that many patients that are not admitted become accessible thanks to technology. A pilot study found that patients who filled an EMA on suicidal ideation may benefit from timely mHealth interventions [133]. [12] show that a GMM clustering of patients according to the variability in their EMAs yields clusters that align with six clinical domains. A meta-analysis of patient EMA data related to chronic pain also showed [108] that various 'functioning measures' of pain can be estimated from features derived from EMA data, including patient variability.

Symptom variability would be almost impossible to study without EMAs, since it would need hospital admission as well as person power to collect data regarding symptoms multiple times per day. Results that connect assessments typically conducted in a hospital with those that can be measured by EMAs (like subdomains, functioning measures [12, 108]) suggest that expert knowledge about patient presentation in the hospital data can be used to find groups in the EMA data generated from patients who belong together, and exclude those that do not.

Formalisation

Since we do not use the kNN in our work as a classification algorithm, we draw inspiration the must-not link constraints from clustering and apply them so that our neighbourhood search is restricted to return only those entities that are allowed to be in each others' neighbourhoods. This means the kNN does not necessarily return the k closest entities as defined by the static data, but rather the k closest entities as per the static data that are not excluded from the current entity's neighbourhood by expert knowledge. Intuitively speaking, this can be thought of as limiting the neighbourhoods of diabetic patients to type 1 and type 2 patients exclusively, regardless of the degree to which other features are shared. This would mean that two patients who are dissimilar in the diabetes type but identical in every other way would still not be placed within each others' kNN neighbourhoods. The motivation behind this decision is that when expert knowledge exists that can inform the predictor that two entities are not expected to follow the same patterns in the dynamic data, then they are to be excluded from each others' neighbourhoods regardless of the degree to which other properties are shared. Therefore, we propose in this work that we 'prune' the kNN-based subset of neighbours $1 \dots k$ such that a set of do-not-link constraints $C = \{c_{i,j}\}$, where $e_i, e_j \in E$ and $i \neq j$ are not violated. A constraint $c_{i,j}$ implies that two entities e_i and e_j are now allowed to be in each others' neighbourhoods. $pruned\_kNN(e_i) = \{n_1^i \dots n_k^i\}$ where $n_j^i \in E$ and $c_{i,j} \notin C$.

Incorporating expert knowledge about anomalous tinnitus patients into neighbourhood

Our work in [125] is aimed at exploiting the expert knowledge to restrict the neighbourhood discovery process to patients who have clinically consistent presentation. We focus on the case where a medical expert is able to provide information that is similar to a must-not-link constraint as commonly defined in semi-supervised clustering methods.

The main inspiration for this work is the fact that there is evidence that while tinnitus loudness and distress are correlated for the large part, there is a subset of 'anomalous' tinnitus patients for whom the correlation between loudness and distress is not consistent with the mean population [36]. About a third of the tinnitus patients with very high loudness reported only mild to moderate tinnitus distress, in spite of similar age, gender, or duration of tinnitus. However, some comorbidities are associated with a higher degree of distress, like hyperacusis, hearing loss, vertigo, etc. Patients with lower self-reported depression were found to have lower levels of distress in spite of high loudness. The fact that there is a complex and multifaceted relationship between tinnitus loudness and distress that might result in patients with high distress in spite of low loudness and vice versa is suggestive of the fact that these patients have an incongruent presentation and need separate assessment [36].

We incorporate this expert knowledge into our system by modifying the kNN algorithm so that the set of users $e \in E$ are split into G groups $E = \{E_1 \dots E_G\}$, where $\bigcup\limits_{g=1}^{g=G} E_g = E$, and $E_x \cap E_y = \emptyset$, for all $x, y \in \{1, \dots G\}$, and $x \neq y$.

In order to find the G groups, we use the knowledge that there are patients who are anomalous in their tinnitus loudness and distress. To create the groups, a data driven partition of the tinnitus loudness and distress are created by clustering the self-reported loudness and distress separately. If there are x clusters for loudness and y clusters for distress, the patient can belong to one of $x * y$ clusters - i.e., $E = \{E_1 \dots E_{x*y}\}$

The rest of the framework for building personalised models is essentially unchanged from that presented in 3.2.1, except for the fact that the kNN is now applied separately within groups. The algorithm for training the personalised in-group model is shown in Algorithm 3.3.

3.3. Experiments

We consider 3 datasets from three different domains - one with products receiving reviews in e-commerce, one from the mHealth domain of patients using TYT (see Section 2.2) to answer EMA questionnaires over time, and a third with sensors from a weather station tracked over time. We first exclude all entities that are too short for learning (exact numbers below), and split the first 60% of each remaining entity's data into the training set, and the test into the test.

Air Quality Index - Carbon monoxide daily summary data from the Environmental Protection Agency The AQI (Air Quality Index) dataset is made publicly available by the Environmental Protection Agency, and the dataset used in this study is the yearly carbon monoxide summaries, which contain the daily and the yearly averages of the carbon monoxide levels at multiple weather stations across the US. As the time of this writing, the official dataset web page has changed and only allows for downloading data after filtering for location and sensor type, but a larger version of the dataset can be found on kaggle at `https://www.kaggle.com/datasets/epa/carbon-monoxide/code`[1]. From the entire dataset, we subset the data from 1990 to 2017. Although not part of the modelling process, the data of 1989 is used to create aggregations that serve as 'static features', with the motivation that long-term averages capture properties that are more reflective of long term weather at the location.

In addition to CO levels, the dataset also contains other metrics like PM2.5, PM10, etc., while we use only the daily CO levels and the additional variables 'Air Qual-

[1] Accessed on 14.01.2024

Algorithm 3.3 Train an entity-centric model for entity e using data augmentation

```
procedure GET_GROUPS(E, n_loudness_groups, n_distress_groups)
    loudness_groups = $kMeans(E[“static\_data”][“loudness”], n\_loudness\_clusters)$
    distress_groups = $kMeans(E[“static\_data”][“distress”], n\_distress\_clusters)$
    $E_g$ = dict()        ▷ $E_g$ stores a mapping from groups to entities within
    for $e \in E$ do
        $e_g \leftarrow$ concatenate(loudness_group[e], "_", distress_groups[e])
        groups_dict[e] $\leftarrow e_g$
    return groups_dict
procedure GET_PRUNED_NEIGHBOURHOOD($e, E_g, k$)
    $static\_data \leftarrow e[“static\_data”]$
    $neighbours \leftarrow get\_kNN(static\_data, E_g)$
    return $neighbours$
procedure GET_COMBINED_DATA($entity\_set$)
    all_data = list()
    for $entity \in entity\_set$ do
        $entity\_dynamic\_data \leftarrow entity[“dynamic\_data''$]
        $all\_data.append(entity\_dynamic\_data)$
    return $all\_data$
procedure GET_EXPERT_PRUNED_PERSONALISED_MODEL(Entity e, entity
    group $G_e$, target, k)
    $neighbours \leftarrow GET\_PRUNED\_NEIGHBOURHOOD(e, G_e, k)$
    $combined\_data \leftarrow GET\_COMBINED\_DATA(\{e\} \cup neibhbours)$
    $personalised\_model \leftarrow train\_regressor(combined\_data, target)$
    return $personalised\_model$
Require: Set of entities $E$, target, k, n_loudness_groups, n_distress_groups
$personalised\_models \leftarrow dict()$        ▷ initialise empty dictionary of models
$E_g \leftarrow GET\_GROUPS(E, n\_loudness\_groups, n\_distress\_groups)$
for $e \in E$ do
    $e\_model \leftarrow GET\_EXPERT\_PRUNED\_PERSONALISED\_MODEL(e, E_g[e], k)$
    $personalised\_models[e] \leftarrow e\_model$
```

ity Index' and 'Max_Observed_Daily_CO_Value' as independent variables that predict the target. The dataset consists of 200 entities (weather stations) with 577482 observations for a period starting 1990 and ending 2017. The target variable 'mean_CO_value' is measured in parts per million (ppm) as a daily average, and has the following properties: $mean = 0.742, min = 0.004, max = 8.461$, and $standard\_deviation = 0.519$.

Attribute space for static features S: The most obvious choice for the 'static' properties of the weather stations are their location available under the latitude and longitude. However, little other information was available that is 'fixed'. In order to extract more static properties, we focus instead of extracting features about the weather station that reflect the long-term trends - towards this purpose, we use the yearly average CO, the standard deviation in CO measurements, the 1st and 2nd max values encountered, as well as the 50th percentile and the 90th percentile of CO measurements. These yearly aggregations are all performed on the data from 1989 (i.e., one year prior to the start of the dynamic data), to prevent data leakage regarding the target in the first year of the dataset.

Attribute space for the dynamic data D: For both the daily and yearly AQI datasets, we removed the observations related to anything other than the carbon monoxide levels (SO2, NO, NO2, and several other pollutants), since not all weather stations were equipped with sensors for each pollutant in the dataset. For the CO values, it was seen that measurements sometimes existed as averages computed over several time periods (2 hours, 4 hours, 8 hours, etc.). In each case, we kept the daily CO averages computed over the longest time periods. The dynamic attribute space, therefore, consists of 3 variables, namely the timestamp, the AQI, and the max_CO_value.

Figure 3.1 shows the number of entities (vertical axis) for a given length in the dynamic data. While there is an order of magnitude difference in the lengths of the short to the lengths of the longest entities, the AQI dataset has the lowest skew in the number of long entities v/s short ones.

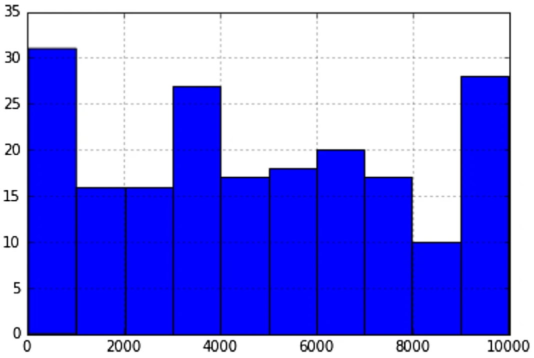


Figure 3.1.: AQI Dataset: #Entities (Y-Axis) versus the length of an entity (X-Axis)

mHealth Dataset: Track Your Tinnitus EMA data The mHealth data used in this study comes from data collected by the Track Your Tinnitus mobile application. As explained in Section 2.2, the users who download and register the app first answer some registration questionnaires (which constitute their 'static' data), and then answer the EMA questionnaires when prompted by the application at randomly set intervals according to user preferences. The EMA questionnaire has 8 variables, of which the question concerning tinnitus distress is used as the target variable (For all questions, please see Table 2.2, Section 2.2).

Attribute space for static features S: The attribute space for kNN computation was done on the registration questionnaires after applying the StandardScaler from scikit-learn [81]. The scaling ensures that questions with different ranges do not affect the distance computation. Since the number of features is large, we also attempted to find the best variables to learn from using a genetic algorithm to find the best features.

Attribute space for dynamic features D: All questions except the target and the last question ("Do you feel unstable at the moment") were used in the dynamic feature space (see Table 2.2). The feature space consisted of 6 EMA variables, which were used to predict the target (tinnitus distress).

All users/entities with less than 5 days of data were removed from the dataset, leaving a total of 516 users. Figure 3.2 shows the number of users (Y-axis) who have a given length (X-axis). It can be seen that the number of users who contribute a certain amount of data reduces strongly with increasing length.

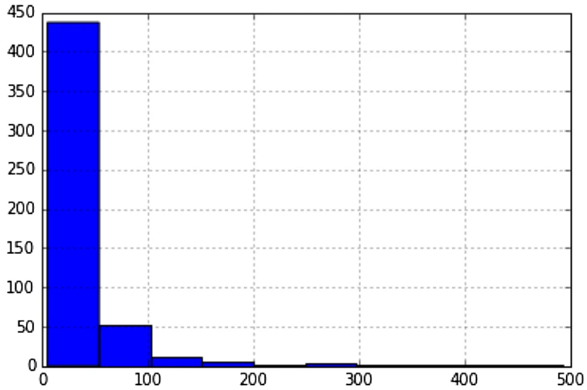


Figure 3.2.: mHealth Dataset: #Entities (Y-Axis) versus the length of an entity (X-Axis)

Amazon: Tools and Home Improvement This dataset is a subset of the dataset introduced in [66], focusing only on the product reviews for the category “Tools & Home Improvement". The full dataset was created by crawling review and Q&A data for 8 categories on Amazon. In this work, only the The Tools & Home Improvement category reviews are used (without the Q&A). The reviews contain a star rating (1-5 stars), review text, and the timestamp for when the review was submitted.

For the Amazon dataset, all products with less than 2 reviews were removed. It can be seen in Figure 3.3 that the distribution of the number of entities (Y-axis) of a given length (X-axis) is much more heavily skewed than in the other two datasets, with the 25th, 50th and 100th percentile of entity lengths being 2, 4 and 4770 respectively. There are 139508 entities (products) in the dataset after filtering, with a mean length per entity of 12.9. It is to be noted, however, that the reviews are not evenly spaced throughout time. Many reviews are known to cluster towards the last timestamps, with also a known trend towards higher star ratings [10].

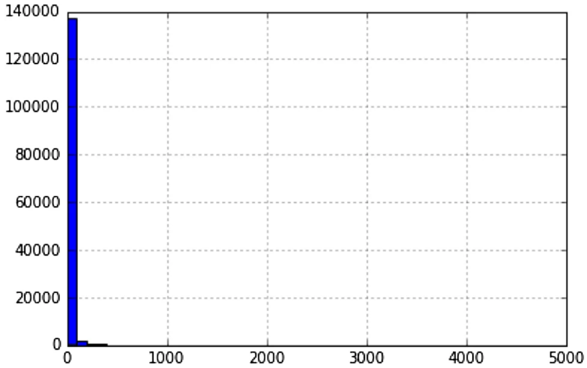


Figure 3.3.: Amazon Dataset: #Entities (Y-Axis) versus the length of an entity (X-Axis)

Attribute space for static features S: Since all reviews are technically in the ‘dynamic’ space, the only possible solution to deriving static data would be to have product descriptions. However, since the dataset contained no product information, we use a word embedding model on the text data with the aim to capture 'products that are described similarly'. More concretely, we build a paragraph vector model as described in [56]. The paragraph vector is very well suited to our problem because it creates fixed-length representations of the paragraphs in the embedded space from the variable-length sentences. We adapt this method by setting the product as the paragraph tag, and the concatenated reviews received by the product as the sentences. Since the paragraph and the text are both embedded into the same space, the sparsity in the number of reviews some products received does not affect the quality of the entity (product) embedding. Used this way, the paragraph vector presents a better performing alternative to bag-of-words models, while the idea of using the product IDs themselves as the paragraph token allows us to capitalise on learned representations like word2vec. The kNN of the paragraph token are therefore used as the way to retrieve similar entities.

Attribute Space for dynamic features D: We follow an approach similar to [10], where we predict the star rating of the review given nothing other than the timestamp. While this is unrealistic, the reader is encouraged to remember that the goal of this work is in assessing how much predictability is lent to the model from knowing the entity.

Is it the neighbour, or just the increased data?: A kRE baseline The goal of our method is to study the predictive power gained through exploiting entity similarity. However, the model is tricky to evaluate, because the addition of an entity in the neighbourhood (especially when using data augmentation) adds not just information regarding the entity, but also increases the amount of data available for learning. This means that additional experiments are necessary to know whether the neighbourhood or the additional data is what helps improve the model. Towards this end, we propose a k-Random Entities (kRE) baseline.

The kRE baseline is designed to test the degree to which the performance of the model improves (or, to be more accurate, changes) because a particular entity was chosen as a neighbour. If the entity that was chosen was indeed the best entity to learn from, then the error metrics would improve to a greater degree than if another entity was chosen. Since the amount of training data available increases along with the number of entities in the neighbourhood, the kRE baseline chooses the same number of entities as the kNN model but chooses the entities randomly without replacement. Running the kRE multiple times and averaging the performance would give a rough idea about how much information is added to the model *without considering entity similarity.* If choosing similar entities is indeed beneficial, then the performance of a kNN model should exceed that of the kRE model given a fixed k.

3.4. Results and Discussion

This section presents the results for the experiments listed above. We start with the results towards training a personalised predictor [123], where we investigate data vs. model augmentation, local entity clocks vs. global clocks, and performance against our proposed kRE baseline. Finally, we look at the change in predictor performance as we move temporally away from the training data. We close the section with the results from [125], where we investigate the degree to which the neighbourhood discovery process can be informed by expert intuition.

3.4.1. Training personalised models:

Data vs. Model augmentation: As described in Section 3.2.1, the data vs. model augmentation experiment compares the personalised model trained on the pooled data from all entities against the quality of predictions resulting from a personalised model that has the averages out the parameters of the individual models that are trained on the data of each of the entity's neighbours separately.

Figures 3.4, 3.5, and 3.6 show the results for the model-augmented and data-augmented regressors for the Amazon, AQI, and mHealth datasets respectively. The charts show the average error of all the entity-level models at predicting the holdout data of the entity.

For the Amazon and the AQI datasets, it can be seen that the model augmentation performs clearly worse (the green lines are consistently above the blue), while for the AQI dataset, the deleterious effect of the model augmentation seems to be lessened by averaging the model parameters over larger neighbourhoods. It is to be noted, though, that the charts show neighbourhoods of size up to 50, which is 25% of the AQI dataset, and only 0.036% of the Amazon dataset. Larger neighbourhoods were not considered for Amazon partially for runtime reasons, but also because the goal of the experiment is to discover the best way to train personalised predictors, and the mHealth use case is not fully reflected in the properties of the Amazon dataset, where we use only the timestamp of a rating to predict the future observations.

Additionally, the fact that model augmentation produces an upward trending error for small neighbourhood sizes is possible because both the AQI and Amazon data have strong dataset-level tendencies, where all Amazon reviews get biased strongly towards 5 stars at the later timestamps in the data, while the air quality has a general trend towards improvement. It could be that selecting for larger neighbourhoods is informing the model about the future trends when using model augmentation, but averaging entity-level slopes is still too noisy.

Interestingly, the two methods perform quite similarly for the mHealth dataset, where we notice another result that deviates from the AQI and Amazon datasets. It can be seen that for the mHealth case, small neighbourhoods seem to improve the predictions of the personalised model, and an increase in the neighbourhood size affects the predictive performance of the model negatively. This is indeed suggestive that the main idea of personalisation has merits for EMA data. The small difference between data and model-augmentation, unlike in the other two datasets, could be because there is no dataset-level tendency to high or low values over time. However, it cannot yet be claimed that the neighbourhood is correct, because the quality of the neighbourhood is tested against the kRE baseline.

Global vs. local clocks: The global vs. local clocks experiment compares the performance of the personalised model where all observation timestamps were used as-is against the model where the data of each entity is assumed to start at its own '0'. This experiment will show larger differences in data-augmented settings because the different entities in a neighbourhood can have very different start times.

Figures 3.7 and 3.8 show the results for the model augmented and data-augmented regressors for the Amazon, and mHealth datasets respectively. This experiment is not applicable to the AQI dataset, since almost all entities in that dataset are

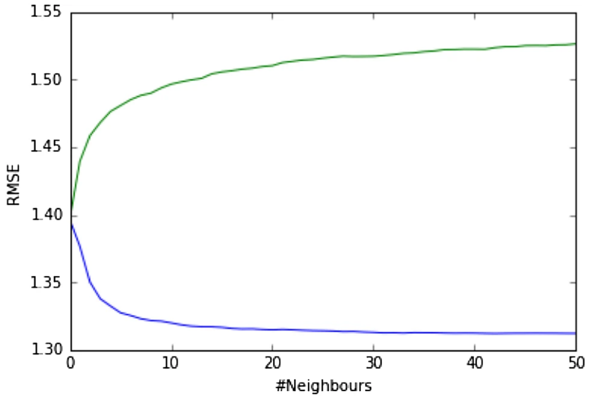


Figure 3.4.: Amazon Dataset: RMSEs (Y-Axis) for models trained with data augmentation (green) v/s model augmentation (blue) for various sizes of the neighbourhood k (X-axis)

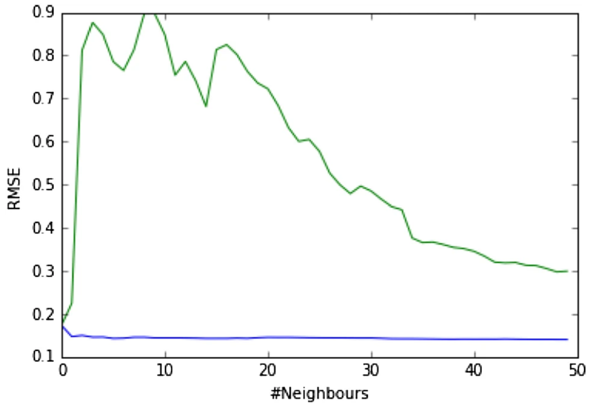


Figure 3.5.: AQI Dataset: RMSEs (Y-Axis) for models trained with data augmentation (green) v/s model augmentation (blue) for various sizes of the neighbourhood k (X-axis)

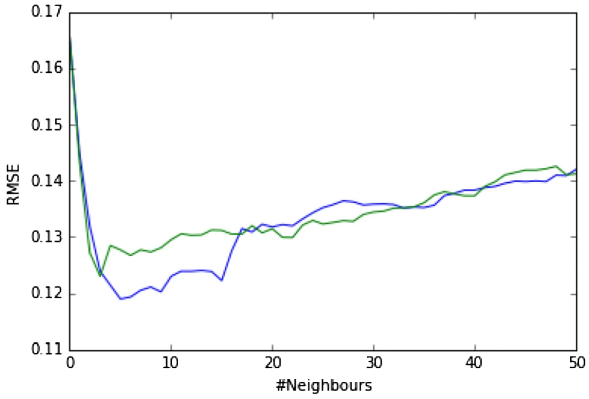


Figure 3.6.: mHealth Dataset: RMSEs (Y-Axis) for models trained with data augmentation (green) v/s model augmentation (blue) for various sizes of the neighbourhood k (X-axis)

already aligned (i.e., they start in 1990). The difference in lengths is rather due to the difference in the last observation date.

The results for the Amazon and mHealth datasets show that aligning the entities to start at 0 has a negative effect on performance. It can be seen that the Amazon dataset is more negatively affected by the performance hit - this is possibly because a greater portion of the entities are very short, slopes are much steeper when the entities are aligned to 0. Short entities that are time aligned at 0 (with observations temporally adjacent) would have steeper slopes than a regressor trained on the same observations temporally farther apart. Using a global clock that is shared by all entities would more likely show this result, especially when more entities tend to 'begin' late in the dataset (i.e., the first observation of most entities is late in the 'global clock').

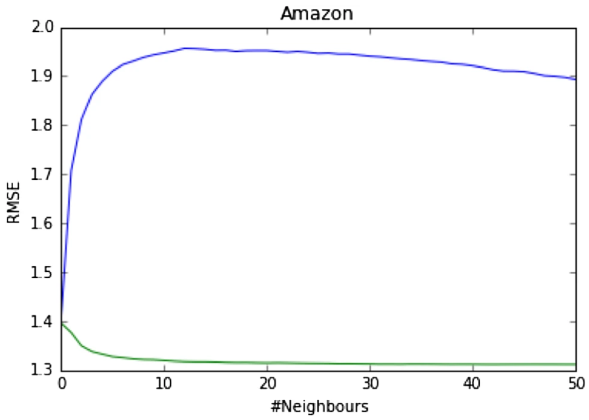


Figure 3.7.: Amazon Dataset: RMSEs (Y-Axis) for personalised models trained using global or local entity-level clocks for increasing sizes of the neighbourhood k (X-axis)

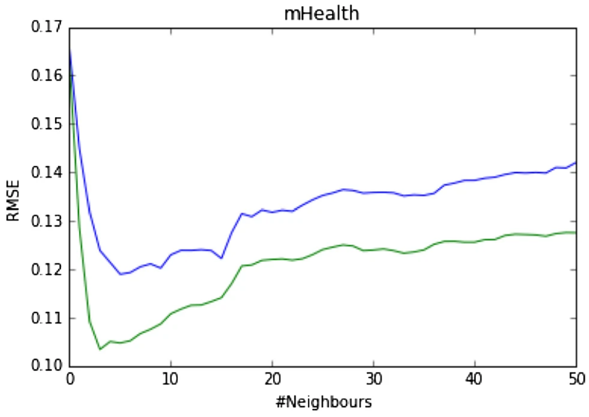


Figure 3.8.: mHealth Dataset: RMSEs (Y-Axis) for personalised models trained using global or local entity-level clocks for increasing sizes of the neighbourhood k (X-axis)

Comparing to the kRE baseline: The baseline model compares the performance of a kNN-based model against that of another model that selects the same number of users, but selects them randomly. If the improved performance is from the chosen

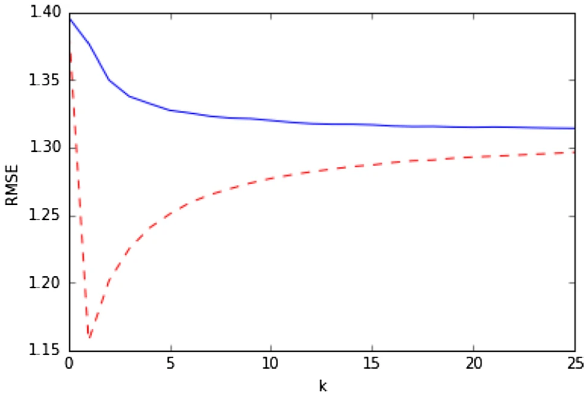


Figure 3.9.: Amazon Dataset: RMSEs (Y-Axis) for kNN (blue) vs. kRE models (dotted red) trained with data augmentation for various sizes of the neighbourhood k (X-axis)

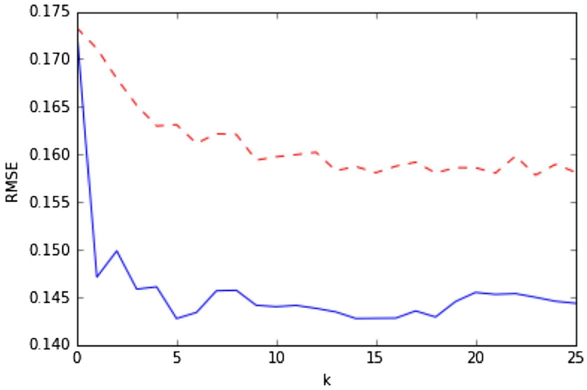


Figure 3.10.: AQI Dataset: RMSEs (Y-Axis) for kNN (blue) vs. kRE models (dotted red) trained with data augmentation for various sizes of the neighbourhood k (X-axis)

neighbours as opposed to simply the increased training data size, the personalised kNN model is expected to outperform the kRE baseline.

The previous results show that the non-time-aligned and data-augmented regressors outperform the other configurations, thus, the best-performing model was tested against the kRE baseline. The errors for the randomly selected entities are done over 30 runs for each neighbourhood size k for the AQI and the mHealth dataset. However, due to the size of the Amazon dataset and the long time required for computation, the results for the kRE are baseline are reported over 5 runs. However, we expect that this does not seriously challenge the validity of the results, since the mean standard deviations for the errors at the entity-level were 0.00046.

Figures 3.9, 3.10, and 3.11 show the results for the performance achieved by the data augmented kNN personalised models vs. the kRE baseline for neighbourhood sizes ranging from $k = 0$ (model trained with data of entity only) to $k = 25$, which includes the data from the 25 nearest neighbours along with the data of the entity for which the personalised model is being trained. Lower numbers on the y-axis indicate models with better performance.

The results on the Amazon dataset are surprising (Figure 3.9), since the error has a

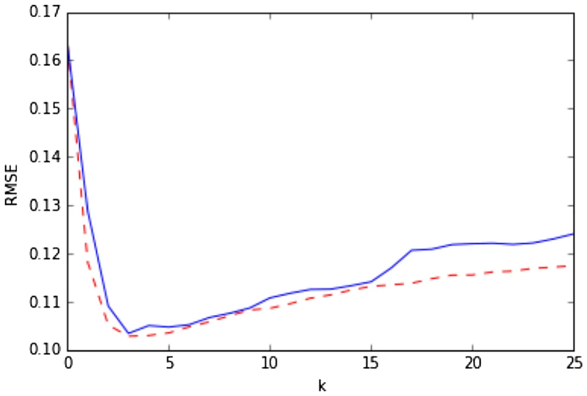


Figure 3.11.: mHealth Dataset: RMSEs (Y-Axis) for kNN (blue) vs. kRE models (dotted red) trained with data augmentation for various sizes of the neighbourhood k (X-axis)

sharp drop from using the entity's data only, to using the data of very few neighbours. However, it is seen that as the number of neighbours increases, there is an increasing trend in the error achieved by the kRE model. This is likely due to the fact that there are a very large number of very short entities (lengths as low as 2), making the kRE more likely to sample a short entity randomly. The addition of these very small entities is possibly introducing too much noise in the system. Although the performance of the kRE model is initially lower (and then trending upwards), the kNN and kRE models are already approaching each other's errors by the $k = 25$. This suggests that the kNN is indeed picking 'good' neighbours, while the kRE isn't, but the fact that the kNN errors are higher also suggests that our method of finding similar entities is not optimal. Exploring the best embedding for the text, however, is not within the scope of our work. Since evaluating embeddings is acknowledged to be difficult [72] and since choosing the best embedding model is not a core component of our workflow, our experiments only assessed the quality of the embeddings by running manual checks on whether the kNN of some entities were all from similar categories. A thorough evaluation of this component would surely be critical if our method is to be used for such a task, however, we leave that for the future, since the main goal of this task was to discover how personalised models can be trained, and to what degree personalised models benefit from concentrating only on data relevant to an entity.

The results for the mHealth dataset (Figure 3.11) are less encouraging. While the data augmented kNN model shows an interesting trend that small neighbourhoods do indeed improve performance, the performance of the kNN model is closely matched by the kRE model, suggesting a problem with the similarity function. The fact that choosing the neighbours 'randomly' vs. based on the static data must mean either that the static data has no role in discovering useful neighbours, or that the feature space does not adequately capture the underlying (possibly non-linear) similarity between patients. However, the results towards of main goal of creating personalised predictors are still encouraging, since it can be seen that small neighbourhoods do indeed help the model achieve better performance, and the RMSE rises again with larger k. This serves as a promising starting point towards investigating better definitions of similarity.

The results for the AQI dataset (Figure 3.10) do indeed show that the personalised models benefit from the kNN over the kRE. It can be seen, however, that the improvements from increasing neighbourhood size flatten as k increases. Since all entities begin at the beginning and the dataset has a general tendency towards better values, it could be that the incremental benefit from adding entities diminishes after a certain limit. Extensions can explore whether more complex models would be able to extract nonlinear patterns from the added neighbours.

Predictions in the near and far future: We take a deeper look at the quality of the predictions from the kNN models by investigating the development in the quality of the predictions as one steps further into the future of the entity (and therefore, temporally more distant than the training data). We do this by comparing the predictive quality of the personalised models for the first N% of the data vs. the last N%. We tested for N=10%, N=20% and N=50% of the test data (which is itself 40% of the total data). Since the trends are similar for all values of N, we report on the 10% case. Predictably, the 50% case shows broadly the same trend but compresses the variability errors between first-N and last-N to be closer to each other.

If the personalised models are indeed able to exploit entity-level dynamics at the observation level, then there will be a systematic bias in the error towards one direction of the mean. If this is the case, it suggests that the kNN models that are learned at the entity level go ‘out of date’, and need to be updated. The degree to which the predictions diverge from the overall mean could reveal the frequency of retraining required to maintain acceptable model quality over time.

If, on the other hand, the entities are ‘stable’ and not experiencing any change, then the models trained on the training data do not need to be adjusted/retrained to incorporate new data, and they can be far into the future. In this case, we would expect that the near and far future predictions would have similar means to the overall prediction error with no systematic tendency to be above/below the means of the data-augmented kNN model.

Figures 3.12, 3.13, and 3.14 show the results for the performance achieved by the data-augmented kNN personalised models for the first 10% v/s the last 10% of the test data relative to the averages already reported. Lower numbers on the y-axis indicate models with better performance. The first 10% of the test data is the 10% of the test data observations that are temporally ‘earliest’.

The Amazon dataset shows a pattern (Figure 3.12) consistent for both the non-medical datasets - that the predictions in the near future have higher errors than the predictions in the far future. While this seems interesting and anomalous, it is probably because of the already reported strong tendency in the dataset towards higher rating values in the future combined with the fact that many entities are concentrated towards the end [10]. As k increases, the difference between the near- and far- future predictions decreases, with both of them approaching the kNN model’s RMSE.

The AQI dataset also shows the same tendency (Figure 3.13) as the Amazon dataset. Unlike the Amazon dataset, though, the near- and far- prediction RMSEs cross at very small neighbourhoods, with the near-future prediction errors increasing and then stabilising with an increasing k. For the far future, however, the errors continue to trend downward. Unlike the Amazon dataset, the gap between the near- and far-future predictions diverge from the error of the data-augmented kNN model. Again,

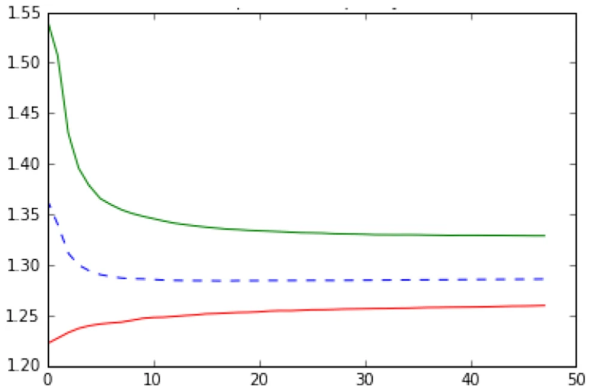


Figure 3.12.: Amazon Dataset: RMSEs (Y-Axis) for kNN (dotted black), with mean RMSE for first 10% of the dataset (green) vs. the last 10% (red) of the entity data. Values are shown for different sizes of neighbourhood k (X-Axis)

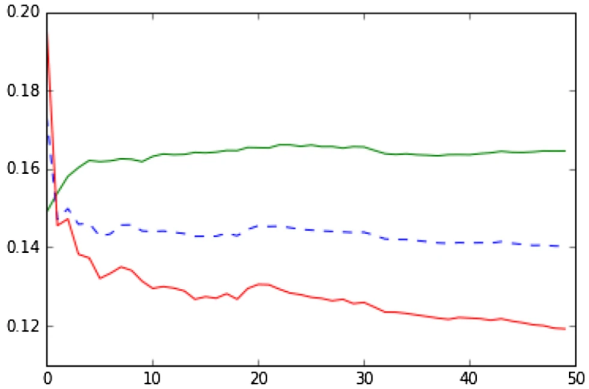


Figure 3.13.: AQI Dataset: RMSEs (Y-Axis) for kNN (dotted black), with mean RMSE for first 10% of the dataset (green) vs. the last 10% (red) of the entity data. Values are shown for different sizes of neighbourhood k (X-Axis)

for the AQI dataset, we suspect that the increased predictability of the observations in the far future is due to the existence of a clear trend in the dataset towards lower CO values that is also reflected in most entities.

In the mHealth dataset, though, we see that predictions in the near future have a lower error than predictions in the far future (Figure 3.14). This suggests that the patients in the dataset show a high degree of variability and that learning their disease patterns at the patient level is beneficial in predicting their near-term development. Since our goal is to model personalised disease dynamics, this is an encouraging result because it combines well with the output of the data-augmented kNN performance.

3.4.2. Incorporating expert knowledge into personalised models:

Extensions to the kNN algorithm have already explored the possibility that not all of the neighbours of an instance will be equally useful, or that some would be misleading [48]. This idea of finding and excluding 'false' neighbours has been used in [63], where the 'neighbouring' days with divergent next-day-demand trajectories are

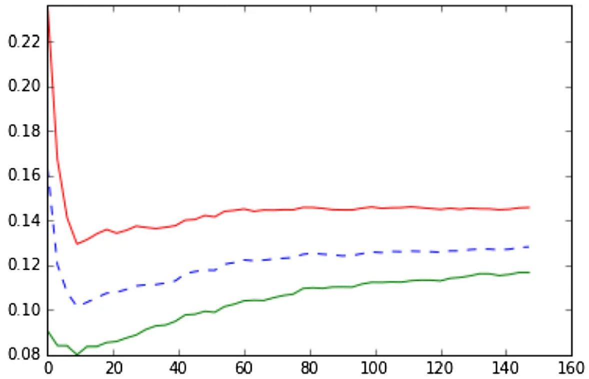


Figure 3.14.: mHealth Dataset: RMSEs (Y-Axis) for kNN (dotted black), with mean RMSE for first 10% of the dataset (green) vs. the last 10% (red) of the entity data. Values are shown for different sizes of neighbourhood k (X-Axis)

removed from each other's neighbourhoods to improve electricity price forecasts. We explore the degree to which the neighbourhood discovery process can be aligned with expert knowledge, and how this information can be integrated into the personalised modelling process. The results in this section are derived from [125].

The expert knowledge available to us for the mHealth dataset is about the 'anomalous' sufferers of tinnitus. The anomalous tinnitus patients have discordant relationships between their tinnitus loudness and distress. Some patients have abnormally high distress even with lower tinnitus loudness, and others have low distress in spite of very high tinnitus volume. We wish to integrate this information into the personalised modelling process using a data-driven partitioning of the users into groups as described below.

Data-driven discovery of anomalous groups in tinnitus: In order to separate the patients into groups of 'concordant' patients, we propose the use of kMeans as an unsupervised data-driven way to discover the cut-offs to segment the data into groups. We run the kMeans with various number of clusters on the tinnitus loudness and tinnitus distress variables collected at the registration time. The number of clusters used is decided by the expert.

Figure 3.15 shows the three clusters discovered by the kMeans algorithm. Grouping the patients by loudness into 3 groups was approved by the expert, with the cut-offs for low loudness being ≈ up to 40/100, and high loudness at ≈ 70+/100. Please note that these are not definitions of low, medium and high loudness of tinnitus, it is simply the cut-offs for splitting patients in our dataset into groups for the purposes of our investigation.

Figure 3.16 shows the partitioning of the users into 2 groups based on their tinnitus distress at registration time. The tinnitus distress is defined here as the score of the TSCHQ questionnaire, which is a sum computed over the questionnaire response and has a score from 0-24 (higher values are worse). The expert preferred 2 clusters here both because of there being no clear pattern in where the scores could be split to low vs. high, and also because fewer clusters would decrease the total number of groups.

Once the clustering process is complete and the groups are created individually

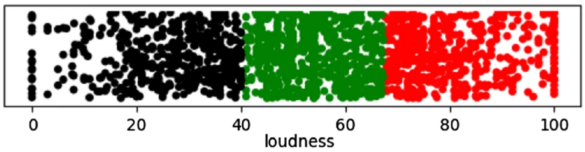


Figure 3.15.: The 3 loudness clusters as discovered by kMeans. The low, medium and high loudness patients are shown in black, green, and red respectively. The number of clusters and the partitioning are verified by an expert. (vertical jitter added to separately show density when multiple patients with the same values)

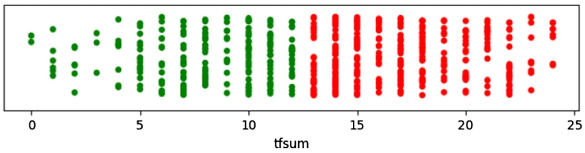


Figure 3.16.: The 2 distress clusters as discovered by kMeans. The number of clusters and the partitioning are verified by an expert (vertical jitter added to separately show density when multiple patients with the same values)

for loudness and distress, each patient belongs to exactly one loudness cluster and one distress cluster. The concordant patient group is all patients who are in the same loudness cluster as well as distress cluster. Since there are 3 clusters based on tinnitus loudness, and 2 based on the degree to which the patient is distressed by tinnitus, each patient can be in one of six groups[2]: {<low-loudness, low-distress>, <medium-loudness, low-distress>, <high-loudness, low-distress>, ... <high-loudness, high-distress>}. The idea is that concordant tinnitus sufferers are those that belong in the same group, and the way this is operationalised is to limit the kNN to in-group neighbours only, while building the data-augmented kNN regressor.

The average distress and loudness among the participants in each group are shown in Table A.1. It can also be seen from Figures 3.15 and 3.16 that the clusters are not well separated - but since the goal of the clustering process was only to find data-driven cut-offs, our expert validation is considered sufficient. Please note that some vertical jitter is added to the plot so that overlapping users with the same loudness or distress are more clearly visible.

The discordant tinnitus patients are those that are in the <low-loudness, high-distress> group and those in the <high-loudness, low-distress> group. The smallest group we discovered has only 35 patients and is one of the discordant groups with low tinnitus distress in spite of high tinnitus loudness. While medical studies have found up to a third of patients are discordant [36], we see that only about 16% show this in the EMA dataset. It is possible that this is due to the fact that tinnitus sufferers who are not sick enough to need medical treatment still use the app.

Performance of group-restricted personalised models: Figure 3.17 shows the results for the personalised models that were restricted to selecting from in-group participants. Since the neighbourhoods are already restricted (non-randomly) to in-group participants, we do not compare the performances in the six group-level

[2] each patient is in one of 3 (loudness clusters) x 2 (distress clusters) = 6 groups

models against a kRE baseline. The results at the group level are plotted alongside the unrestricted kNN model (dotted back line).

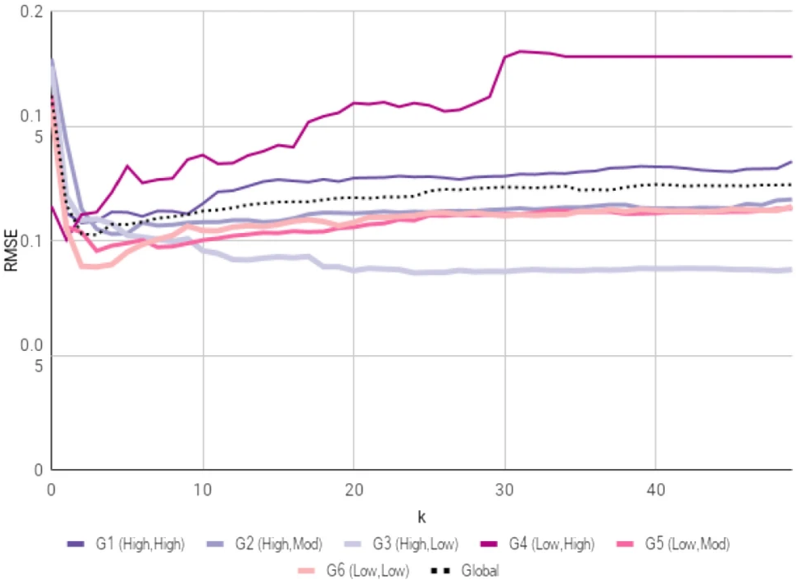


Figure 3.17.: RMSEs for group-restricted kNN personalised models for the six groups. kNN neighbourhoods are restricted to groups with participants of similar loudness (L/M/H) and distress (L/H)

It can be seen that restricting the kNN to in-group patients improves performance for 4 out of 6 groups identified by the expert as compared to the unrestricted kNN model (shown in the figure with a dotted black line). From the two discordant patient types identified, the <low-loudness, high-distress> patients are easier to predict than the <high-loudness, low-distress> patients. The discordant group with low distress in spite of high loudness are seen to be the ones hardest to predict, which suggests that those that low tinnitus distress in spite of high loudness are all dealing with their tinnitus in their own different ways, while the group of patients with high distress in spite of low tinnitus loudness are more similar to each other.

The group level errors also show that the small improvements to most groups gained from restricting the kNN to within-group patients are cancelled out by the high error for the <high-loudness, low-distress group>. Overall, however, it is clear that restricting the kNN is indeed beneficial, and that the neighbourhoods are tunable with expert information. Eliminating 'false' neighbours does indeed improve performance for 4/6 groups.

3.5. Conclusions

This chapter investigated the main question of how entity-similarity can be exploited to train personalised models. We proposed two methods, one that trained personalised models based on the exploiting the 'static' data of similar entities to build models in the 'dynamic' space, and another that built upon this idea to incorporate expert knowledge into the discovered static neighbourhoods.

3.5.1. RQ1.1: Augmenting with the right data for training a personalised predictor

Since too little data is available at the entity level, we follow the approach of augmenting the data of the personalised model with the data of other entities that are 'similar'. However, since similar entities cannot be discovered using the time series or dynamic data, we propose discovering such entities using the static data (like patient age, gender, etc.) that describes them.

We use the kNN algorithm over the static data of entities to find the best entities to augment the data of an entity - however, this still leaves some open questions before the personalised model can be trained:

1. After finding the entities, how to train a personalised model using similar entities' data?

   We explored two methods to train personalised models based on the data of an entity, and that of k similar entities. One was to pool all the data from the $k + 1$ entities to train a single model. We call this 'data augmentation'. The second method avoided pooling the data and focused instead on augmenting the personalised model for entity e with one model each from each of the k neighbours. In our simple case of linear regression models (chosen because many entities have too little data for training), this is analogous to averaging the model parameters, but the idea is also conceptually similar to using a 'local ensemble' of k models from the neighbours voting along with the model of entity e itself.

   We found that data augmentation performed better than model augmentation, and that very short entities make the model prone to learning extreme slopes, especially when the observations are temporally near. Since we would like to not disadvantage short entities in our workflow (they far outnumber the long ones), we focus on data augmentation as our preferred method for training personalised models.

2. How to deal with timestamps at the entity level when each entity has their first observations at a different timepoint?

   We explored training the personalised model with timestamps used as-is without modification, and also aligning all entities such that they have their own 'local clock' - i.e., the first observation for each entity is at $t = 0$.

   It was seen from our experiments that using the timestamps as-is helped when training models in an environment where strong dataset-level trends exist. For the mHealh scenario, which is closer to our intended use case, the difference was less pronounced.

3. To what extent do neighbourhoods improve performance?

   We saw that small neighbourhoods do indeed contribute positively to the performance of personalised level models. However, when we created our baseline that tests the relevance of the exact neighbours that were selected during the training process, we saw that our proposed kRE baseline was very competitive in performance. The unexpectedly high performance of the kRE model, as well as the fact that the kRE shows the same tendency towards favouring small neighbourhoods suggests to us that further investigation is necessary in selecting the neighbourhood.

3.5.2. RQ 1.2: Incorporating expert knowledge into the discovered neighbourhoods

The unexpectedly competitive performance of the kRE baseline model prompted us to investigate whether the neighbourhood computed by the kNN can be improved using expert knowledge. Towards this end, we investigated:

1. What is the expert knowledge, and how can it be integrated into our proposed workflow?

   In the mHealth dataset relating to tinnitus, it is known that there is a minority of users who experience distress that is not concordant with the severity of their symptoms (either they are too distressed for their symptom severity, or vice versa). We used clustering with 3- and 2- Means on the self-reported tinnitus loudness and the tinnitus distress in the static data, and used the membership of each patient in the loudness and distress clusters to restrict the kNN to search only among others who are in-group. Out of six total groups, two were 'discordant': the <low-loudness, high-distress> group and the <high-loudness,low-distress> group.

2. To what extent does the expert input reflect in the personalised modelling process?

   We incorporate expert knowledge about the existence of anomalous tinnitus patients with either high tinnitus loudness and low distress, or vice versa. In our experiments, we found that the group-level average performance for personalised models that were restricted to pick in-group neighbours only performed better than the unrestricted kNN counterpart for 4/6 groups.

3. How does the performance reflect on the discordant/anomalous tinnitus patients? We found that the patients who fall into the group with low distress despite high tinnitus loudness were the hardest to predict out of all six groups. On the other hand, the group of patients with high distress in spite of low tinnitus loudness was the best-predicted group, with an error of <0.1 on a scale of [0,1]. This suggests that the patients capable of dealing with their tinnitus all have their own personal ways of dealing with it, while those that are strongly affected by even mild symptoms are more like each other. This finding was acknowledged to be of value from a medical point of view by the medical experts.

   In short, contrary to what Tolstoy says about marriages, one might argue: "All unhappy tinnitus patients are alike; each happy tinnitus patient is happy in their own way"

4. Towards RQ2: Exploiting data in dynamic domain to improve neighbourhoods

This chapter is based on the outputs from the following papers:

[126] Unnikrishnan, Vishnu et al. “Predicting the Health Condition of mHealth App Users with Large Differences in the Number of Recorded Observations - Where to Learn from?” In: Discovery Science. Ed. by Annalisa Appice et al. Cham: Springer International Publishing, 2020, pp. 659–673

[124] Unnikrishnan, V. et al. “Love thy Neighbours: A Framework for Error-Driven Discovery of Useful Neighbourhoods for One-Step Forecasts on EMA data”. In: 2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS). 2021, pp. 295–300.

- The work presented in 4.2.1 exploring HMMs and Granger causalities is as-yet unpublished.

4.1. Motivation and Comparison to Related Work

In the previous chapter, we explored methods that train personalised predictors based on neighbourhoods discovered on the static data. However, we saw that our proposed baseline methods that augments the data of an entity with another randomly selected entity performs very competitively to our proposed methods. This suggests that relying solely on the static data is not sufficient for learning personalised models. Towards this end, we want to explore methods that can also exploit similarities in the dynamic data while discovering neighbours for training personalised models.

However, we would also like to build upon our previous results and adapt our modelling process to make it more realistic. It was seen in the previous chapter that the kRE method that chooses random neighbours has an advantage in the modelling process when the timestamp of the observations is included in the modelling process (see Section 3.5). The advantage gained from including the timestamp increases in datasets where there is a systematic tendency towards higher/lower values as time progresses (for example, the tendency towards better air quality over time, possibly due to legislation, and the tendency towards 5 star reviews over time, due to an increase fake reviews). In order to not give our methods an unfair advantage by leaking this dataset tendency, we change our personalised modelling problem from one of predicting the value of a target variable at time t given all other variables at time t, to predicting the target variable at time $t+1$ given all the variables (including the target) at time t. This short term forecast is acknowledged by the expert to be valuable [134, 19, 116], since it can serve as an early warning system, especially for psychological issues, for changes in the disease state.

There are many ways to discover similarities for data in the dynamic domain, the most obvious of which is simple euclidean distance. While simple, the inapplicability of this distance measure is well reported [32]. The chief issues are the sensitivity of euclidean distance to phase and time shifts, scale, and most importantly for our work, its ability to deal with sequences of unequal lengths.

A popular alternative to euclidean distance is the phase and time shift invariant DTW [101]. DTW is a dynamic programming algorithm that computes the distance between two time series after computing an optimal alignment (where the individual time series are 'warped' and/or squished) between them. However, it has the drawback that the alignment can sometimes cause a so-called 'pathological warping'[140], where a small part of one time series is mapped to a very large proportion of the other during the warping process. Zhang, et. al. [140] solve the problem by limiting the number of links during the optimisation process, but other works also limit pathological warping can be avoided by setting various constraints on the alignment path discovered by the DTW, like using the Sakoe-Chiba band or the Itakura parallelogram[28].

Using historical dynamic data to predict the future development of a sequence of observations is of great interest for stock market forecasting due to its direct economic value. Various methods exist that exploit historical data for learning, but the main idea is to find the most similar sequences in historical data, and once they are retrieved, combine them to create a forecast of the future.

A dynamic multi-perspective personalised similarity measurement is proposed in [141], where the time series data is segmented into multiple time windows, and during similarity computation, greater weight is given to more recent time windows. The similarities between the segments themselves are computed using Canberra distance [11] with DTW. Canberra distance is chosen instead of euclidean distance because of the sensitivity of euclidean distance to 'singularities' (i.e., a single anomalous observation with a large deviation) - this single anomalous data point shifts the entire distance between two sequences towards higher values. The Canberra distance is a dimensionless quantity that captures the mean pointwise deviation between two equi-sized sequences normalised over the absolute magnitude of each of the points (thereby downweighting the impact of a single outlier). Integrating the Canberra distance into DTW solves the drawback of Canberra distance being sensitive to time shifts in the sequences.

Several derivatives of the DTW algorithm exist for dealing with specific problems. AWarp [70] proposes running DTW on a run-length encoded time series vector [118] so that the distances between sparse time series can be computed efficiently. Extensions like Blocked DTW utilise representations that extend the idea of AWarp so that they can capture (non-zero) value repetition [115] to make the dynamic time warping faster, more accurate, or to allow approximate distance computations based on transformations applied to the source time series. These methods also rely on representing the original time series using different aggregations before feeding them into the DTW algorithm.

Other methods to compute similarity between time series aim to find transformations of the time series to allow certain comparisons. [57] encodes the time series as a summation of kernel functions, and the coefficients for the signal (which are zero at time points where the kernel are not applicable) are found by a probabilistic maximum likelihood approach. The idea is that the model is expected to naturally denoise the original time series. Methods like [58] uses a bag-of-words approach to find subsequences in the time series, but focus on long term similarity computation by representing time series as a histogram of frequencies of patterns (bag of words). Methods like Piecewise Aggregate Approximation (PAA) [49] chunk the time series into equisized bins, and replaces the value in each chunk with the mean value in the bin, and this is extended upon in Symbolic Aggregate Approximation (SAX) [59] where these means are replaced with letters from an 'alphabet' to create words, which can then be mined with algorithms inspired for language processing. The assumption is that the long time series can be broken down into several short segments, each of which can be featurised separately, and then some combination of the segment-level information can be used to represent the original time series.

The idea of describing time series as the parameters of models that predict them has also been explored in works like [50, 136, 100, 29]. In [50], the time series are described as a mixture of Autoregression (AR) models, which is extended in [136] to Autoregressive-Moving-Average (ARMA) models. The parameters of the models learned on these time series form the basis of featurising the individual sequences, which are used for classification tasks - but could conceivably be used for time series similarity as well. Apart from AR and MA models, the HMM is also used for representing a time series, where [100] propose a method for matching the states of different HMMs fit for different time series, and computing a dissimilarity between those 'matched' HMMs. A similar approach is followed in [29], where a separate HMM is fit on every sequence, and an HMM-distance measure is used to perform clustering. In our case, the idea of using model parameters to summarise a time series is a useful one, but we would need to accommodate the fact that our solution needs to allow for big differences in the lengths of the data - where the smallest sequences might be too short to train reliable HMMs. Even if the models were trained on the short sequences, the resulting state spaces might not be comparable, even if some method created verifiably correct representations.

Studies like [40] have successfully applied HMMs to EMA data, and have found that subtypes of schizophrenia are captured by the HMM model. They ensure the learning of subtype-specific individual variability by fixing the transition matrix to be block-diagonal, where each block is a cluster. Each patient can only transition between blocks that are specific to their own cluster, so the practical implementation is straightforward - they train as many HMMs as there are clusters. In the extreme case of 'fully personalised' models, this would be one HMM per input sequence, with the other extreme being a fully global HMM model where all patients can visit all states (i.e., patients are not 'barred' from visiting some disease states by virtue of their pre-identified phenotype). More importantly, however, they acknowledge several problems that are critical to training personalised predictors, especially based on EMAs: Short sequences at the individual level, large dimensionality, unknown (and possibly small) number of underlying states, idiosyncrasies in responses, etc. However, HMMs are shown to be useful in modelling disease subtypes of data capturing psychological affect [38], and for detecting the best time to sample patient experience based on phenotype and individual variability[41].

The idea of Granger causality as a way to measure relationships between time series is a mature one [31], with applications in fields ranging from politics [26] to gene expression [27]. The utility of Granger causality as a more generic tool is also evident from the fact that many works use it as a first step to describe the time series properties and relationships in facilitating downstream data science tasks. For example, [137] use Granger causalities to describe the relationships between multivariate time series, and use the representation to perform a downstream classification task, while [27] solve a clustering problem for gene data, where they propose extending the current methods that cluster based only on expression data with more ‘functional’ information, where the Granger causal relationships between the time series are also considered.

Extensions like [2, 1] enable the use of Granger causalities towards time series sampled irregularly. This is done through aligning the time series being investigated for Granger causalities using DTW, and then computing the Granger causal relationships only on the aligned series. While a promising extension, its applicability to EMA data is still limited, since ratio of the shortest to the longest series is still too large to make DTW applicable. However, the idea of limiting the comparisons to comparable series is still a promising one. For example, previous work in our group [42] has explored the applicability of Granger causality in determining commonly occurring relationships in the EMA data of tinnitus. It was found that the irregularity can be tackled by limiting the analysis to within-day observations, where it was also seen that some registration-time questionnaires are associated with certain Granger causal relationships in the EMA data.

4.2. Exploiting dynamic data for neighbourhood selection

The following sections describe the main approaches we explore towards including the EMA data into the similarity discovery process. We begin in Section 4.2.1 with two proposed methods to include the EMA data into the neighbourhood discovery process while training personalised models, while Section 4.2.2 and 4.2.3 explore a broader definition of similarity at a meta-data level, by exploring whether the users who contribute a relatively larger portion of the data need different definitions of similarity compared to those that contribute relatively little.

Please note that the methods and the results proposed in Section 4.2.1 are not published work, and while it is unconventional to include such ‘failed’ experiments in the main body of this work, they still contribute towards a convincing argument, since the metadata-based experiments were only considered once the data itself was proven to not bring substantial benefits to the neighbourhood discovery process.

4.2.1. RQ2.0 Summarising EMA data to discover similar patients

The idea of including EMA data to contribute towards the personalised modelling process is a natural next step to our findings in Section 3.5. The previous chapter explored the degree to which the static data of an entity can help us identify similar entities when learning personalised models. In this chapter, we explore exploiting data from the dynamic data towards building personalised predictors.

We explore summarising the time series information using two methods: One that uses a HMM to learn the underlying disease states, and another that learns the intra-day Granger causalities in the EMA data. Once these relationships are learned, we can replace the variable-length EMA time series with their fixed-length representations that capture the information learned by the modelling process. The following sections describe how the EMA data is summarised and then used in the neighbourhood discovery process.

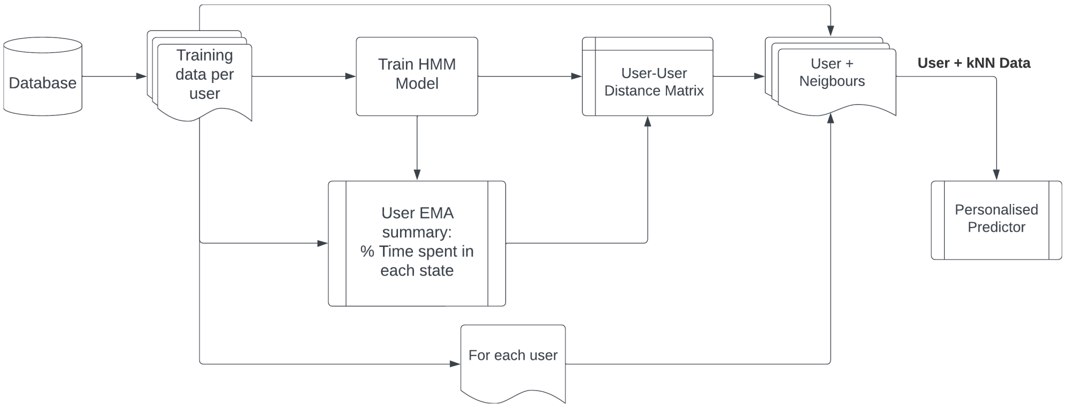


Figure 4.1.: The process for training personalised models using HMMs to discover users with similar EMAs

RQ2.0.1 Summarising EMA data using HMMs

As discussed in Section 4.1, the goal of our personalised predictors is changed from that presented in Section 3.2 to producing personalised models that forecast the target variable for the next observation given the data from the previous observation. In this workflow that uses hidden markov models, we propose that the patients move through some underlying disease 'states', and that similar patients are those that progress through similar disease states.

There are three main components in the workflow:

- Utilising the available data to create the underlying state-space model
- Discovering the neighbours of the user using the state-space model
- Creating the model with the optimal neighbourhood

The first step in the modelling process is to fit an HMM on the EMA data. The goal of the HMM model output is to have a representation of the underlying states of the disease, and how they relate to one another through the transition matrix. Since there is no clear understanding in the medical context regarding disease states (apart from the already discussed result that there are anomalous tinnitus patients whose distress is not commensurate with their tinnitus loudness), we are forced to follow an approach of training an HMM model that is complex enough to capture variability between patients, while not becoming overly complex. In order to estimate the correct number of states, it is of course necessary to consider quality measures like Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and log likelihood, but we also need to consider that a well fit model also needs to capture representative disease states. In other words, a model that learns states that are visited only by one or very few patients needs to be penalised.

An overview of the full HMM-based personalised neighbourhood model is shown in the Figure 4.1. The first step is to decide the logic for holdout validation, where the choices are between holding out the last x% of each user's data for validation vs. holding out the entire data of x% of the users in the holdout set. Each method has its own disadvantages - for example, holding out x% of the data from each user will make it impossible to know how good the HMM representing the disease really is, since a little bit of each user's data has been fed into the HMM during training. Holding out entire sets of users, on the other hand will add the dimension of complexity that the errors that we report at the end will be computed over users of different lengths, and the held-out users can be chosen ‘unluckily’ to contain only users that are short (such users are the majority in the EMA dataset anyway). Since our goal is personalised models, we will choose to hold out the last observations of each user for validation, in line with our decisions in Section 3.2.

Training the HMM model: In training the HMM model, we aim to create a map of the underlying states in the subjective experience of tinnitus. Two main decisions need to be made for training HMM models on EMA data - the number of states, and the variables from the EMA questionnaire that are used to train the HMM model.

There is no known ‘correct’ number of underlying states in tinnitus, except the medical intuition that there are patients whose subjective experience of tinnitus is anomalous (low distress in spite of high loudness, and vice versa). While it is in general preferable to train the state space models on the full feature space captured by the EMA questionnaires in each app, training on a subset of comparable questions can make a state space model that is transferable between datasets. However, this is also something that needs empirical validation.

Training personalised models based on the HMM output: Once the HMM model is trained, it allows us to create a ‘summary’ of each user, where each user can be represented by the same number of features as there are states in the state space model. The user is summarised by the percentage of observations emitted from each hidden state learned by the HMM. i.e., in a four-state model, a user for whom 100% of the (training-data) EMA observations were all emitted form state 1 will be featurised as <1.0, 0, 0, 0>. It is clear that counts cannot be used since the users differ in their lengths.

Although the method will fail to distinguish directional information (i.e., two users who spend 50% of their time in each of two states, but in opposite order), the intuition behind our proposed method is still that users who spend similar amounts of time in similar states are more similar than those that do not. Once each user has been represented as their state-space-occupancy vector, the neighbourhood of each user can be computed using the k-nearest neighbours algorithm, and the data of each of the neighbours discovered can be combined to train the personalised model.

Transferring the disease states from UNITI to TYT: Since the experience of tinnitus is independent of the app that is used to record it, it is also possible to learn the disease dynamics in one app, and apply the rest of the workflow on the data of another. In this case, it would make sense to learn the disease dynamics on the data of the app that has the cleanest, most regular data. In order to investigate this, we also learn the HMM on the full data of the UNITI app (since it has more regularly sampled observations for each user), and apply it to decode the sequence for the TYT dataset. The rest of the workflow is held identical, except that the full data from the UNITI app is used for training, instead of only the training data. Transferring the HMM model to another dataset comes with three main benefits:

- overfitting can be avoided,

- an HMM learned over an older app can be deployed immediately into the prediction workflow of a younger app that has not yet collected enough data - this can be useful for future apps to “hit the ground running”, and
- it is possible to learn the HMM model on the app that has ‘cleaner’, more regular data to decode the possibly irregular, sparse sequences collected from apps with less clean data.

Formalisation: More formally, we saw in Section 2.3 that for each entity $e_i \in E$, the dynamic data D_i consists of a sequence of timestamped observations $D_i = \{o_i^1 \dots o_i^{T_i}\}$. Each entity e_i has an ‘entity length’ equal to the number of observations in the dynamic data D_i, which in the above case would be $|D_i| = T_i$. Although not technically correct, let us assume for notational simplicity that that T_i observations are the length of the dynamic data in the *training data* of entity e_i. Since we are only learning the relationship between the current observation o_i^t and the target variable in the future observation and $x_{target}^{t+1} \in o_i^{t+1}$ where $t \leq T_i - 1$, we can ignore the exact timestamp at which an observation arrives.

Our goal is to learn an HMM model on the dynamic data D_i of each entity e_i. The number of states in the HMM model is chosen to balance metrics like AIC and Log-likelihood, but also to create a set of states that are sufficiently representative of the disease dynamics, without getting too specific to individuals (i.e., a state that is visited only by one or very few individuals is not ‘useful’).

Once the HMM is trained, the model can now be used to describe each user as $user\_encoding(e_i) = state\_visitation\_percentage(HMM, D_i))$. The user encoding converts the variable length sequence D_i into a fixed-length representation with H dimensions, for HMM models with H states. This is done by encoding the data in sequence D_i as a sequence of states $1 \dots H$ that generated each observation in D_i, and then the value at $h \in 1 \dots H$ is the percentage of observations in the sequence generated from that state. This encoding implicitly accommodates for differences in length and represents all users in the same number of dimensions as the number of states in the HMM. Each dimension is bounded by 0 and 1 and the sum of all the dimensions adds up to 1.

Once the fixed-length representation of each patient is created, this is used as the basis by personalised model to discover the neighbourhood for each entity e_i. This is accomplished through the kNN algorithm that operates on the user encoding of the entity. The output of the kNN is an ordered set $kNN(e_i) = kNN(user\_encoding(e_i), \mathcal{D}) = n_i^1 \dots n_i^k$, where each $n_i^j \in E, j \neq i$, and $similarity(user\_encoding(e_i)$, and $user\_encoding(e_j)) \geq similarity(user\_encoding(e_i), user\_encoding(e_k)), \forall k > j$. In our work, we compute kNN using the euclidean distance (the inverse of which is used as the similarity).

Once the neighbours of an entity have been identified, the next step is to train the personalised model. Since the output of $kNN(e_i)$ is a set of neighbours $n_i^1 \dots n_i^k$, the dynamic data of the entity D_i is combined with the dynamic data of each of the neighbours $D_n, \forall n_i \in kNN(e_i)$. i.e., the personalised model for e_i is trained as

$$train\_personalised\_model(e_i) = train\_model(D_i \cup \{\bigcup_{n_i^1}^{n_i^k} D_{n_i}\})$$

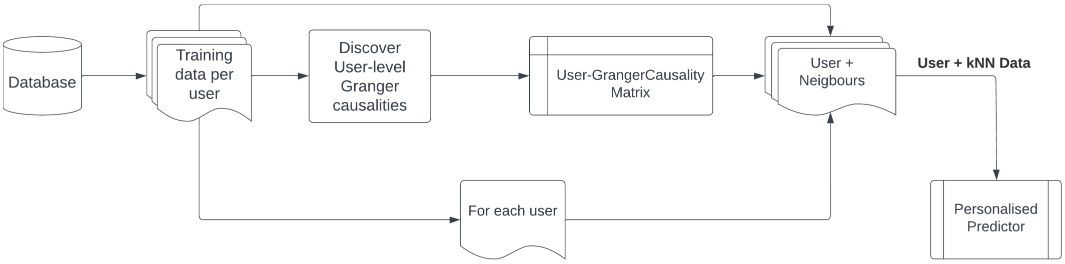


Figure 4.2.: The process for training personalised models using intra-day Granger causalities to discover users with similar EMAs

RQ 2.0.2 Summarising EMA data using Granger causal relationships

Apart from the HMM model, we turn to another method to compress the time series information available in the dynamic data of an entity by reusing a previous result from our workgroup [42]. Given that there is a time-of-day dependence between tinnitus loudness and distress [86], and given the heterogeneity of tinnitus [9], the work in [42] builds a model that extracts Granger causal relationships at the patient level. The work also explores the Granger causal relationships at the individual level, their likelihoods of appearing in relation to each other, and their connection to the static data. However, we will work more conservatively and use only the individual-level Granger causalities since there is a likelihood of compounding the Type I error.

The general overview of the process is shown in the Figure 4.2. The steps are broadly similar to those in the HMM workflow, except that the output of the model that learns the Granger causal relationships is a list of Granger causalities that are expressed in the EMA of a user. This list (by default, a one-hot-encoded list of Granger causalities as a binary vector) can directly be ingested to create a user-Granger causality matrix, over which you can compute a kNN. The rest of the process is similar to the methods already introduced, the model is trained on the aggregated data of the patient and their neighbours. The neighbourhood size needs to be tuned, like in all cases.

Discovering Granger Causal Relationships: The GC Discovery component is run for every patient $e_i \in E$. We use a method heavily inspired by [42], except the important difference that we do not investigate whether GC relationships exist only within the EMA observations spanning the same day, since that restriction limits the computation to very few patients.

Inferring GC between variables x_A and x_B (where $x_A, x_B \in X^p$) involves training two models - the unrestricted model UR ,and the restricted model R. The unrestricted model includes all variables $unrestricted\_vars = x_1^t \dots x_n^t$ (where $x_A \in unrestricted\_vars$) as inputs to predict the future of x_B (x_B^{t+l}, where l is is the number of lag periods. The restricted model R is trained to include all variables except x_A to predict the same lagged variable in the unrestricted case, ie $restricted\_vars = \{x_1^t \dots x_n^t\} - \{x_A\}$. The models are vector autoregressive models trained using an OLS estimator. Once the models are trained, we use an F-test to confirm if the unrestricted model UR is significantly better than the restricted model R, and if yes, we know that x_A Granger causes x_B for patient p.

The above procedure is repeated for every pair $x_A, x_B \in X^p, A \neq B$ to get matrix where 1's indicate the existence of a Granger causal relationship between A and B for user p. The output of this component is a set of $|p|$ matrices, one for each user. Each matrix $gc\_matrix(p)$ has along the columns the variables that granger cause the variables in the rows. This means that you can get a vector $get\_gc\_vector(p, gc\_target)$ where $gc\_target \in x_i, i = 1 \ldots n$

The Neighbourhood Component: This component ingests the $|E|$ user-level matrices generated by the GC discovery component, and aims to convert it into a list of neighbouring users ordered in decreasing order of similarity to p. The similarity is computed on the basis of a particular variable x_{gc_target} (one row in the matrix), so that we can find other users who are similar to the current user w.r.t their causal relationships in the vector $get\_gc\_vector(e_i, x_{gc\_target})$. In order to better capture the interrelationships between the causal relationships that appear together and separately from each other, we apply the UMAP dimensionality reduction [67] on the list of vectors. Apart from capturing common relationships among frequently appearing causalities, it is also expected that the possibility for visualisation of the dimensionality-reduced causal vectors might help the practitioner choose an appropriate gc_target variable when computing neighbourhoods. The neighbourhoods themselves are computed using a simple kNN approach, where the $kNN(e_i, gc\_target, k)$ are the k nearest patients as computed over the dimensionality-reduced representations of the patients' causality vectors on gc_target.

The Personalised Model: Once an appropriate neighbourhood of patients $kNN(e_i, gc\_target, k)$ has been identified for a patient e_i, the data from all the users $train\_data(n_i^k)$ $\forall n_i^k \in kNN(n_i^k, gc\_target, k)$ are concatenated (for all $k \in 1 \ldots k$), and a one-step forecaster is trained on the combined data, where all the variables at time t are used to forecast the target variable at $t+1$. In this case, the target variable is chosen as the distress caused due to tinnitus, since it is of clinical relevance. One one-step forecast model is built per patient e_i, and each of these models can now be used in the next step to predict the next-day distress given the EMAs of the current day for each of the e_i users in the test data $test\_data(e_i)$.

4.2.2. RQ2.1 Exploiting user interaction length to discover neighbourhoods

Section 4.2.1 explores two methods designed to exploit the EMA data in finding neighbours. However, all the methods proposed so far share the disadvantage that many users are necessarily excluded from the personalised modelling process, who contribute too little data for learning. Additionally to the fact that all 'short' users are excluded, there is also the fact that all new users to the system are necessarily 'short' when they are in their early stage of interactions with the system.

Our design decisions were mainly guided by the needs towards the TrackYourDiabetes app defined in Section 2.2, and like described in Section 4.2.1, we focus on the scenario of forecasting the target variable for the next observation for a user given the previous. Since the goal of the app is to achieve patient empowerment, the target variable is chosen to be the "feeling of control" that a patient has over their diabetes. The EMAs of each user constitute our entity-centric time series with a patient for every entity e_i in the dataset. However, we will split this set of users into two groups based on their behaviour in the app - all users who contribute less than an arbitrary cut-off length are considered “short" users, and those that have time series of length that exceed this threshold are categorised as “long" users. We will refer to these users as belonging to U_{short} and U_{long} respectively, and aim to investigate the following questions:

- To what extent is the data of users in U_{short} predicted by the data of U_{long}? Does a model trained on data of U_{long} transfer to U_{short}?
- To what extent can the model for U_{short} be personalised by adding data of the short users as they become incrementally available (i.e., as they become longer)?

Splitting the users into two groups based on length: As already explained, each user in the dataset belongs either to U_{long} or U_{short}, depending on whether the length of their EMA time series exceeds the threshold τ_{length}. while theoretically unbounded, τ_{length} is designed to be set in a data driven way, since the number of short users will depend on the maturity of the app, and also the nature of user interactions. Since the distribution of user interactions follows the power law in most EMA apps, setting this threshold to a small number should already split the full set of users into two groups with a relatively large bias towards short users. The sequence of observations for each user is a mix of categorical and numerical variables, as can be seen in Table 2.5.

Handling idiosyncrasies in categorical data using a TF-IDF inspired approach: Since the main goal of this work is to create personalised predictors, it is necessary to handle the fact that different patients respond to different questions with different tendencies. Using the sklearn ‘StandardScaler’ (applying $\frac{x-\mu}{\sigma}$) at the user level can centre the numerical variables on the user, and redefine the time series where each person’s observations are reframed as their deviations from their means. For example, if 2 patients with means $[mean(e_1) = 10, mean(e_2) = 20]$ and standard deviations $[\sigma(e_1) = 1, \sigma(e_2) = 2]$ generate the data $D_1 = [10, 11, 9, 10]$ and $D_2 = [20, 22, 18, 20]$, then applying the standard scaler to each patient separately results in two time series $D_1 = [0, 1, -1, 0]$, and $D_2 = [0, 1, -1, 0]$. It is clear from this toy example that the user-level application of the standard scaler can accommodate for differences in the tendencies between individuals, and capture the relative changes between them. However, no equivalent is readily available for categorical data. To give an example analogous to the numerical example above, two patients may report on day 100 that they had no signs of hypoglycemia. However, if one of those patients has hypoglycemia every day, and the other has signs of hypoglycemia for the first time in 100 days, then these two observations are not equivalent semantically, even though the data would suggest so.

We handle this issue by preprocessing the categorical data to accommodate for the fact that some answers are more likely than others. Please note that we need not to accommodate for the frequency of an answer at the dataset-level, but at the patient-level (i.e., we need to capture the ‘surprise’ in the fact that a person who normally answers a question with a “no" has said “yes", even if “yes" is the most common answer in the dataset). We propose a TF-IDF [92] inspired method to handle the amount of surprise in a submitted answer by focusing the user’s answer as a 'word' that appears in the user's history of answers, which can be seen as a 'document'. The one-hot encoding of the categorical features results in a binary list that has the same size as the number of options in the dropdown. After getting this list, we apply the formula $preprocessed\_value(term) = f_{term} \dot{-} log \frac{n_{term}}{N}$. Since the categorical values are chosen from a dropdown list, the f_{term} is 1 for the value selected from the dropdown list, and 0 for all others. The inverse document frequency component measures the frequency of a word in the user history.

Learning on U_{long} to predict U_{short}: Given the data of $e_l \in U_{long}$, we learn a model where for each time point $t \in 1 \dots t$, we learn a linear model to predict the value of the target variable at the next time point $t+1$. The last observation of each user's data is excluded from the training dataset, since no known label exists for the target at the next time point. This model can now be applied to each user $e_s \in U_{short}$, where the next day's target value for the data of the user $e_s = x_1 \dots x_{t-1}$ is predicted. This serves as a baseline to compare the degree to which the data of long users predict the data of short ones.

Personalising the U_{short} model with incrementally available data: The method developed above with the model trained on all data from U_{long} is not personalised, so we investigate the degree to which a prediction can be obtained that is specific to the history of the user $u_s \in U_{short}$. We propose to do this by creating an additional model that is trained on the user's past, and delivers predictions along with the model trained over all long users. This is done by training a kNN regressor that has been trained only on the accumulated observations of user u_s. Please note that the last session is not part of the training data, since the value of the target variable from the next day is as yet unknown.

Each observation can not be predicted by the user-level kNN regressors that see patient specific data, and also by the U_{long} model trained over the data of all long users. To combine them to get a single prediction that adjusts its weights as incremental data becomes available, we propose that we combine the two predictions based on the accuracy of each of the models (like in [10]) towards the user. i.e., for each user, we store the prediction errors towards the observations of that user from (a) the U_{long} model, and (b) the user-specific kNN regressor. After each prediction, the errors are updated, so that the next prediction can be weighted to reflect the accumulated mean errors from each model. It is of course necessary to weight according to the inverse of the error, and not the error itself, since higher errors are not desirable. An overview of the workflow is shown in Figure 4.3.

4.2.3. RQ2.2: Improving the neighbourhoods discovered using user interaction length

Section 4.2.2 explores fitting personalised models for "short" users (users whose EMA time series lengths do not meet a cut-off threshold τ_{length}) by using a combination of personal and global models. The predictions of global models trained over long users (their time series are longer than τ_{length}) and an incremental kNN regressor over the short user's own sequence of observations can be combined weighted by the errors from each of the models towards the user.

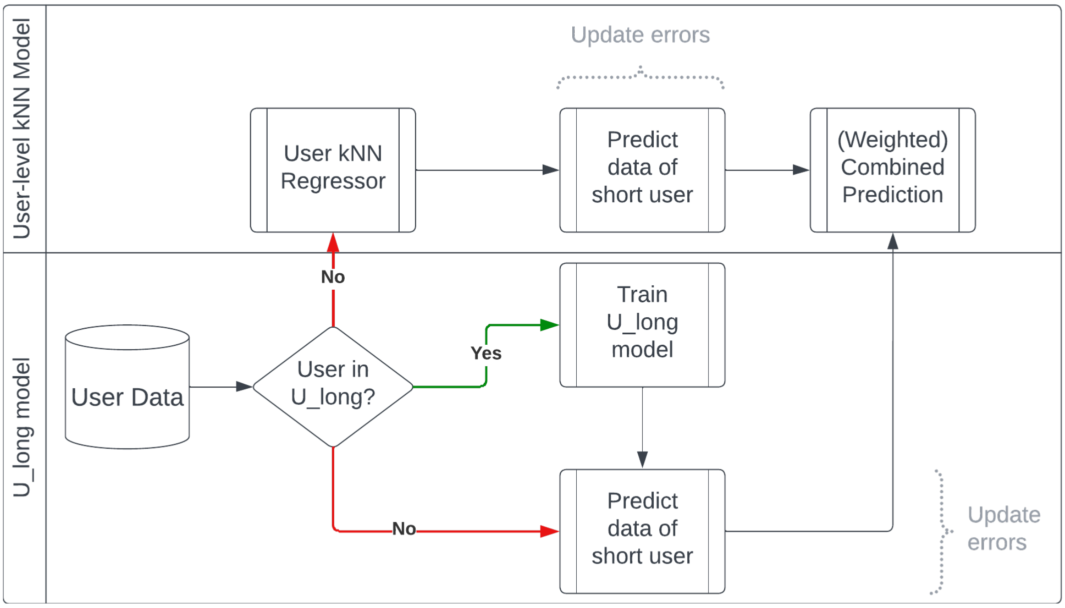


Figure 4.3.: The process for training personalised models for users in U_{short}

In this section, we explore whether each long user U_{long} is equally valuable in the process of training personalised predictors for users in U_{short}. In order to find the best neighbourhood for a user in U_{short}, we propose a workflow that discovers the best neighbourhood by searching through an ordered list of users in U_{long}, incrementally expanding the neighbourhood to include the data from each long user as long as their ability to predict the data of the short user $u_s \in U_{short}$ is not compromised. It is indeed true that the workflow is wasteful in terms of the number of models trained, but the reader is encouraged to remember that the amount of data is small and the models simple. The runtimes of even unoptimised code is expected to be well within feasible limits. It is also important to consider that while performance benefits are a nice-to-have, our proposed methods come with two additional advantages:

- our methods deliver, along with personalised models, a personalised neighbourhood. In contrast with previous methods we have looked at, we are proposing a method that would deliver a personalised neighbourhood with a personalised neighbourhood size. The accuracy of the model (compared to a non-personalised one) serves as a good starting point for the physicians when it comes to trusting the model, while the main benefits might be in the ability to ask questions like ‘what properties do the similar users share?’. A personalised neighbourhood size enables additional questions like ‘why are some people predicted by fewer neighbours than others’. Unfortunately, such questions are not expected to be answered by non-physicians, and, in cases of diseases like tinnitus, it might be that they are not yet answerable, since the features that contribute to certain subtypes in the experience of tinnitus might not yet be captured in the data (for example, for as-yet-unknown genetic similarities).

- our methods are inherently robust to data removal requirements. i.e., not just the removal of one person’s data from the database, but also the removal of the effects of that person’s data on the models. In our case, the complete removal of one person’s data would be as easy as deleting that person’s model, and a re-training of every model that uses that person’s data (a simple reverse-lookup

Code	Questionnaire	Question
R_1	Random	Current estimation of blood sugar level
R_2	Random	Actual measurement of blood sugar level
F_1	Food	Average blood sugar level before eating
F_2	Food	Level of hunger
EOD_1	End-of-day	Frequency of sugar level measurement
EOD_2	End-of-day	#Minutes of physical activity
EOD_3	End-of-day	Total bread units consumed
EOD_4	End-of-day	Signs of hypo- or hyper- glycaemia
EOD_6	End-of-day	Feeling of control over diabetes

Table 4.1.: A summary of the information captured by the three EMA questionnaires in the TYD app

in the dictionary) in another person's neighbourhood without the data that needs to be removed. Our methods also naturally support tiered access rights - that some users might consent that their data be used in the neighbourhood discovery process, and others, not. Out-of-the-box support for such requests is not trivial in times of strengthening regulatory requirements (like with GDPR).

Definitions Our workflow was developed with the TrackYourDiabetes (TYD) apps described in Section 2.2.3. Separate versions of the apps were rolled out in Bulgaria and Spain, and the datasets used by them are the ones that are explored in this work. Unlike the tinnitus apps, the TYD apps have three EMA questionnaires that are collected from the users - the food questionnaire asked at meal times, the 'random' questionnaire that is asked randomly multiple times a day, an the 'end-of-day questionnaire' that is asked only once, at the end of the day. The different cadence of the three questionnaires needs to be handled if the information of the three questionnaires is to be combined. Table 4.1 shows the list of questionnaires and the information they capture.

We continue with formalising some of the frequently used terms below.

F_n, R_n, and EOD_n: The food, random and End-of-day questionnaires are referred to by F, R, and EOD. The n in the subscript denotes the feature from the questionnaire.

Session: A user submission for any questionnaire, captured at some point in time is a 'session'.

U_{long}, U_{short}: Similarly to the convention used in Section 4.2.2, the set of 'long users' U_{long} is a set of users whose EMA time series exceed threshold length T_{length}. All users who do not exceed this threshold are 'short users' in set U_{short}.

Baseline Model B: This model is the model built on all the data of all users in U_{long}.

Similarity Measure: The function used to measure the similarity between users in U_{long} and U_{short}. The various options available for measuring similarity. Since user behaviour in EMA apps differs so strongly between users who submit little data and those that submit more, the medical experts suggest that one option to measure similarity would be to look at users who are similar in their interaction patterns with the app (the intuition being that there is some underlying similarity in the disease between the short users, that makes them not use the app long). Towards this end, we explore basing the similarity either on the data of the EMAs submitted, or on the metadata of the submitted EMAs. In case of the metadata, we use the *number of submitted responses*, and in case of the data, we use the data submitted in the response. In either case, the function we apply is cosine similarity.

Since some questionnaires can be submitted multiple times per day, while the EOD questionnaire can only be submitted once, we compute the daily average for all questionnaires that were submitted more than once. Additionally, since cosine similarity can only be applied over vectors of similar lengths, we need to fix the size of the vector in one of two ways:

- Choose the data from the first-N days of user interactions: the first N observations of $u_s \in U_{short}$ are compared against the first N observations of $u_l \in U_{long}$. This method measures the similarity in the early experience of app usage between the two users. However, has the disadvantage that the data used for the user in u_l can be very 'old'.

- Choose the data from the last-N days of user interactions: The last N days of observations for $u_s \in U_{short}$ are compared against the last-N days of $u_l \in U_{long}$. Since the data for user u_s is all recent (ie, they are short, and it is the latest data you have), you prioritise the similarity for people who are experiencing the disease similarly in the most recent data for u_l.

Personalised neighbourhood $PN^i = PN(e_i)$: A personalised neighbourhood for a patient/user $e_i \in U_{short}$ is a set of users $PN^i \subseteq U_{long}$ such that a quality criterion C is maximised (in our proposed workflow, this is the RMSE)

We propose two ways to build a personalised neighbourhood:

- Early termination: For each user $u_s \in U_{short}$, the early termination method discovers a personalised neighbourhood $PN(u_s) = \{e_1 \ldots e_k \in U_{long}\}$ such that for some similarity function S, $S(e_i, e_j) \geq S(e_i, e_k) \forall k > j, 1 \leq i \leq k-1$. A model $\Omega(PN(u_s))$ trained over this neighbourhood is subject to the condition that the next user added to the neighbourhood increases the error (by a margin greater than the threshold τ_{error}). In other words, users are added to the neighbourhood in decreasing order of similarity as long as the error of the model in predicting u_s decreases. A threshold τ_{error} is set so that very small increases in error can be ignored. This is necessary because the early stages of the neighbourhood can result in volatile changes to the error, and we expect that very small increases in error might be tolerable in interest of better generalisation thanks to the additional data of the user. Algorithm 4.1 presents the algorithm.

- Exhaustive search: The previous method searches an ordered list until the first user is found (in decreasing order of similarity) that increases the prediction error of the personalised model over the data of short user $u_s \in U_{short}$. Some degree of tolerance is built into the search progress to avoid local minima, but the neighbourhood discovery process is highly dependent on the ordering of users. Given that our earlier results have already shown that a kRE model can extract roughly as much information as a kNN, we can make our system less dependent on the ordering by searching the full list of users regardless of whether the added user increases the error. For example, consider the case where 3 users A, B, and C are to be searched when building a personalised neighbourhood. If the 'true neighbourhood' is {A, C}, but the addition of B after A affects increases the error, then the neighbourhood search stops and returns only A as neighbour. This problem can be avoided by using exhaustive search. Exhaustive search has also the additional benefit that the results serve as a way to validate the results from the early termination - for example, if the discovered neighbourhoods diverge between early termination and exhaustive search, then this shows that the similarity function is not ordering users in order of decresing 'usefulness'. Algorithm 4.2 presents the algorithm for exhaustive search.

Algorithm 4.1 Discover a neighbourhood for short user u_s from U_{long} using early termination.

```
Require: User $u_s \in U_{short}$, user $u_l \in U_{long}$, threshold $\alpha$, similarity function S()
user_personalised_models = dict()
for $u_s \in U_{short}$ do
    training_data = $\emptyset$
    Sort $n_i \in U_{long} | S(u_s, n_i) \geq S(u_s, n_j) \forall i < j$
    training_data = $training\_data \cup data(n_1)$    ▷ Start with nearest neighbour
    Train model $\Omega(training\_data)$
    prev_error = $RMSE(\Omega, data(u_s))$
    $\Omega_{prev} = \Omega_{curr}$
    for $i | i \in 2 \ldots |U_{long}| - 1$ do
        training_data = $training\_data \cup data(n_i)$
        Train model $\Omega_{curr} = \Omega(training\_data)$
        curr_error = $RMSE(\Omega_{curr}, data(u_s))$
        if curr_error $\leq$ prev_error + $\alpha$ then
            $\Omega_{prev} = \Omega_{curr}$
            prev_error = curr_error
            Remove $n_i$ from training_data
        else
            break                ▷ Stop expanding neighbourhood, return model
    user_personalised_models[$u_s$] = $\Omega_{curr}$
return user_personalised_models
```

4.3. Experiments

This section presents the results for the two methods explored under Section 4.2.1, and subsequently those in Section 4.2.2 and Section 4.2.3. The methods described in Section 4.2.1 are tested on the TYT and UNITI datasets for tinnitus, while the methods described in Section 4.2.2 and Section 4.2.3 use the data generated by the two instances of TYD app, deployed in Spain and Bulgaria respectively. The data sharing agreement does not allow the data of these apps to be combined, so the performances are reported separately for each. Theoretically, though, the data of the two versions of the app can be combined, since they differ only in the language in the user interface. An additional warning is that the TYT and UNITI dataset are constantly evolving, and experiements conducted at different times have used different amounts of data.

Algorithm 4.2 Discover a neighbourhood for short user u_s from U_{long} using exhaustive search.

```
Require: User $u_s \in U_{short}$, user $u_l \in U_{long}$, threshold $\alpha$, similarity function S()
        user_personalised_models = dict()
for $u_s \in U_{short}$ do
    training_data = $\emptyset$
    Sort $n_i \in U_{long} | S(u_s, n_i) \geq S(u_s, n_j) \forall i < j$
    training_data = $training\_data \cup data(n_1)$    ▷ Start with nearest neighbour
    Train model $\Omega(training\_data)$
    prev_error = $RMSE(\Omega, data(u_s))$
    $\Omega_{prev} = \Omega_{curr}$
    for $i | i \in 2 \dots |U_{long}| - 1$ do
        training_data = $training\_data \cup data(n_i)$
        Train model $\Omega_{curr} = \Omega(training\_data)$
        curr_error = $RMSE(\Omega_{curr}, data(u_s))$
        if curr_error $\leq$ prev_error + $\alpha$ then
            $\Omega_{prev} = \Omega_{curr}$
            prev_error = curr_error
        else
            Remove $n_i$ from training_data
            continue                                    ▷ continue with next user
    user_personalised_models[$u_s$] = $\Omega_{curr}$
return user_personalised_models
```

4.3.1. Datasets

Datasets: TYT The TYT dataset used in this study was exported on 22.01.2022, with data starting 2013.08.13, and containing EMA data until the date of export. The questions in the EMA questionnaire are already included in 2.2. There are 3269 users in the database before applying filters for minimum length, etc., of which 3161 have disclosed their gender. 2094 of the 3161 users identify as male, and the 1067 users as female. The shortest user in the dataset has contributed (unsurprisingly) only 1 day of data, while the longest contributes 1721 days of data. The mean number of days per user is 13.7, but with a standard deviation of 52.9, it is clear that the average is heavily skewed by the outlier users who contribute a lot of data. The 25th, 50th and 75th percentile for the user engagement are 1, 2, and 8 days respectively.

For the TYT dataset, the Granger causal neighbourhood method in 4.2.1 requires a minimum of 20 days of training data to be included in the analysis. A user needs at least 32 days of data to be included in the analysis. With minimum of 32 days, a maximum length of 1721 days, and a standard deviation of 156.79 days, it can be seen that the distribution of lengths of interactions is very heavily skewed, with the shortest user contributing approx. 50 times less data than the longest. The data collected spans from Aug 2013 to January 2022. The variables used in the EMA questionnaires are listed in Table 2.2. In the case of the TYT EMAs, the questions q2 to q7 from the EMAs refer to loudness, distress, mood, arousal, stress, and concentration respectively. The two binary questions q1 and q8 were not included as part of the analysis. The target variable for the forecast step is set to ‘q3’ (tinnitus distress), which is the variable of clinical interest.

Datasets: UNITI For the UNITI dataset, all users who contributed less than 30 days of data were excluded. This left us with 179 patients, and on average 71.6 days of data per person, with a mininum of 30 days, a maximum of 207 days, and a standard deviation of 30.6 days. The data spans from Apr. 2021 to Jan. 2022, the data does not allow for very long time series, although the more uniform interaction patterns may also be a consequence of closer monitoring of these patients by physicians (the app has features for the physician to monitor patient state and provide feedback).

Dataset: TYD Bulgaria The methods proposed in Section 4.2.2 used the TYD dataset from Bulgaria, with N=11 users after eliminating 5 users with less than 3 days of data. Setting the $\tau_{length} = 30$ results in 6 users in U_{long}, and 5 users in U_{short}. The figure below shows the number of days of interaction for each of the remaining 11 users (both U_{long} and U_{short}).

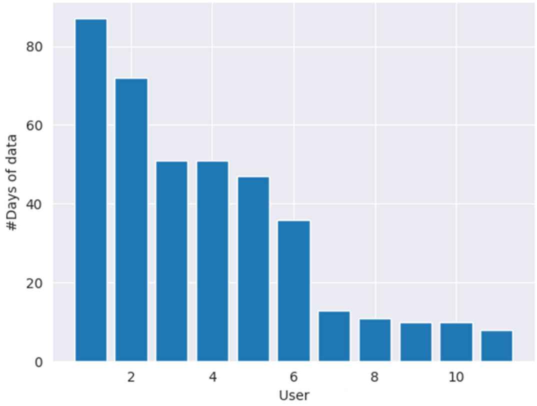


Figure 4.4.: The TYD Bulgaria dataset: Number of days of interaction for each user. Image reproduced from [126]

Datasets: TYD The methods proposed in Section 4.2.3 use two datasets from the TYD platform. Both datasets collect the same information from the same app, but implemented in different languages, since they were targeting a study for Bulgaria and Spain respectively. The main characteristics of each is dataset are outlied below:

- **Bulgaria dataset:** A total of 387 EMAs are collected for N=10 users. N=5 users with more than 30 days of data belonging to U_{long}. In this group, the average number of EOD-Questionnaire responses was 63, and avgerage number of random questionnaire responses was 141.

  N=5 users belong to U_{short}, with an average of 13 EOD questionnaire responses, and an average of 57 random questionnaire responses.

- **Spain dataset:** A total of 650 EMAs were collected for the Spain dataset from a total of 12 users. Applying a cut-off of $\tau_{length} = 30$ to split the long and short users yields N=4 users in U_{long} and N=8 users in U_{short}. The average number of responses for the EOD questionnaire is 127 for users in U_{long} and 18 in U_{short} respectively, while the number of responses for the EOD questionnaire was 362 for U_{long} and 129 for U_{short} respectively.

4.3.2. Summarising EMA data using HMMs

The experiments towards the HMM-based method described in Section 4.2.1 involve three steps:

1. Determining the optimal number of hidden states for the HMM
2. Computing the nearest neighbours of a patient using the trained HMM
3. Fitting kNN models based on the neighbourhoods derived from the representation in the step above.

Determining the number of hidden states: The first step has various quantitative and qualitative solution approaches. Model complexity metrics like the AIC, BIC and log-likelihood can be used to determine the quality of the fitted model. However, given that the user lengths vary strongly, relying on them alone can give rise to some negative outcomes.

For example, the small size of the training data available would bias us towards oversimplified solutions with models that have fewer states than capture the full patient variability. This risk is exacerbated by the fact that patterns that exist in the tinnitus variability might appear for short users, making them less learnable. On the other hand, increasing the number of states beyond a limit would increase the likelihood of creating states that are occupied by one / few users. In general, we expect this difference to be visible in the state transition matrix, where states that are 'traps' (all outward transitions are very rare) are undesirable.

Since there is no expert knowledge available on the number of states that capture the full variability among tinnitus patients, we investigate several values for the number of hidden states, ranging from 2 to 10. Additionally to monitoring the state transition matrices, we also monitor the emission matrix to monitor whether there are states that are very close to each other semantically.

Keeping all the above points in mind, we fit HMM models with states ranging from 2 to 10, and use each of the resultant models in the downstream task of training personalised models. An HMM which results in very effective personalised models suggests that the number of states used by it are useful for grouping tinnitus patients.

- AIC, BIC
- nStates, state-values, and transitions
- Discouraging models where many users are single-state users. (despite high AIC)

Converting the user to a fixed-length representation: Once the HMM model with $\mathcal{H}$ states has been trained, the training data of each patient is used to create a fixed-length representation of that patient to enable a similarity/distance computation between them. In our work, we represent the patient using an h-dimensional vector, where each dimension $h \in 1 \ldots \mathcal{H}$ represents the percentage of the user's EMA observations emitted from that state. Using a percentage instead of a counter enables computing similarities between patients that have different lengths, but has the disadvantage that the values for some states become round numbers (ie, values like 10.0%, 12.50%, 25.0%, etc.) for short sequences. Since these round numbers can result in zero distances / 100% similarity to other users. In order to apply a tie-breaker, we add a very small randomly generated number of the order 10^-6 to each state ('salting' the vector).

For a patient $e_i = \{o_1 \dots o_{|D_i|}\}$, omitting the subscript for the entity i and replacing it with the order in which the observations arrive. Each observation o_i will be generated by a state $s_1 \dots s_{\mathcal{H}}$ from the HMM. i.e., decoding the sequence $o_1 \dots o_{|D_i|}$ gives a sequence of states that generated each observation. $HMM\_decode(e_i) = HMM\_decode(\{o_1 \dots o_{|D_i|}\})$, where each observation o_i is generated from a state s_i^h, where $h \in 1 \dots \mathcal{H}$. The HMM representation of a user is computed on the basis of these states.

$$HMM\_representation(e_i) =< \frac{count(s_i^{h==1})}{|D_i|}, \frac{count(s_i^{h==2})}{|D_i|} \dots \frac{count(s_i^{h==\mathcal{H}})}{|D_i|} >$$

,$\forall i \in 1 \dots |D_i|$. Once each user is represented by the h dimensional vector, we define the user's neighbourhood as the the kNN of the user's HMM-based representation: i.e., $kNN(e_i) = kNN(HMM\_representation(e_i))$.

kNN on HMM representation Once the HMM is trained and the neighbourhood for each user is computed, the personalised model is trained with data augmentation as already described in Section 3.2.1. To train the personalised model for a user e_i, the data of the user is pooled with the data of the k neighbours of $n_i^1 \dots n_i^k$, and a model is trained on the pooled data.

The personalised model trained over the pooled data of the user and its kNN are compared against the global model that uses all users' data, and against an N=1 model that is trained on the data of the user alone. Tinnitus distress is used as the target variable ('question3' for TYT and 'cumberness' for UNITI).

Transferring the disease states from UNITI to TYT: As we saw in Section 2.2.2, the questions regarding tinnitus loudness and distress are central to more than one EMA app. We train our HMMs on the loudness and distress variables shared by more than one app, and use the HMM trained on the UNITI data to decode the sequences observed in the TYT dataset. Since the UNITI data does not get exploited in any way while training the prediction models, we use the full dataset from UNITI (including what would have been held out as test data) to train the HMM model. This model decodes the TYT sequences and the rest of the neighbourhood discovery and prediction workflows are kept identical. We investigate whether the transferred UNITI HMM is able to match the predictive quality of the neighbourhood computed by the HMM trained on the TYT data.

4.3.3. Summarising EMA data using Granger causalities

Computing similarity using Granger causal relationship matrices: The first step of the workflow is the discovery of Granger causal relationships in the EMA data. Towards this, we train the restricted and unrestricted models to find which variables Granger cause each other. The result is a matrix of Granger causalities per user. In our first experiment, we use the entire matrix in computing the user similarity. This is done through using the Jaccard similarity score, which translate intuitively to the percentage of shared Granger causalities between two users. If GC_1 and GC_2 are the Granger causalities discovered between 2 users, the Jaccard similarity between them is computed as $\frac{|GC_1 \cap GC_2|}{|GC_1 \cup GC2|}$.

For the TYT dataset, the Granger causalities were computed over all variables except q1 and q8, yielding one 6x6 binary matrix per user, with the diagonals fixed to zero, since a variable cannot Granger cause itself. For the UNITI dataset, all the EMA variables were chosen, yielding one 10x10 matrix per user. The Granger causalities are discovered using the implementation in the *statsmodels* [111] Python package. The causalities are considered with lags ranging from 1 to 7.

Restricting the similarity to Granger causal relationships towards one variable: The motivation behind this experiment is to capture whether some relationships between the variables in the EMA data are more important than others. This was done by focusing particular rows of the Granger causality matrix derived for each user, since each row captures all the variables that have a Granger causal effect on a particular EMA variable. In our experiment, we reduced the dimensionality of this sparse vector using the UMAP algorithm to 2 dimensions. Although a low number, we feel this option is justified since the original dimensionality of the vector is not very high (6 and 5 dimensions for the TYT and UNITI dataset respectively)

For the TYT dataset, the granger causalities were discovered for EMA questions q2 to q7 as target (q3 granger causes q2, etc.). 'question3' (distress) was picked as the target variable for the forecast component. Intuitively, this may be interpreted as: the causalities on which the neighbourhood is based is allowed to vary across q2 - q7, and the 'usefulness' of a neighbourhood discovered on a particular causal relationship is judged on the basis of its predictive power over next day's tinnitus distress (q3). ie, if neighbourhoods computed after setting q6 as the granger causality target result in models with better forecast accuracy, we have reason to suspect that stress and things that affect it play a more important role in symptom development than another variable, like loudness.

Similarly, for the UNITI dataset, the granger causalities were computed over the variables 'tin day caused', 'loudness caused', 'tin cumber caused', 'tin max caused', and 'cumberness caused'. This reduced set of variables were used because they are the variables that collect momentary information regarding the tinnitus. The other variables collect data about tinnitus from the whole day. The forecast target was set as 'cumberness'. All experiments involving k-Nearest Neighbours were repeated for values of $k \in 1 \ldots 50$.

Building the personalised model on the neighbourhood: Once the causalities are discovered, each patient becomes a Granger causality matrix, yielding 179 and 251 matrices for the UNITI and TYT datasets respectively. For each of the definitions of similarity discussed above, we compute the similarity between the users either as the jaccard similarity between the (flattened) matrices, or as the (inverse of the) distance between the reduced dimensional representations of the causalities discovered towards each EMA variable under consideration. The neighbourhood is computed using the kNN algorithm and our experiments vary the number of neighbours up to 50. For each user and each k, the data of the user is augmented with the data of the k

nearest neighbours and a model is learned on the pooled data. We stick to simple regression models in our case due to the small data. Tinnitus distress is the target variable ('question3' for TYT and 'cumberness' for the UNITI dataset).

4.3.4. User neighbourhoods based on their lengths of interaction

A baseline over all data: The performance of our proposed predictors that learn only on the data of the long users need to be placed into context over the degree of performance achievable over the dataset. Towards this end, we build a proof-of-concept model that follows the traditional approach of training on 75% of all users' data. The errors of this model are computed over all users, and also separately towards the users in U_{short} and U_{long}. Large differences in the prediction errors for the model towards short / long users indicate that the data of short users are unlikely to be well predicted by the data of the long users. The end-of-day "feeling in control" is the target variable.

Transfer a model trained on U_{long} to users in U_{short}: The transfer-learned predictor is one of two predictors that are combined in our workflow to create personalised predictions. The transfer-learned predictor is trained on the complete data of all users in U_{long}. The model is used to predict the data of the users in U_{short}, the data of which have never been exposed to the model. The prediction errors are computed over the predictions for each submitted EMA questionnaire, but additionally to the mean prediction errors for all U_{short}, it is necessary to also compute the user-level errors. This will help avoid being biased towards the short users that are longer. Since the data limitation is extreme, we stick to linear regression models for the prediction. The "feeling in control" question of the end of day questionnaire is the target variable, and the model is trained to predict the next day's feeling in control given the current day's EOD questionnaire.

Augmenting the transferred model with a kNN regressor: Since the model trained over the data of users in U_{long} only is making predictions for users that they have never been trained on, the predictions are augmented by a user-centred predictor that makes predictions from the earliest possible time point. The predictions of the U_{long} model are therefore augmented by the predictions from a kNN regressor trained over each user in U_{short}. The predictions from the two models are combined by weighting them with the inverse of the errors that each model achieves over the predictions for that user. We compare the mean errors achieved by each predictor, and also the development of the error as more data becomes available for the users in U_{short}.

4.3.5. Improving user neighbourhoods that exploit length of user interaction

The experiments for the methods described in Section 4.2.3 are carried out separately for the Spain and Bulgaria TYD datasets, since privacy considerations do not allow for the mixing of data from the separate locations, although the app and the backend database are shared. Although this strongly limits the amount of data for learning, these restrictions are not atypical for medical data.

Similarity function: Since the method explores building a similarity-guided personalised neighbourhood for each user in U_{short} based on discovering the best users in U_{long} to learn from, we begin with an investigation of the similarity measures, and the number of early interactions we use to measure similarity between pairs of users in U_{short} and U_{long} respectively. We explore two definitions of similarity, one that is based on submitted data, and another, that is based on the length of user interactions (we call this data and metadata-based similarity). The most suitable similarity function is used in the further experiments, and in the absence of a clear difference between the various options, we will prefer measures that use fewer data points, and result in a larger diversity among patients (since a similarity measure that yields the same result for everyone is not useful as a ranker). To summarise, the definition of similarity involves choosing between the following options:

- What to compute similarity on?:
  - Data: Similarity is computed on the answers submitted in the EMA questionnaires, over the time period under consideration. (i.e., people who answer the EMA questionnaire similarity might experience the disease similarly).
  - Metadata: Similarity is computed on the number of times a user submits an EMA response over the time period under consideration. (i.,e people who engage with the mHealth app at different levels of intensity might experience the disease similarly).
- Over which time period is similarity to be computed?
  - First N observations
  - Last N observations
- How long is the time period?: The value of N is varied between 5 to 8.

Baseline model: The performance achieved by the proposed methods are measured against a baseline model that is trained using a classical machine learning approach of using 75% of all available data for training the model. The end-of-day feeling in control was used as the target variable for both datasets.

Early Termination v/s Exhaustive Search: The early termination and exhaustive search methods fit multiple models for each user in U_{short}, as they explore the best set of users in U_{long} that predict them. The addition of users from U_{long} happens in decreasing order of the similarity measure chosen in the previous experiment. We compare the degree to which the early termination models approach the performance of the exhaustive search model, and also compare the performance of both of these methods against the baseline described above. In addition to the performance of the models, we also consider the amount of data used by each of the models to achieve their performance.

4.4. Results and discussion

The following sections describe the results for the methods presented in Sections 4.2. We begin with the degree to which the EMA data can be summarised using

HMMs (4.4.1), and Granger causalities (Section 4.4.2) for neighbourhood discovery. We follow with the methods that consider the length of user interaction instead of the EMA data itself, with results for the methods from Section 4.2.2 discussed in Section 4.4.3, and results for the methods from Section 4.2.3 discussed in 4.4.4.

4.4.1. Personalised models on EMA data summarised using HMMs

Number of HMM states: The first step in the workflow is to discover the optimal number of HMM states. We fix the training and test data for our model to 70% train and 30% test. i.e., 70% of each user's data is added to the training set and 30% belongs to the test set and is not used for learning.

We run experiments that vary the number of states between 2 and 10, and use the open source *hmmlearn* package from PyPI repository. We train GaussianHMM models without restricting the covariance matrix (i.e., covariance_type is set to 'full'). The minimum sequence length is set to 21, which is a consequence of fixing the shortest length of a user to 30 days of data, since 70% of 30 days is 21 days. The HMM is trained on the 2 variables that are most commonly present across all EMA apps of tinnitus, namely, 'loudness' and 'distress' for TYT, and 'loudness' and 'cumberness' for the UNITI app. Both variables capture the same information - the momentary loudness and distress due to tinnitus.

For each trained HMM model, we have several pieces of information to decide which model fits the data best. We have the AIC, BIC and log-likelihood values for each of the HMMs, but we also need to compare these numbers against the state transition matrix and the values of loudness and distress for each state to make a judgement call on which model to use. This process is repeated for each dataset.

Figures 4.5 and 4.6 show the plot of the model scores (AIC, BIC, and log likelihoods) for the TYT and the UNITI datasets respectively. We see that the AIC and BIC scores show steady decline with increasing number of states, while the log likelihoods increase. This is expected since the dataset as a whole is still best described by low number of states, while the explanatory power of the model for patient specific sequences increases. This result is also backed up by the fact that as the number of states increase, the number of users with 100% of their data explained by one or few states increases. Further evidence that the highest AIC model is not useful is shown in Figures 4.7 and 4.8, where it can be seen that the states capture <low loudness + low distress> and <high loudness + high distress> states respectively mediated by transition matrices that make transitions between them unlikely. The results for all the HMMs trained between 3 and 10 states is provided in Appendix A.2.1. Although it can be seen that when number of hidden states ≥ 6 there are state-pairs that are formed with very similar mean loudness and distress. We therefore apply the patient representation for all HMMs, following the expectation that a 'useful' representation of the disease (number of states that results in low error) is likely to be correct.

Performance of a personalised kNN model: The performance of a model trained on the pooled data of a user and its kNN is shown in Figures 4.9 and 4.10 for the TYT and the UNITI datasets respectively. Each line in the chart depicts the quality of a neighbourhood computed on the basis of a model trained with 2-10 states. Unfortunately, the lack of variation between the different states shows that the HMM approach does not work well. The results also do not appear to be dataset-specific.

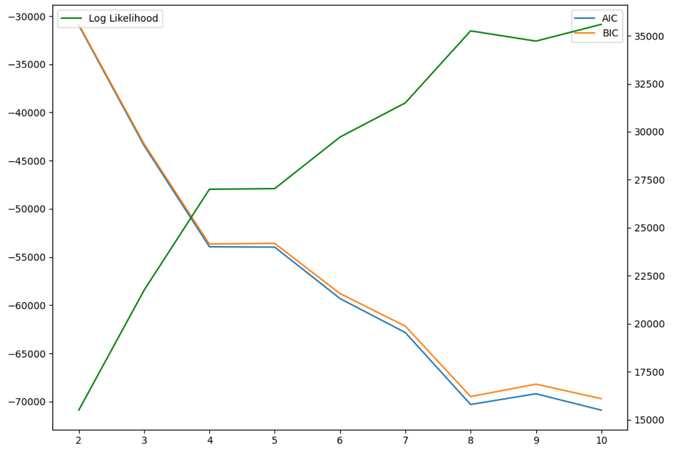


Figure 4.5.: The TYT dataset: The AIC, BIC and log likelihoods

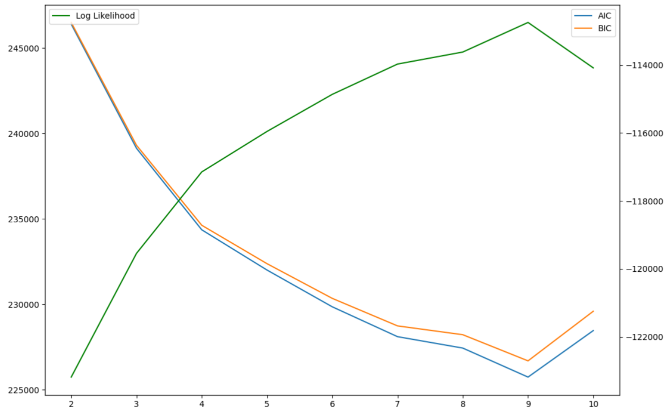


Figure 4.6.: The UNITI dataset: The AIC, BIC and log likelihoods

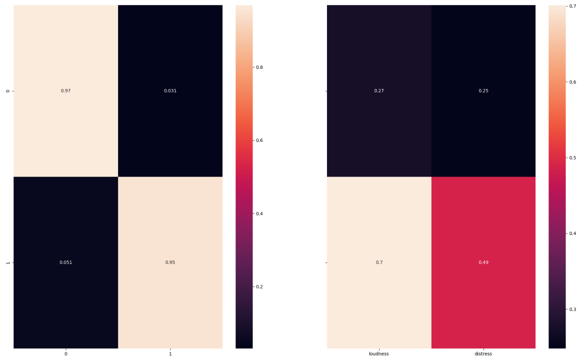


Figure 4.7.: The TYT dataset: State means and transition matrices for HMM with 2 states

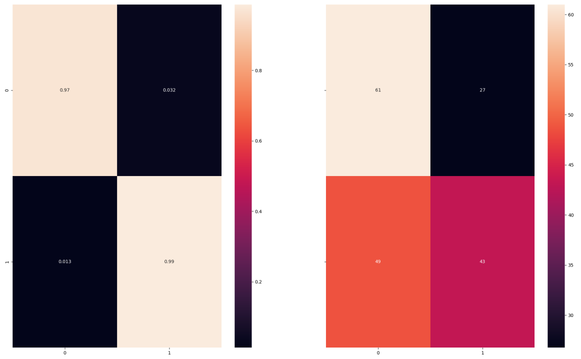


Figure 4.8.: The UNITI dataset: State means and transition matrices for HMM with 2 states

The UNITI dataset is more ‘regular’ than the TYT dataset, but apart from there being a sharper fall in the errors for the first few neighbours, the exact number of hidden states in the HMM seems to have little impact on the overall prediction quality. It does seem, however, that the increased irregularity in the TYT data does have a detrimental impact on the HMM, since the performance in the UNITI dataset is marginally better.

While the results are modest, it is important to note, however, that in both datasets, the personalised model using neighbourhoods beats the performance achieved by the global model. The performance advantage of the personalised model is small: $\approx 2.9\%$ improvement in RMSE for UNITI and $\approx 9.6\%$ improvement in RMSE for the TYT dataset. However, it is important to note that these modest performance improvements come with using just $\approx 6\%$ and $\approx 1.5\%$ of the total users in the dataset. These results suggest that there is some advantage in using HMMs, but it is the author’s opinion that the benefits are probably derived more from indirectly ‘leaking’ the average loudness and distress values rather than the HMM learning the disease dynamics.

The model trained on all available data of the UNITI dataset was also used to decode the sequences in the TYT dataset prior to the neighbourhood computation. The performance achieved by the transferred HMM model personalised with the kNN of each user is as shown in Figure 4.11. Similarly to the results for the model trained over TYT data, the transferred HMM model does not have a ‘correct’ number of states for which the error is substantially lower. However, it can also be seen that the errors are only marginally ($\approx 10\%$) higher.

4.4.2. Personalised models based on EMA data summarised using Granger causalities

Granger causalities discovered in two datasets: We run the Granger causality discovery for each patient in the EMA data for the TYT and UNITI datasets, and Figures 4.12 and 4.13 show the most frequently identified relationships in the two datasets.

The frequency of causalities observed in the TYT dataset vary within narrower ranges

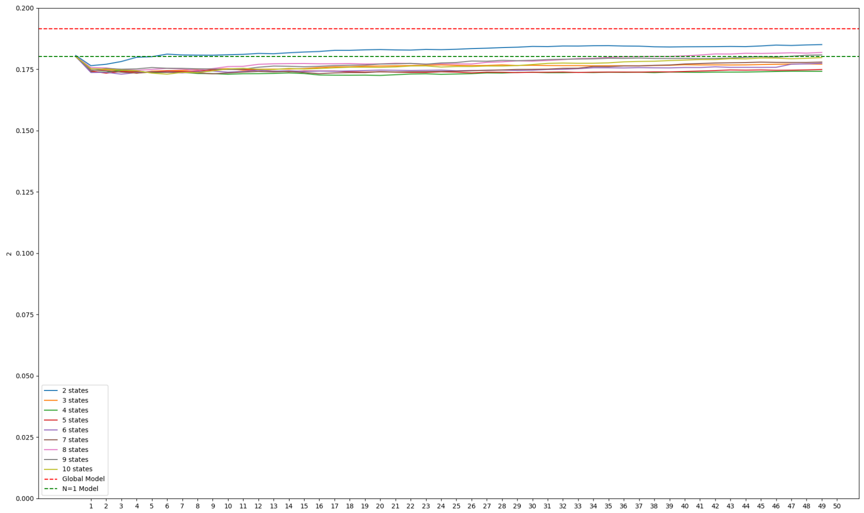


Figure 4.9.: The TYT dataset: RMSE (Y-axis) for kNN model trained on HMM-inferred neighbourhood for different values of k (X-axis)

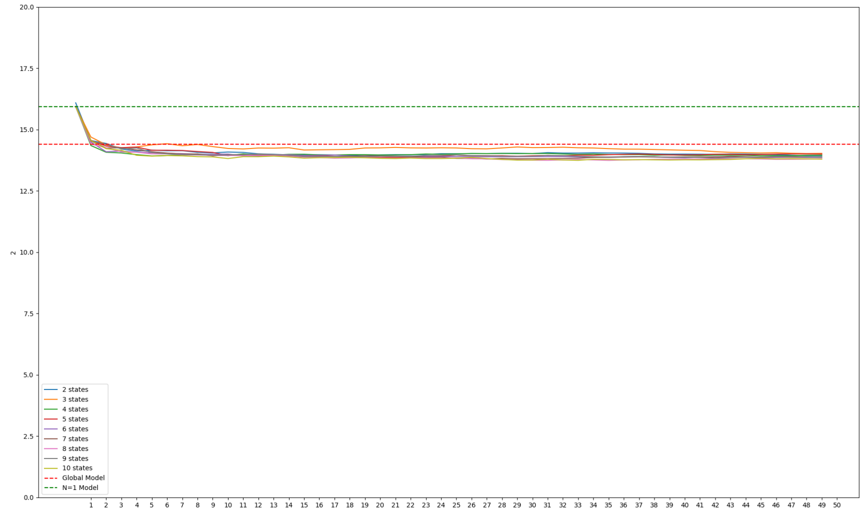


Figure 4.10.: The UNITI dataset: RMSE (Y-axis) for kNN model trained on HMM-inferred neighbourhood for different values of k (X-axis)

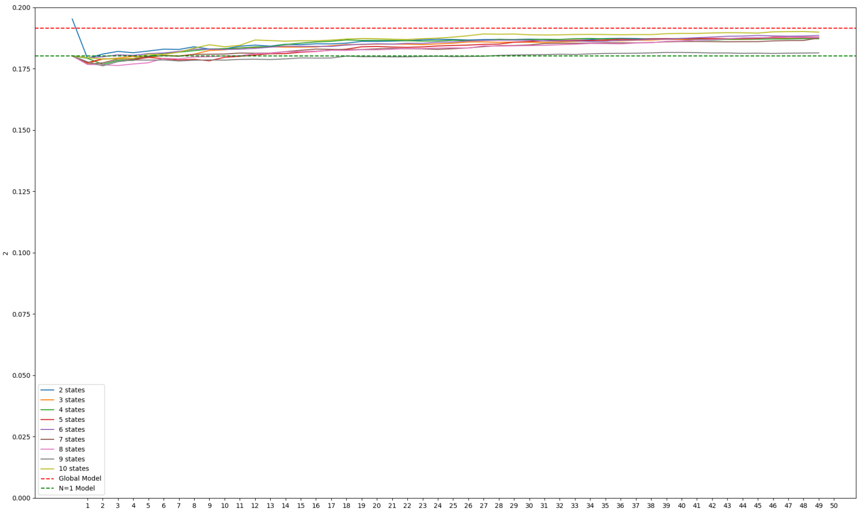


Figure 4.11.: Transferring the HMM: RMSE (Y-axis) for kNN model trained for TYT on HMM-inferred neighbourhood transferred from the UNITI dataset for different values of k (X-axis)

than those found in the UNITI dataset. This indicates either that the causalities are the results of Type I error in the TYT dataset to a greater degree than in the UNITI dataset, or that the UNITI dataset has more complex interactions between variables due to the fact that the app mixes questions about the whole day with those about the moment.

Neighbourhoods build on the full similarity matrix: Since each user is represented as a matrix of Granger causalities within their EMAs, one definition we explore for similarity is to define the similarity between two users as the jaccard similarity between the flattened Granger causality matrices. Using this definition for similarity, we apply the kNN algorithm to discover the kNN for each user, and train a data-augmented personalised model for $1 \leq k \leq 50$. The results for the TYT and the UNITI data are shown in Figure 4.14 and 4.15. The results for the kNN model are also compared against two baselines - one that is trained on all available data (Global model), and one that is trained only on the training data for that user (N=1 model). The average error for the kNN model is shown in blue, with global model shown with a red dotted line, and the N=1 model shown with the green dotted line.

For the TYT dataset, it can be seen that the N=1 model outperforms the global model trained over all users, but more importantly, it can be seen that the kNN model has a small performance advantage over the global model, which gradually diminishes as the k gets large. While the neighbourhood model is not able to meet the performance of the N=1 model, the granger causalities appear to be able to find a better neighbourhood than the global model. While the performance benefit is not encouraging, it is important to remember that the model with $k < 10$ is able to exceed the performance of the global model trained on a lot more data. This is especially relevant when considering the fact that the neighbourhood of a user is

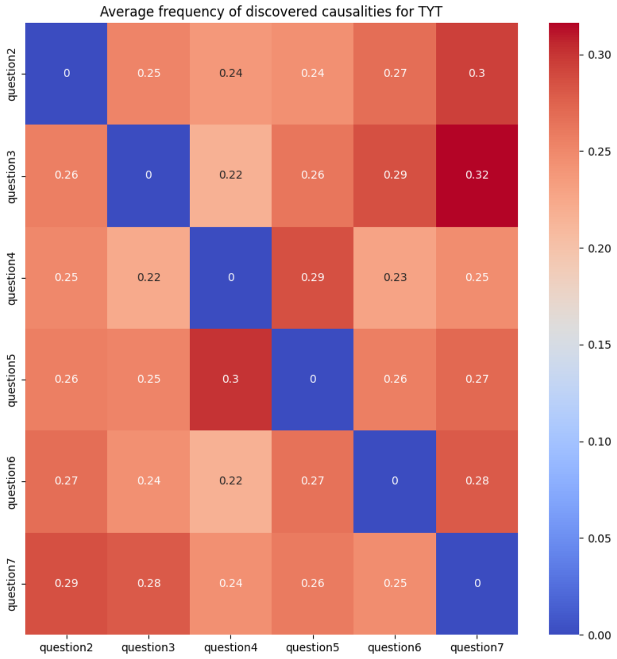


Figure 4.12.: The TYT dataset: Percentage of patients with a particular Granger-causality. (Variables on the Y axis are caused by the variables on the X axis)

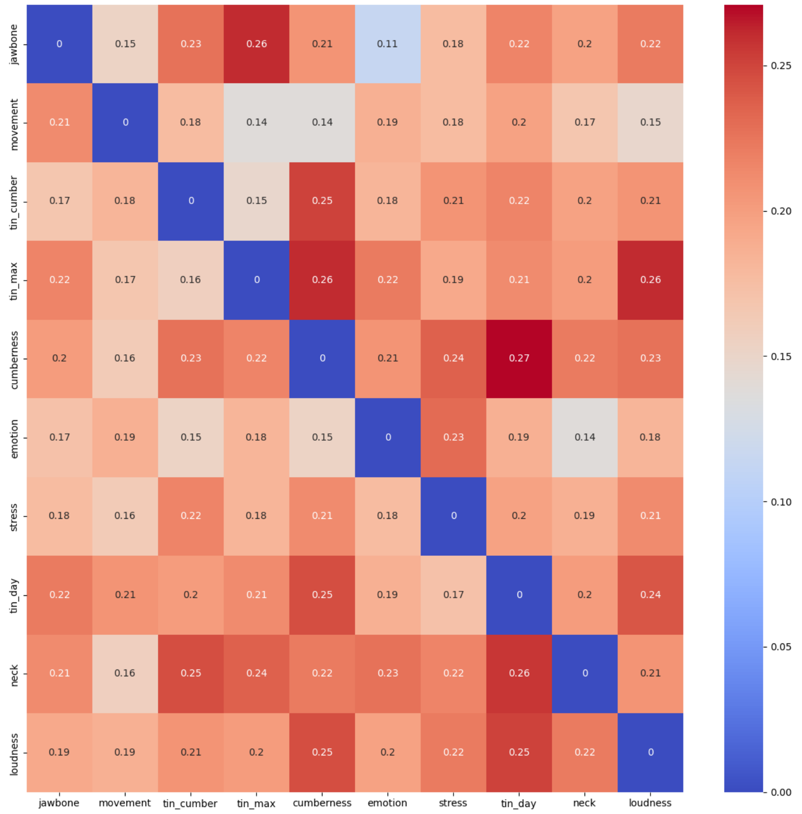


Figure 4.13.: The TYT dataset: Percentage of patients with a particular Granger-causality. (Variables on the Y axis are caused by the variables on the X axis)

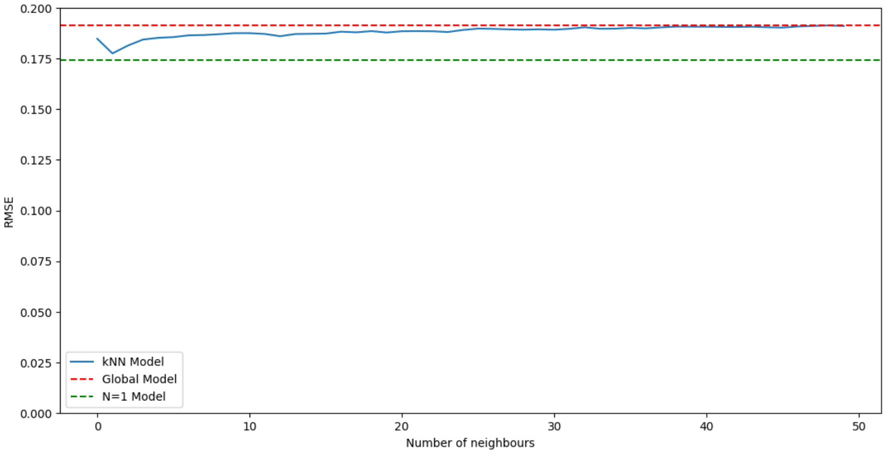


Figure 4.14.: The TYT dataset: Performance achieved by the personalised model for different sizes of k.

considered a useful output along with a predictive model for that user. The fact that the neighbourhoods are useful is also suggested by the fact that for large values of k the performance of the model approximates the performance of global model.

For the UNITI dataset, the relationship between the global and the N=1 model is reversed, with the global model performing better than the N=1 model. A deeper analysis of whether this is due to the longer sequences available for the UNITI users is necessary to understand this results. However, we also observe that the kNN model comfortably beats the performance of the N=1 for small values of k, and that for $k > 10$, the performance of the kNN model exceeds the performance of both models, although beating the global model by only a small margin. However, it needs to be noted that this performance comes with a lot less data than the global model uses. Larger values of k appear to bring no additional benefit, suggesting that smaller neighbourhoods are sufficient to make personalised predictions.

Since the personal models are worse for UNITI, we investigated whether the average-user level RMSEs were skewed towards higher values by a few long users with high prediction error. We see from Figure A.17 in the appendix that this is not the case, since the boxplot for the user-level errors shows that the global model has lower errors for more users than the personalised model.

Neighbourhoods based on Granger causalities towards specific variables: Our first investigation of the commonly occurring Granger causalities in the EMA data (see Figures 4.12 and 4.13) showed that some causalities appear more frequently than others. Since a large number of models are fit with multiple lags in the discovery of Granger causalities between each pair of variables, the likelihood of Type I errors is quite high. We therefore investigated whether focusing on causalities towards one specific variable results in better neighbourhoods than including all the variable-pairs in the computation. To accommodate for the fact that the *absence* of a commonly occurring Granger causal relationship also contains valuable information, we reduce the dimensionality of the binary Granger causality vector to 2, and use this two-dimensional vector to compute the kNN of the users. The rest of the modelling

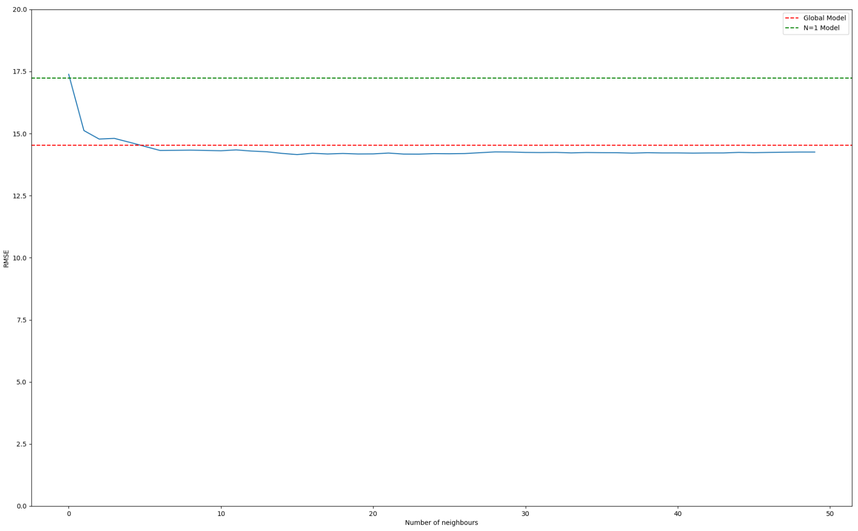


Figure 4.15.: The UNITI dataset: Performance achieved by the personalised model for different sizes of k.

process is kept as-is, with the target being tinnitus distress: ‘question3’ for the TYT dataset, and ‘cumberness’ for the UNITI dataset. For the UNITI dataset, we only investigate the relationships between the five EMAs that capture momentary information, rather than daily summaries.

Neither dataset showed improvements in the performance of the personalised neighbourhood model upon restricting the neighbourhood computation towards a single question. Figures A.18 and A.19 show the results for TYT and UNITI dataset respectively.

The TYT dataset shows an increasing error trend on the restricted neighbourhood computation as compared to the unrestricted, suggesting that valuable information is lost when focusing only on casualties towards one particular question. Apart from not affecting the averages, the decision to focus only on the interactions between some of the EMA variables also does not appear to affect the spread in the errors, since the dotted lines showing the 5th and 95th percentile of the errors is not affected by the gc_{target}.

4.4.3. Exploiting lengths of user interaction to build personalised neighbourhoods

Baseline model: The first experiment establishes the degree to which a model trained over the data of all users predicts the observations of the users in U_{short} and U_{long} respectively. The performance (as measured by the MAE) of the global model over the predictions of the long and short users are shown in Figure 4.16. The fact that the performance of the global model over the short users and long users is not too different suggests that the data of the long users can indeed be used to predict the data of short users (i.e., they are not so different that a model trained on one cannot be used for the other). However, it is important to remember that these

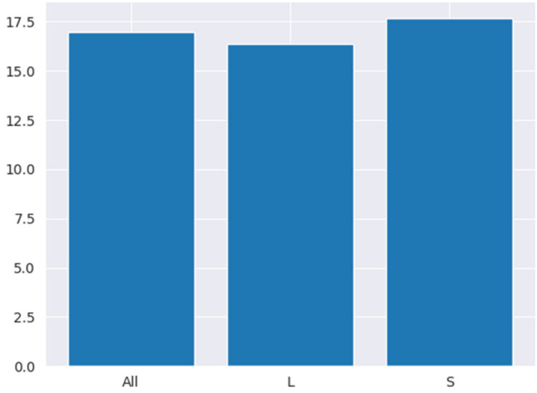


Figure 4.16.: The TYD-Bulgaria dataset: Performance of the global model on the predictions for all users, U_{long} (L), U_{short} (S) respectively. (Image from [126]).

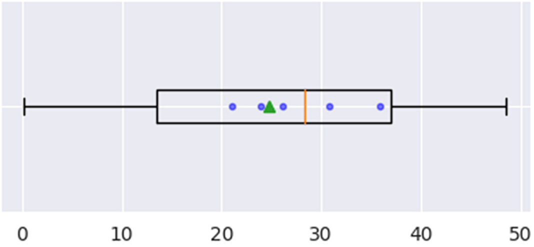


Figure 4.17.: The TYD-Bulgaria dataset: Performance of the global model on the predictions for all users, U_{long} (L), U_{short} (S) respectively. (Image from [126]).

errors are not fully reliable, since the amount of holdout data for the short users is also very small, since we fixed 25% of the data for testing. However, the error for users in U_{short} serve as a lower bound for what errors can be achieved by the model transferred from U_{long}.

Performance of the model transferred from U_{long}: The errors in Figure 4.16 establish a lower bound on the errors that are achievable for a model transferred from U_{long} to U_{short}. In the case of the transferred model trained on U_{long} to predict for users in U_{short}, we train the model on all users with more than 30 days of data. Figure 4.17 shows a boxplot of the prediction errors for the transferred model.

Using the model transferred from U_{long} increases the MAE from 24.76 compared to 17.2 when the data of all users is used. It is to be noted that although this error is higher, the predictions are now being made for patients that the model was never trained on. The error measure is also unfortunately biased towards the longer users in U_{short}, since they have more sessions to predict. The boxplot show the errors for each prediction for each session for all users in U_{short}, but we also track the errors of the transferred model at the user level. The MAE achieved by the model for each individual user in U_{short} is shown by the blue dots in the box plot. We see that the MAEs range from 21 to 35, and the fact that the mean is 24.76 shows that the longer users are indeed better predicted by the transferred model than the shorter ones.

Personalising the transferred model with incrementally available data from U_{short}: In this workflow, we add a kNN regressor over the past history of each user in U_{short}, which makes predictions along with the transferred model from U_{long}. While the transferred model is not personalised, the U_{short} model is fully personalised and relies exclusively on the user's own past to generate predictions. The training data for the kNN regressor is the user's history up until that point, and the target is the "feeling in control" at the next time point (this means, of course, that the last session in every user's sequence is not usable since the value of the target variable is not known). In our experiments, we set the kNN regressor with $k = 2$ (the minimum possible value), so that the smallest number of observations are lost to training, and the maximum number of predictions are available for evaluation. Please note that the kNN regressor proposed here works differently to the kNN approach used in the previous methods discussed in this work. While the previous methods applied the kNN at the user level to find a list of 'best users to learn from', the kNN regressor in this case is applied at the session level, and the use case is closer to "how did the user behave the last time their data looked like the current session?".

For each item in the sequence, a prediction is possible from the U_{long} model, as well as the user's own kNN regressor. As with any ensemble, several approaches can be used to combine the two predictions. We propose that the two predictions are combined using the inverse of the user-level errors accumulated by the two models on the observations so far. i.e., instead of fixing the weight of U_{long} model and the kNN regressor, we track the performance of each of these predictors at the user level, and combine the predictions weighted on the inverse of the errors they achieve. Figure 4.18 shows the results achieved by the transferred U_{long} model, the kNN model, and the model that combines the predictions from each using the weighting scheme described above. It can be seen that although the user-centric kNN and the transferred model have similar errors, the kNN workflow has a lower median MAE, even if it comes at the cost of increased number of large errors. The fact that the means and the medians of the two models deflect in opposite directions by roughly the same magnitude suggest that the two models both make large errors, but with opposite tendency. The error-weighted combination of the two predictors leads to a more balanced set of predictions (with mean and median MAE ≈ 20), with a comparable mean MAE to the kNN regressor, while also avoiding the extreme errors of both models.

In addition to the numbers from the combined weights, we also investigate how the error of the predictors develop over time for the users in U_{short}. Figure 4.19 shows how the mean MAE over all users develops over time as the sequence length of the users in U_{short} increases. Since we set $k = 2$ for our kNN regressor, the first two observations are predicted purely by the linear regressor trained over all U_{long} , and the mean errors for the subsequent predictions are plotted separately for the linear regressor (U_{long} model), the user centric kNN regressor, and the error-weighted combination of the two. It can be seen that the weighted combination (green line) follows closely the development of the kNN regressor centred on the user's own data. This suggests that the predictions generated by th user centric regressor get higher weight (due to lower error). Although the small size of the dataset warns against generalisations, the downward trend in the kNN regressor's errors is also encouraging, since it suggests that the longer users in U_{short} are developing in a way that makes them more predictable given their own data.

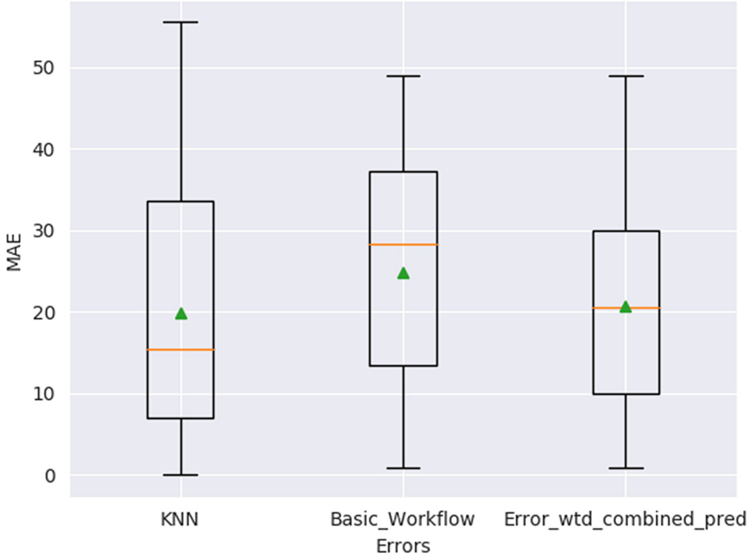


Figure 4.18.: Performance achieved by the transferred U_{long} model (basic workflow), the user-centric kNN regressor, and the model that combines the predictions of both based on user error. (Image from [126]).

4.4.4. Personalising neighbourhood sizes built on user interaction length

This section discusses the results for the experiments described in Section 4.2.3. We build upon the results from Section 4.4.3, and investigate whether all users in U_{long} are equally valuable for the users in U_{short} when building personalised predictors. To this end, we use a minimal subset of the dynamic data in order to measure the similarity between pairs of users in (U_{long}, U_{short}), and explore the users in U_{long} incrementally in decreasing order of similarity as long as the resultant model achieved good performance over the user.

To reiterate the list presented in Section 4.3.5, we investigate 3 aspects of the similarity - whether it is computed on the data or on the number of data points contributed by the user, whether we use the first N or last N observations from the user, and how many days of data do we consider for the computation. We do not increase the N beyond 8 in interest of preserving as much data for model training as possible. The full results for all options for the Bulgaria dataset are shown in the appendix under Figure A.20. For the rest of this work, we choose data-based similarity, computed over the first N=5 days on the data in the EMA responses. The reasons for these choices are: (a) The plots show that the decisions do not have a huge impact on the similarities computed by the methods. (b) we preferred the decisions that caused a larger interquartile range in the boxplots, with the motivation that a similarity measure that finds everyone highly similar is of no use in ranking, and (c) the first-N days of all users are directly comparable since they are similarly mature within the system, while the last N days of different users might come from very different levels of maturity. A snapshot for the Bulgaria dataset is shown in Figure 4.20.

Exhaustive Search v/s Early Termination: Comparison with Baseline Our two proposed methods, the exhaustive search and the early termination are both compared against the classical machine learning baseline of training on 75% of all available

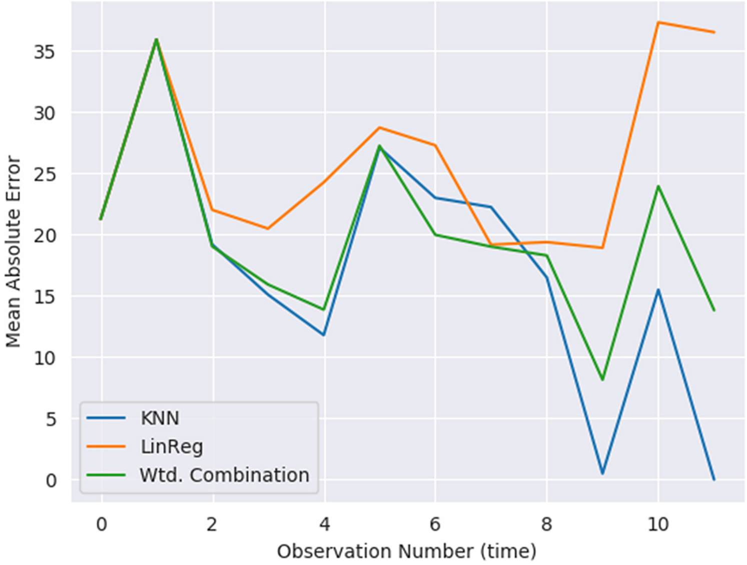


Figure 4.19.: Comparing the errors of the transferred U_{long} linear regressor, the user centric kNN regressor, and the error-weighted combination of the two. (Image from [126])

data from all users. The errors achieved by the models are shown in Table 4.2, for using the data based similarity measure with N=5. Table A.2 in the appendix shows the counterpart for metadata-based similarity for N=5. We see that the choice of similarity value that guides the neighbourhood exploration process does not impact final model performance strongly.

Dataset	Baseline Model (RMSE)	Exhaustive Search (RMSE)	Early Termination (RMSE)
BG	24.300	22.524	23.700
ES	23.165	14.675	14.675

Table 4.2.: TYD Bulgaria (BG) and TYD Spain (ES) datasets: Errors achieved by the exhaustive search and early termination method compared to the baseline model. Similarity measure used: First-5 data points

For both Bulgaria and Spain, the exhaustive search method yields personalised models with better performance than using all available data. This is a remarkable result, since the models are learning on subsets of already small data to achieve, in the case of the TYD Bulgaria dataset, comparable results, and in the case of the TYD Spain dataset, much better performance. The early termination method also performs comparably to the baseline model for the Bulgaria dataset, and much better than it for Spain. However, the performance of the model needs to be investigated a bit further, since both the exhaustive search and the early termination models can converge on the entire dataset for learning (since it is logical to assume that more data is better). To investigate the neighbourhoods discovered by the model, we need to look closer at the results to the user-level models, their errors, and the amount of data they are trained over.

The user-level analysis of the errors and data used by the baseline (B), Exhaustive Search (ES), and Early Termination (ET) models are shown in Table 4.3 for the

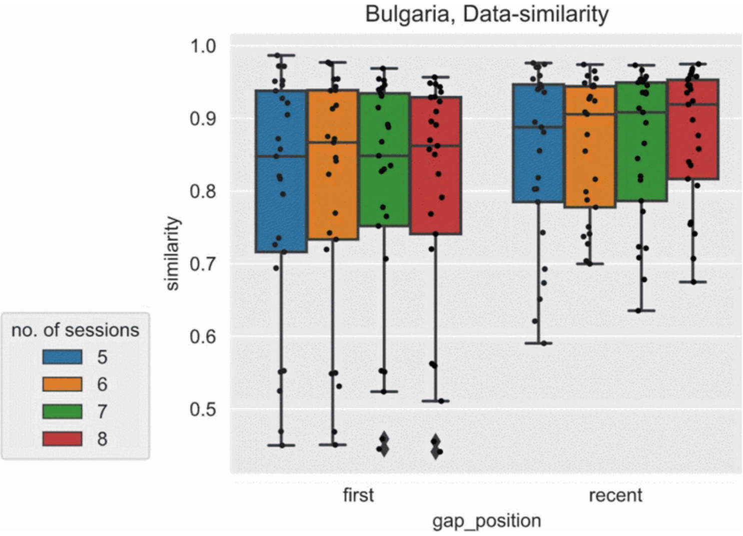


Figure 4.20.: Boxplot for computed similarities (Bulgaria dataset): Final choice of data-based + N=5

similarity based on EMA data (for results for metadata-guided similarity, please see Table A.3 in the appendix). We see that for 3/5 users in Bularia, and for 7/8 users in the TYD Spain dataset, the exhaustive search model exceeds the performance of the baseline model, while using only 57% and 30% of the data respectively. It can also be seen that the performance difference between the early termination and exhaustive search models are very close for both datasets. It can be seen that for 3/5 users in Bulgaria, and for 7/8 users in Spain, the early termination method found the same neighbourhood as the exhaustive search. Coupled with the fact that the exhaustive search converges on a neighbourhood that is not the entire dataset used by the baseline model, this suggests that the similarity guided framework does indeed find suitable neighbours to learn on.

4.5. Conclusions

In Section 3.2, we introduced approaches that exploit the static data of entities to build neighbourhoods that can be exploited to create personalised predictors. In Section 4.2 , we explore another approach towards the same goal by exploiting the dynamic data of the entities. Towards this end, we investigated the approaches presented in Section 4.2.1, which summarise the data in the dynamic domain using either HMMs or Granger causalities, while the approaches in Section 4.2.2 and 4.2.3 explore exploiting the length of user interactions to find groups of users that can be used in the training of personalised predictors.

4.5.1. RQ2.0.1 Exploiting similarity in dynamic data using HMMs:

In this approach (see 4.2.1), we propose training HMMs on the pooled EMA data of all users to learn the 'disease dynamics', and then represent each patient's dynamic data as a 'summary' as described by the HMM states they visit.

- We trained the HMMs on data from two EMA apps and investigated the states and transition matrices to find that model quality metrics like AIC and BICs

Dataset	User_ID	B	ES	ET	|B|	|ES|	|ET|
		(RMSE)	(RMSE)	(RMSE)			
BG	1	36.26	24.69	31.58	316	145	100
	2	30.07	28.38	28.09		267	216
	3	19.38	20.15	20.15		171	171
	4	23.24	25.39	25.39		171	171
	5	23.47	22.01	22.01		149	149
ES	1	32.14	30.49	30.49	505	89	89
	2	42.82	35.93	35.93		89	89
	3	30.36	7.11	7.11		89	89
	4	23.82	11.3	11.3		84	84
	5	18.38	6.78	6.78		280	280
	6	23.92	10.81	10.81		195	195
	7	21.97	14.52	14.52		195	195
	8	11.77	12.23	12.23		195	195

Table 4.3.: User-level RMSEs and training data size for the exhaustive search, early termination, and baseline models

have difficulties with balancing quality and capturing idiosyncrasies of users.

- We proposed a method to represent EMA sequences of unequal lengths as fixed-length vectors that summarise the EMA sequences. This is done by summarising a patient as the percentage of time they spend in each hidden state.

- Since investigating the model's parameters did not help find the 'correct' number of HMM states for both datasets, we investigated the efficacy of all the trained HMMs (between 2 and 10 states) in discovering personalised neighbourhoods - the rationale is that any number of states h that results in predictive neighbourhoods for the personalised models is capturing underlying hidden states that accurately represent the disease. However, we found that the performance of the personalised predictors does not depend on the number of hidden states h.

- Although the number of states was not found to affect performance too strongly, in both datasets, the HMM-based neighbourhood resulted in better predictions than the global models with small neighbourhoods:

  – UNITI Dataset: the 10-state summary resulted in the models with the best performance (RMSE of 13.9), beating the global model (RMSE of 14.4) by 3.5%. It is, however, the opinion of the author that this representation is probably overfit. The model with $h = 6$ shows the sharpest declines in error with increasing neighbourhoods compared to all other models, and even with only the data of the 10 nearest neighbours, the model achieves an RMSE of 13.999 v/s the global model's 14.403 (2.9% improvement).

  – TYT Dataset: Like the UNITI dataset, the charts show that the performance of all HMM-based neighbourhoods were comparable, but the TYT model with 4 states showed the best performance with $k = 21$, with an RMSE of 0.17248, compared to the global model error of 0.19145. i.e., the performance is 9.9% better than the global model while using the data of only 6.6% of the users. Other models (6-state HMM) achieved an RMSE of 0.173, i.e., a 9.6% improvement using only the data of 5 nearest neighbours (1.5% of the users!).

- Our investigations towards transferring the knowledge of tinnitus dynamics from one app into another showed that an HMM trained on the UNITI dataset can (although to a lesser extent), inform about patient neighbourhoods in the TYT dataset. Small neighbourhoods exceeded the quality of both the N=1 and the global model, with larger neighbourhoods adding less related users, increasing the errors to approach the global model. Even modest neighbourhoods achieved performances of RMSE = 0.177, an ≈ 7.5% improvement.
- Further investigations are necessary to determine if the HMMs are indeed a useful part of the workflow. The opinion of a clinician on the states and their transitions can help fix the number of hidden states, and the further investigations can also explore the possibility to separate out users that visit similar states but in different order. The HMM can also be trained on data preprocessed to capture deviations in symptoms instead of symptom intensity itself (this method was not explored because this makes the HMM states much harder to explain).

4.5.2. RQ2.0.2: Exploiting similarity in dynamic data using Granger causalities:

In this approach (presented in Section 4.2.1), we proposed a method to discover neighbours for each user by summarising their unequal-length EMA sequences as a matrix of Granger causal relationships within each sequence. Since the space of possible Granger causalities in the EMA time series is $n(n-1)$ for an EMA time series of dimensionality n, we can represent each variable length sequence with its fixed-length Granger causality matrix. We explored two methods to use the matrix to discover neighbourhoods for training personalised predictors, and tested the approach on the TYT and UNITI datasets.

- The Granger causality methods discovered some causalities occur more frequently than others in the EMA datasets. This is intuitively true, since some variables preferentially interact with the future values of others.
- Since the output of the Granger causality discoverer is a binary matrix of the discovered causalities between the variables, we propose computing the similarity between users using the jaccard coefficient in their 'flattened' matrices. This translates to the similarity between a pair of users being defined as the percentage of total causalities that they share. For the TYT dataset, a personalised model built on the kNN of a user as defined by this similarity was seen to outperform the global model built over all users' data, but the error of this model increased to approach the global model for larger values of k. This suggests that the nearest neighbours are indeed identified correctly, but the large number of tests for Granger causalities introduces too much noise into the system. This problem is further exacerbated by the fact that the TYT dataset has more sparse data. For the UNITI dataset, the error shows a downward and flattening trend, with even small neighbourhoods exceeding the performance of the global model by margins similar to the HMM use case. This is also in line with expectations since the UNITI dataset has more regular data.

- Combining the results form the two datasets, we see that for more regular EMA data, Granger causalities may indeed approximate / improve on the global model with far less data (less than 10 neighbours), while providing out-of-the-box GDPR-compliance (the effects of data removal are localised to few users).
- Although the results are shown for various values of k, each user has different causalities over which neighbourhoods are computed. The neighbourhoods of a user can serve as a valuable starting point for clinician when assessing the importance of the symptoms and their relationship with tinnitus distress.
- The idea of restricting the similarity computation towards individual rows in the Granger causality matrix showed that causalities towards no single EMA question dominated the others in terms of model performance (i.e., relationships toward 'distress' were not more important than relationships toward 'exhaustion', for example). This shows that neighbourhoods need to consider all causalities when assessing similarity. It is possible that more complex methods that capture (non-linear) relationships in the causality matrix - especially those that can track the importance of the *absence* of a commonly occurring relationship, can extract more information from the EMA data. Our attempt to use UMAP dimensionality reduction did not succeed.
- More than one method may be simultaneously applied to discover neighbourhoods. The use of Granger causality may be applied alongside other methods (for example, the HMM method above) to find users similar that benefit the model.

4.5.3. RQ2.1-2.2: Exploiting user interaction length to train personalised models

The approaches presented in Sections 4.2.2 and 4.2.3 consider learning for users with very different lengths of interaction. As the problem of 'short' users is a common one in EMA datasets with self-reported questionnaires, models that are able to make predictions for users that are typically excluded from most machine learning methods due to insufficient data is a challenge. In order to investigate if predictions are possible, it is necessary to first examine whether the data of 'long' users does indeed predict short ones, and to what degree these predictions can be augmented for personalised predictions. Our approach is tested on two TYD datasets, one from Spain and the other Bulgaria. The two datasets are dissimilar in which type of diabetes is more common (Type 2 diabetes is more common in the Bulgaria dataset), and data privacy restrictions do not allow combining the datasets for learning.

Our main findings are summarised below:

- We explored the degree to which the 'long' users (that contribute more than 30 days of data) predict the behaviour of 'short' users. We see that the model transferred from long users shows higher error than the baseline model trained over all users' data.
- We found that the predictions of the transferred model learned over the long users can be augmented by a second personalised predictor that sees only the history of the short user. We build a kNN regressor for each user, and found that using an error-weighted combination to combine the predictions

of each model balances out the extreme errors of each model, with the mean and median prediction errors coming closer. We see therefore that the length of the user interaction does help increase predictability at the user level, and that the differences between the long and short users can be bridged using the error-weighted combination of the two models.

- We did not test for larger values of k since our datasets were too small, but further work can explore the effect of this approach on other datasets less constrained by user length. However, our goal of making predictions early in the user's history prefers smaller values of k. Future work can also integrate other sources of data / questionnaires, since our current work has only investigated the EOD questionnaire.
- Next, we explored the possibility that some long users might be more 'useful' in terms of final prediction quality than others. Towards this end, we propose two iterative neighbourhood discovery methods that searches the space of available long users in decreasing order of similarity. We found that the exact definition of similarity (based on data / user interaction intensity) does not strongly affect model performance. This suggests that while the incremental neighbour discovery method does have benefits, the similarity between users itself is not well captured by the data or the metadata of the user.
- We found that the for a majority of the users (especially in the TYD Spain dataset), the greedy search of most similar neighbours yields the same user neighbourhood as the exhaustive search method that does not terminate the loop on encountering the first user that increases error. This finding is at odds with the insensitivity of the methods to the exact definition of similarity, and needs to be replicated in larger datasets. We also see that our early termination and exhaustive search methods both create models that outperform the baseline model that is trained on all users' data, performance gains of 2.5%-36% while using only 40%-69% of the training data. This fact is to be seen in the context that the user's own data is not used to train the models, but only to find the best neighbours.
- Since the neighbourhood discovery methods work on predicting users that are not included in the training data, extensions to our method can easily make it more practical - i.e., the large number of models trained during the neighbourhood discovery process can be 'memoised' so that a newly joining user can easily poll the existing models for prediction quality without the need to retrain. However, the overhead needs to be assessed for efficacy since our datasets are small and the linear models train quickly even on commodity hardware.
- Like with our other methods that deliver a neighbourhood for each user, the neighbourhood itself might contain information that enables the physician to conduct further analyses.
- The proposed methods are not validated on larger datasets. The small size of the two TYD datasets is the most significant threat to validity. Although the results have replicated for 2 datasets, they are both small.

The methods proposed in this chapter build upon the results that static data can be used to discover neighbourhoods to create personalised predictors, but the neighbourhoods based on static data are not much better than choosing neighbours randomly. The methods proposed in this chapter, therefore, attempt to use the dynamic data in two ways: (a) one that summarises the variable-length sequences based on their temporal properties, and (b) one that uses interaction length to split users into two groups, with the aim of creating predictors for the 'short' users on the basis of their 'long' counterparts.

Our accumulated results suggest that building personalised predictors on the basis of neighbourhoods derived from static data and the personalised neighbourhoods derived from dynamic data both do not adequately capture the complex underlying similarities required to augment kNN-based personalised models. This is because of the unreasonable effectiveness of the kRE baseline, the insensitivity of the results to the number of HMM states and Granger causality targets, and the insensitivity of the neighbourhood discovery process to the similarity function. In the next chapter, we consider the logical extension that discovers neighbourhoods iteratively using models validated over the dynamic data.

5. RQ3: From neighbours to useful neighbours: Towards a supervised approach to modelling similarity

This chapter is based on the outputs from the following papers:

[128] Unnikrishnan, Vishnu et al. “A Similarity-Guided Framework for Error-Driven Discovery of Patient Neighbourhoods in EMA Data”. In: Advances in Intelligent Data Analysis XXI. Cham: Springer Nature Switzerland, 2023, pp. 459–471

- The work presented in 5.2.2 that removes user ordering from the neighbourhood discovery process is as-yet unpublished work.

5.1. Motivation and comparison to related work

Our proposed methods in Section 4.2 explored various aspects of including the dynamic data to help create neighbourhoods. Both the methods exploring summarising the EMA sequences were unsatisfactory in terms of model performance. Although they approximated the performance of the global model with a fraction of the data (<10% of the users), the Granger causality method likely suffers from negative effects of irregularity in the collected sequences. This can be seen by the increasing error in the kNN model, as well as the rather uniform distribution of discovered causalities. The HMM model for EMA summarisation, similarly, has a tendency to be influenced by the longest users. Both methods show worse performance on the TYT dataset, suggesting that the more sparse the EMA data becomes, the less reliable the methods become.

In addition to the two EMA summarisation approaches, we also saw that the iterative user neighbourhood discovery function was quite insensitive to whether the data or the metadata of the user was used while discovering its neighbourhood. Intuitively, this suggests that our similarity function does not adequately capture the true underlying similarity between the users. Since our results also show that ‘long’ users alone do not predict ‘short’ users, we explore the possibility of using the dynamic data from each user as a validation set to discover user similarity.

Towards this end, we propose two methods that discover the optimal neighbourhood by iteratively expanding the neighbourhood they are trained on as long as the performance of the personalised model does not deteriorate. It is our hope that the iterative neighbourhood discovery brings two benefits:

- Our results from Section 4.4.4 show that not just the neighbourhood, the neighbourhood size can also be personalised. Apart from the obvious benefits of personalisation, we also see this as natural for EMA-like datasets since each additional neighbour brings different amounts of training data to the model.

- The idea of using the performance of the model instead of the user characteristics might help avoid the complexity of similarity computation, and capture neighbourhoods that accommodate the complex heterogeneous data.

Like for the methods in Section 4.2, we focus on creating predictions for the target variable at time $t + 1$ given all other variables at time t. Our proposed methods are described in 5.2, but we will first begin with a discussion of some related literature.

The main link to algorithms that self-select their learning data comes from semi-supervised learning and active learning. Active learning and its basics are introduced in the mature but comprehensive survey of Burr Settles [113]. The main idea behind active learning is that the learning algorithm itself chooses the data that it is trained on, with the additional assumption that unlabelled data is plentiful and labelled data is scarce. Most work focuses on 'pool based' active learning scenarios, where there is a small pool of labelled data L, and the goal of the active learner is to exploit the labelled data as well as the unlabelled data in order to find the best instances from the unlabelled pool U that should be added to the labelled pool L. The labelling process is assumed to be carried out by an 'omniscient oracle', which is typically a human in the loop. The fact that the labelling process involves a human places strong emphasis on active learning algorithms to request as few labels as possible, and the goal is to achieve the best performing model for the data in U and L.

Several approaches exist for active learning - some examples are *stream based selective sampling* [3], which decides on the fly whether to query the currently processed instance for a label based on some criterion. If the data is small enough to be stored, pool based approaches in those explored in [114] typically pick instances to pick greedily, according to an informativeness measure. The measures can pick instances based on model uncertainty, based on disagreements between ensemble members, the amount of change to the model, the drop in the model error, or decrease in model variance after incorporating the new information. Of all the methods presented in the survey, our proposed method is closest to expected error reduction, since we use the validation data of each user as a proxy for how well the model would perform upon the addition of the next user. However, unlike in most methods discussed in the literature where algorithms choose their own training data, we do not pick individual data points, but rather entire sets of EMA data based on the user in question.

More recent interest in active learning focuses on combining active learning methods with deep learning. According to a survey on the topic [95], the ability of deep learning to extract features can be combined with active learning's ability to query instances iteratively to improve the model. The methods have been successfully tested in data from domains ranging from image recognition to text classification. The methods presented in the survey detail how many sampling strategies can be used in a deep learning context. Since our data is not large enough for deep neural networks, and since there is no need for a human labeller of instances, we do not explore this topic further. The interested reader can refer to the book [68] for a comprehensive overview of the main methods and where they can be applied.

Semi-supervised learning is another body of work to which we relate tangentially. As explained in the survey [129], semi-supervised methods aim to combine existing unlabelled data and labeled data in order create better classifiers. From the taxonomy presented by Van Engelen & Hoos, the “wrapper” methods are closest to our work, where out-of-the-box classifiers are applied on the available training data, and are then iteratively expanded with labels of instances that are predicted with high confidence by the model. This is analogous to the case where an out of the box linear model is trained on the data that we expect reduces model error. This is called “self training”, and although a mature idea [139], continues to find use in more recent times [120]. The authors warn that self training methods are sensitive to the order in which training data is added, since the ranking of instances determines the model itself, and the confidence with which it makes predictions. This warning applies to our workflow as well, but we hope the impact is slightly lessened by our workflow selecting the next *user* to be added, instead of simply the next instance.

Another family of methods of tangential relevance is automatic feature selection algorithms. Our proposed method is similar to the wrapper based methods discussed in the survey [23]. The exponential nature of the problem of discovering the ideal subset of features is obvious, and it is clear that performing neighbourhood selection also presents an exponential search space of all possible permutations for all users for all neighbourhood sizes. For example, the branch and bound algorithm suggested in [71] explores every possible subset, and is clear to not scale well with increase in features. More recent methods attempt to overcome these limitations by searching for neighbourhoods using a heuristic to rank possible feature subsets, for example, with genetic algorithms [45] that are combined with dynamic programming to minimise fitness evaluations. The closest cousin of our method from the feature selection world is, however, the sequential feature selector. While forward feature selectors start with a single feature and add the most useful features greedily, backward sequential feature selectors start with all features and iteratively eliminate features that detract from final model performance. Our proposed method is analogous to a forward sequential feature selector, except that at each stage, new rows of training data are added to the model (the data of the next most useful user) instead of new columns.

5.2. Towards RQ3: To what extent can the notion of similarity be supervised?

As explained in Section 5.1, our goal is to derive neighbourhoods for each user in the dataset without fixing the neighbourhood size, as kNN methods do. Towards this end, we need to assess the degree to which each additional neighbour contributes to the model, and stop growing neighbourhoods at the right moment. Therefore, our main question is:

RQ3 To what extent can the dynamic data of the user be used to assess the quality of a discovered neighbourhood?

To answer this question, we propose a method with two components:

- An iterative similarity-guided neighbourhood discovery component to create personalised neighbourhoods using dynamic data for validation.
- An iterative neighbourhood discovery component to create personalised predictors that is insensitive to user ordering.

The first component creates a personalised model for each user by searching through a list of potential neighbours in decreasing order of similarity. This workflow is very similar to that introduced in Section 4.2, except that it does not restrict itself to finding neighbourhoods for long users only. By generalising this framework to all users, we build not only personalised predictors, but also deliver the user neighbourhood, something that we suspect will be valuable for future analyses. Of course, the other benefits of our workflow like its out-of-the-box support for data privacy and data removal requests is not to be overlooked.

Since the above method is sensitive to the order in which users are searched for addition into the neighbourhood, it has a tendency to get stuck in local minima. Therefore, the second method serves as a baseline against which the performance of the first can be compared. Since a full exploration of every possible neighbourhood of every possible size is combinatorially prohibitive, we still need to apply an ordering to the way users are added into a neighbourhood, but this method, apart from providing the best neighbourhood, also serves as a way to verify the degree to which guiding the neighbourhood using similarity results in stable neighbourhoods that can be trusted by an expert.

5.2.1. Error-driven neighbourhood discovery framework

Figure 5.1 shows an overview of our proposed neighbourhood discovery framework. The workflow produces the personalised model for each user in the dataset, and this is done by iteratively adding the next-most-similar user to the training data of the personalised model, as long as the model performance does not deteriorate. It is clear that there is a danger of overfitting the models, so we split each user into three parts, with 49% of the data used for training, 21% for validation, and 30% for testing (i.e., the full dataset is split 70:30, into train+val:test, and then the non-test data is split again by 70:30 to get the train and validation data). It is of course not permitted to split the data into train and test randomly, so the user-level split is done temporally, with the first 49% of the data for train, the next 21% for validation, the last part for testing, etc.

Guiding the neighbourhood selection process and avoiding local minima: As explained in Section 5.1, it is computationally prohibitive to explore all the possible neighbourhoods of all possible sizes for each user. In order decrease the number of neighbourhoods explored, we take over the idea from Section 4.2.3, where the next-most-similar user is added on the basis of the similarity between the current user and the candidate user's observations. In this work, we use the cosine similarity between the first N observations of the two users.

Recall from Section 4.4.4 that the exact N over which the similarity was computed did not strongly influence the performance of the model trained for short users - we understand from this result that it necessary to accept that both the choice of N and the similarity measure over the EMA data are somewhat arbitrary (although intuitive) choices. In order to accommodate for the possibility that the next included user is nevertheless not a 'useful' neighbour, we must allow the algorithm to overcome a poor ordering of the users. In order to achieve this, we ensure that unlike the early termination method introduced in Section 4.2.3, we follow rather the "exhaustive search" method, which does not terminate the loop on encountering the first user that increases model error. This is the first of two decisions in our workflow that is aimed at decreasing the likelihood of overfitting a neighbourhood to a user.

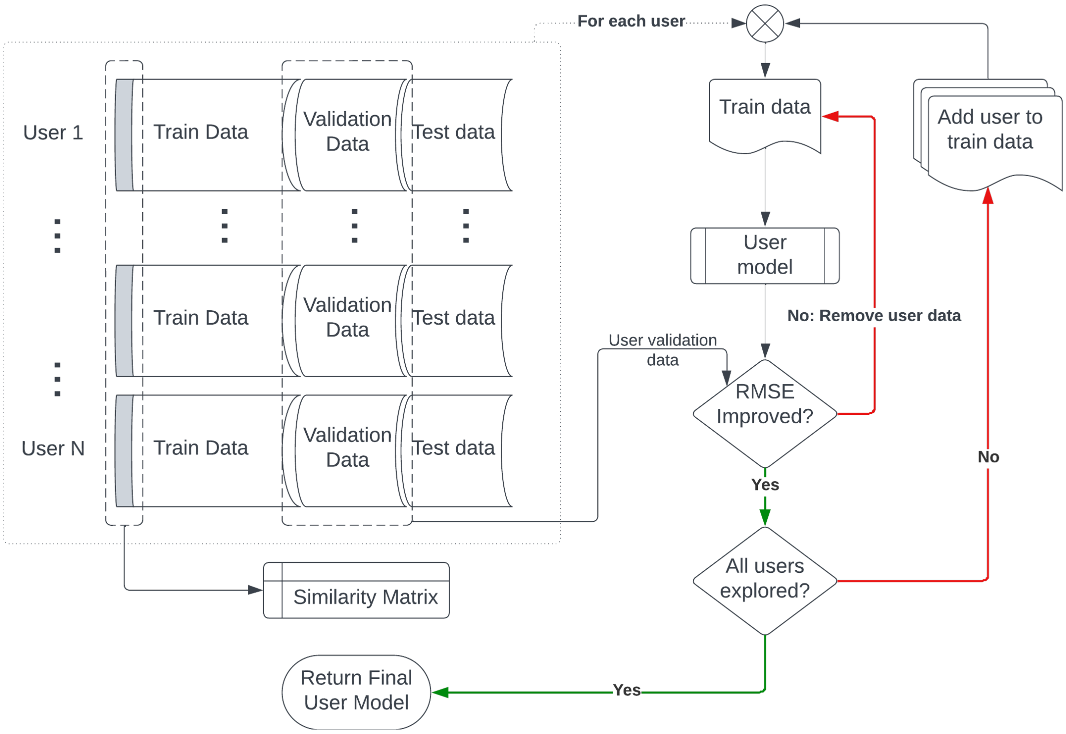


Figure 5.1.: The error-driven neighbourhood discovery framework

The second way to ensure that a model doesn't overfit the neighbourhood to the user is to allow some tolerance for small changes in error during the neighbourhood discovery process. This is especially critical when the neighbourhoods are small. For example, imagine a model trained on the data of the user itself and just one additional neighbour. The fact that the error increases by a small amount (say 0.01%) for such small neighbourhoods is highly unlikely to be because the 'best' neighbourhood has already been found, and more likely than not a statistical artefact. In fact, even if the error increased by a bigger margin, it can be argued that terminating the search so early can be detrimental to the personalised model's generalisation ability. Therefore, we include a tolerance parameter in our loop, which allows neighbourhoods to expand and the search to continue, as long as the model error does not increase more than 10% from the best-performing neighbourhood discovered so far. Please note that applying the threshold as a percentage and not a fixed parameter essentially tailors the threshold to the user, since a user who has an RMSE of 5 will reject models that increase RMSE by more than 5.5, while a poorly predicted user with an RMSE of 50 will only reject candidates that increase the error beyond 55. The algorithm to discover the best neighbourhood for a user from a list of candidates is given in Section 5.1.

Intuitively, this means that we prioritise including a neighbour and its data over excluding it, as long as the increase in error does not exceed the tolerance. Please also note that the tolerance limit is defined against the best validation error seen *so far* during the neighbourhood discovery process, and not against the current validation error. This is necessary since a chain of u users that are a poor choice, but whose addition consistently increase the error by a margin smaller than the threshold (for example, by 1% at every step) would gradually draw the user's model away good performance (given a long enough chain u). Fixing the tolerance against the best performing neighbourhood encountered so far allows for small increases in the error that increase the amount of training data for (and hopefully the generalisation capabilities of) the model.

Algorithm 5.1 Error-driven method to discover a neighbourhood for user $u_{current}$ from list of candidate neighbours $U_{candidates}$

```
Require: User $u_{current}$, users $U_{candidates}$, threshold $\alpha$, similarity function S(), $validation\_data(u_{current})$
training_data ← $data(u_{current})$
Train model $\Omega_{prev} \leftarrow \Omega(training\_data)$
prev_error ← $RMSE(\Omega, validataion\_data(u_{current}))$
best_error ← prev_error
Sort $N_{current} = n_i \in U_{candidates} | S(u_s, n_i) \geq S(u_s, n_j) \forall i < j$
for $n_i \in N_{current}$ do
    training_data ← $training\_data \cup data(n_i)$
    Train model $\Omega_{curr}(training\_data)$
    current_error ← $RMSE(\Omega, validation\_data(u_{current}))$
    if (currrent_error ≤ prev_error) | (current_error ≤ $best\_error * (1 + \alpha)$) then    ▷ Performance is allowed to deteriorate by $\alpha * best\_error$
        if current_error ≤ best_error then
            best_error ← current_error
        prev_error ← current_error
        $\Omega_{prev} \leftarrow \Omega_{curr}$
    else
        training_data -= $data(n_i)$    ▷ Do not terminate loop, continue search
        continue
return $\Omega_{curr}$
```

5.2.2. Add-best-neighbour baseline: Removing the effect of user ordering from neighbourhood discovery

The method described in Section 5.2.1 has the disadvantage that the neighbourhood discovery process gets 'locked in' to a user once it is seen that the user improves prediction error over the validation data (no matter by how small a margin). To explain with an example, imagine a user u_1, whose ideal neighbours are $u_3, u_4, andu_5$. The workflow described in Section 5.2.1 might add u_2 to the model since it decreases the error (which is indeed likely since in the early stages the model has very little data). The current model that has u_1, u_2 might show a high error when combined with u_3. i.e., the users u_2 and u_3 are incompatible, and the fact that u_2 was added first to the model means the model will never converge on the optimal neighbourhood.

To fix this problem, we propose the following baseline method - instead of iteratively adding each user that decreases the error of the model towards the validation data, search the entire dataset, and at the end of the loop, add the user that *most improved* the model. In the case of the example presented above, the loop searches models $train\_model(u_1, u_2), train\_model(u_1, u_3), \ldots train\_model(u_1, u_5)$, and find that each has errors $error_{1,1}, error_{1,2}, \ldots error_{1,5}$. If the model with (u_1, u_4) showed the best performance, then the current model's neighbourhood is expanded by one user to include the data of u_4. Since only the user with the *best* performance is added at each step, we remove the sensitivity of the method proposed in Section 5.2.1 to the ordering of users according to the similarity measure, and hopefully make the algorithm less likely to get stuck at local minima.

5.3. Experiments

5.3.1. Datasets:

Our proposed approach is tested on TYT and UNITI datasets already used in some of our previous work (see Sections 4.4.1 and 4.4.2). From both datasets, we exclude users who contribute less than 30 days of data. The TYT dataset contains data collected over a much longer time frame (2014 to 2022), and the UNITI dataset has data collected more intensively, but from a shorter time frame of 1 year. Table A.4 in the appendix shows the statistics for the number of users and their lengths. It can be seen that after applying the cut-offs, the longest users are 30 times and 9 times as long as the shortest ones for the TYT and UNITI datasets respectively. The TYT dataset also contains users that are much longer, since the data is collected over a much larger time frame. The UNITI users, though more comparable in length, are shorter since they are limited to a maximum length of 1 year at most.

For TYT, all questions have numerical answers between 0 and 1, and the two binary questions q1 and q8 are not included in the analysis. The tracked variables are loudness, distress, mood, arousal, stress, and concentration. The forecast variable is tinnitus distress (question3). For the UNITI dataset, all answers are on a scale of 1 to 100, and the EMA questionnaire has two parts, the part with the momentary assessments (4 questions), and the part with daily information. The momentary questions track tinnitus loudness, distress, tension in the jaw and neck. The daily questions query the number of times the patient thought about tinnitus, the degree to which the day was affected by tinnitus, maximum volume, stress, exercise and general mood.

5.3.2. Analysis of neighbourhoods discovered by the similarity-guided framework:

Since our approach to fit one model per user in the EMA app is without precedent, and because the heterogeneity of tinnitus makes it impossible to anticipate the nature of a ‘correct’ output, the quality of the neighbourhood discovery framework needs to be judged by the quality of the predictions at the user level. Towards this end, the prediction quality of the neighbourhoods discovered by the similarity guided neighbourhood discovery framework is compared to the quality of the predictions from the global model. The average size of the neighbourhood (number of users), and the average size of the training data (number of rows of data used to train the model) are compared against the global model, along with the performance of the personalised model in predicting the user’s data. The discovered neighbourhoods are also analysed for the likelihood of users being used to a greater or lesser degree in the personalised neighbourhoods of other users.

A baseline over all data: Like our other personalised neighbourhood models, we test the efficacy of our proposed method against a baseline model that has been trained on 70% of all users’ data. Since the validation set is no longer necessary (there is no iterative neighbourhood discovery model to validate), the part of the sequences that would have been held out for validation of the neighbourhood are added to the global model training data, giving the model even advantage in terms of amount of data to learn from. The models are trained to use all the questions from the current EMA response to predict the value of the target variable (tinnitus distress) at the next time step. We test the approach on the TYT and the UNITI EMA datasets.

Analysis of user-level errors: We compare not only the final aggregated errors of all predictions generated by our proposed workflow compared to the predictions generated by the global model, but also conduct an analysis to confirm which users are better predicted by the global model compared to the global model. Since each user's data can be predicted by both the global model as well as the personalised model, we will analyse the degree to which a user is better predicted by the global model than the personalised model with the discovered neighbourhood.

Analysis of impact of model caching (memoisation): Many dynamic programming methods use memoisation to precent the repeated invocation of models that will be invoked more than once in the attempt to solve a problem. In our case of neighbourhood discovery, we store each intermediate model (whether it proves to be 'useful' in terms of decreasing error or not) in the global cache of trained models so that a subsequent user who explores the degree to which a previously explored neighbourhood can avoid re-training a previously explored model, and directly apply it to the validation data. We check the number of times the memoised model is invoked, and compare this to the total number of models trained.

5.3.3. Comparing the similarity guided neighbourhood discovery framework with the unordered neighbour discovery baseline

Since the similarity guided neighbourhood discovery workflow searches the candidate users in decreasing order of similarity when training personalised models, the unordered neighbourhood baseline investigates the degree to which this method is approximated by a method that is insensitive to the order in which users are explored. If the similarity function does indeed order the users in decreasing order of 'utility', then the two methods should converge on the same neighbourhood.

We make the following three comparisons for the unordered neighbourhood discovery baseline:

- The overall performance of the unordered search compared to the global model.
- The user-level errors achieved by the unordered search compared to the user-level errors achieved by the global model.
- The average size of the discovered neighbourhood from the unordered search.

5.4. Results and discussion

5.4.1. The similarity-ordered neighbourhood discovery method

Comparing the neighbourhood discovery model against the global baseline: Figure 5.2 shows the boxplots that compare the performance achieved by the global baseline trained with all users' data, compared to the predictions generated by the neighbourhood discovery framework presented in Section 5.2.1. Each pair of box plots show the RMSE at the observation level for each observation in the test set, when predicted by the discovered neighbourhood (left) and global model (right) respectively. For both datasets, we see that the mean and the median errors from the discovered neighbourhood are both smaller. It is also to be noted that, the bottom whiskers are lower in both datasets (there are more predictions made that have lower error), although it appears that the IQR is indeed larger (i.e., it does still make large errors). It seems therefore that the neighbourhood discovery is able to pick up on some idiosyncratic answering patterns (and hence able to make slightly more very-low-error predictions), but this personalisation does come at the cost of a possibly larger spread in the errors as well. A user-level analysis is necessary to understand if this spread comes from a few poorly predicted users, or whether all users are affected.

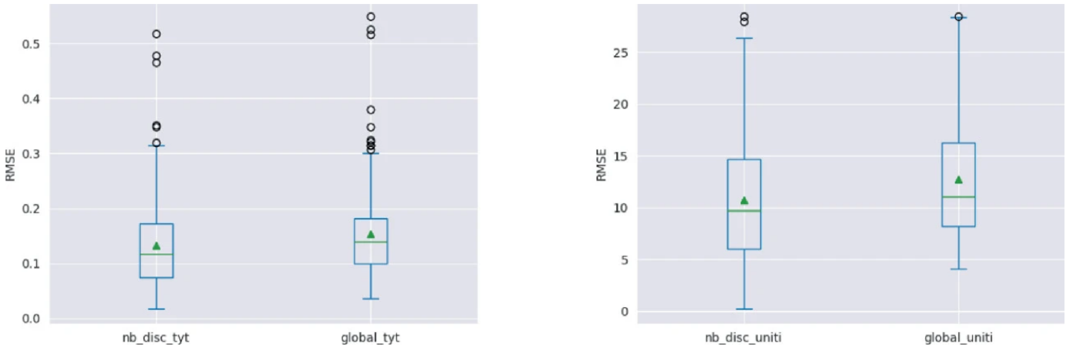


Figure 5.2.: Box plots for the performance achieved by the neighbourhood discovery framework compared to the global baseline trained over all users for the TYT (left) and UNITI (right) datasets

Table 5.1 shows the overall performance of the global model and the models built on the discovered neighbourhoods. The difference in mean RMSEs (already shown in the boxplots as the green triangles) between the global model and the discovered neighbourhood method are 0.153 vs. 0.136 for the TYT dataset, and 12.767 vs. 10.772 for the UNITI dataset respectively. The overall percentage improvements are not large, but need to be understood in the context that the personalised neighbourhood models are trained over less data to achieve a $\approx 13-15\%$ improvement. An analysis of user neighbourhoods is necessary to confirm whether the personalised neighbourhood discovery method benefits all users.

User-level error analysis: We have already seen that the mean errors of the neighbourhood discovery framework do indeed improve the overall error. The fact that the medians are lower already suggests that the benefits are shared across most users. In order to investigate the results at the user-level, we compute the RMSEs for the predictions from each model (global and the discovered neighbourhood) for each user

	TYT	UNITI
Global model (RMSE)	0.1534	12.767
Discovered neighbourhood (RMSE)	0.1355	10.772
Percentage improvement	12.99%	15.62%

Table 5.1.: The performance of the global model vs. the discovered neighbourhood on the TYT and UNITI datasets

separately. The user-level error figures for the discovered neighbourhood can then be compared to the same user's error when predicted by the global model. In our results, we found that 188/227 and 199/222 users in the TYT and UNITI datasets respectively were better predicted by the personalised neighbourhood model. This result is remarkable, because it shows that 82.8% of the TYT users and 89.6% of the UNITI users were better predicted by the similarity guided method than the global model. Although the difference is small, it is clear that our proposed approach benefits the majority of the users, and the performance gains are not concentrated on a few users alone.

A Wilcoxon signed-rank test was performed to test if the user-level RMSE from the global model and the discovered neighbourhood approach were generated by the same distribution. We found that the null hypothesis can be rejected with p=4.55e-27 and p=3.15e-35 for the TYT and UNITI datasets respectively.

Although our proposed neighbourhood discovery method benefits the majority of users, the degree of improvement in predictive power is still rather small. An analysis of the neighbourhood sizes suggested to us that the neighbourhood discovery method might serve the purpose of excluding poor neighbours (i.e., most users are useful, and the method gives better predictions because it identifies those users that need to be excluded to improve performance). To test this, we plot the adjacency matrices of the discovered neighbourhoods for each dataset as a 'heatmap'. The different users' neighbourhoods are separated in the Y axis (one row per user), and the X axis indicates whether a user was used in the neighbourhood of another (indicated by a 1), or not (indicated by a 0). The heatmap shows the 1s as bright spots, and the 0s as black ones.

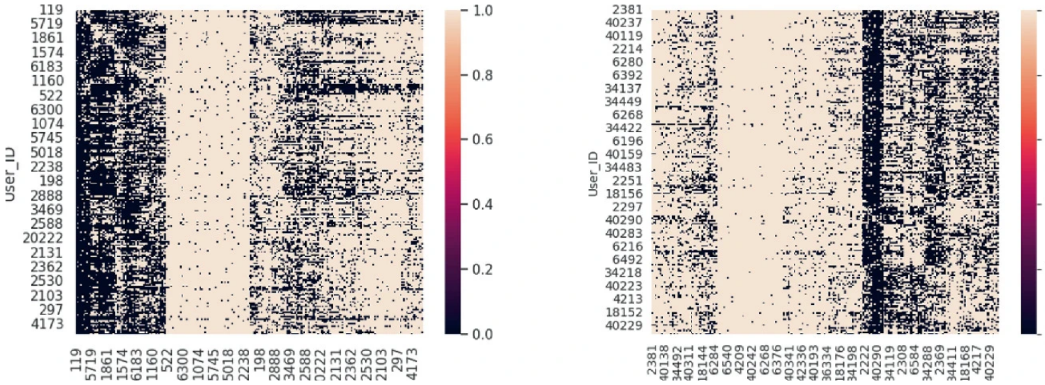


Figure 5.3.: Adjacency matrices for neighbourhoods discovered in the TYT (left) and UNITI (right) datasets respectively.

Figure 5.3 shows the adjacency matrices as heatmaps for the TYT and UNITI datasets respectively. We see an unexpected result that for both datasets, there is a black vertical band in the heatmaps. On the TYT dataset (left side of image), the black band is at the beginning, and in the UNITI dataset (right side of image), the black band is thinner, and is on the centre-right of the plot. This suggests to us that there is a group of 'ostracised' users. i.e., these users are rejected by most other users because they increase the error of their models. The fact that there is no corresponding horizontal band in these users is very interesting, since it shows that while most users' models suffer from the inclusion of these anomalous users, their models themselves are improved by the inclusion of other viewers' data.

Although not as clear, there is also a vertical bright band of users that are included in almost every other user's neighbourhood. This band can be seen at the centre-left of the heatmaps of both datasets. This suggests that apart from the users who contribute negatively to almost all users' models, there are also very popular "celebrity" users that improve the models of almost every other user.

Although unexpected and undoubtedly interesting, a full analysis of what makes these users special is unfortunately out of this work's scope, since it requires extensive medical expertise. However, we would like to call out that this is precisely the type of benefit we stressed when we argued that the user-centred models where neighbourhoods are delivered along with the models themselves allow one to discover some previously unknown relationship or property in the users.

Impact of model caching and reuse: Since our iterative neighbour discovery framework trains a huge number of models, we implemented a chaching mechanism that is inspired by dynamic programming approaches that saves each trained model in a dictionary. If another user explores the same neighbourhood for predictive quality over its own validation data, then the retraining step can be skipped and the cached model can be reused, saving time. We implemented a counter in our code to check the number of model training invocations that scored a hit on our cached set. Unfortunately, we see that for both the UNITI and the TYT dataset, there were very few cache hits. The TYT dataset trained 51,529 models, out of which only 105 (0.2%) got a cache hit. The number of hits was comparable for UNITI dataset as well, with 49,284 models trained, with only 177 cache hits (0.3% hit rate). Although in our case the cache hit rate is very low, it is important to note that even small increases in the number of users causes an exponential increase in the number of trained models. It might be that the idea of caching trained models is still useful, just not in the case of these two datasets. It might be that other datasets with a diverse set of users who are all similar to a tight knot of users might benefit from caching the models in similarity-ordered search.

5.4.2. Unordered neighbourhood discovery baseline

Comparison against global baseline Figure 5.4 and 5.5 shows a comparison of the unordered neighbourhood discovery baseline against the performance of the global model for the TYT and UNITI datasets respectively. It can be seen that for both methods, there is almost no performance difference between the global and discovered neighbourhoods. This suggests that our unordered neighbourhood discovery method is too greedy, and tends to get stuck in local minima. We see that for the TYT dataset, the overall RMSE of the neighbourhood discovery method is only 5.5% better than the global model, and only 0.34% better for the UNITI dataset.

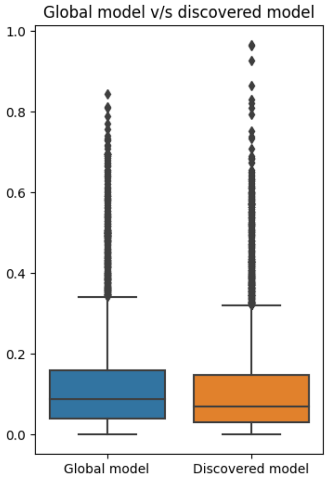


Figure 5.4.: TYT Dataset: Errors for the global model and unordered neighbourhood discovery model

Since the performance of the model is so close the global model (and because it is worse than the similarity guided discovery method), it is reasonable to suspect that neighbourhoods are overfit for at least some users.

User-level errors comparison: At the user level, we find that the neighbourhoods discovered by the unordered search method only beat the RMSE from the global model for ≈ 46% of the users in the TYT dataset, and ≈ 44% of the users in the UNITI dataset. The boxplots for the user level errors are shown in Figures 5.6 and 5.7. Combined with the results at the overall level, it is clear that the greedy brute-force neighbourhood search method overfits the user neighbourhood. While the user-level boxplots show that almost half the users do benefit to some degree from the personalised neighbourhood, the large increase in the number of worse-predicted users suggests that the similarity ordered search is less prone to getting stuck in local minima. This is further backed up by the low numbers for the mean, median, and max discovered neighbourhood sizes shown in Table 5.2.

# Neighbours	TYT	UNITI
Mean	11.16	10.99
Median	8	10
Max	49	56

Table 5.2.: Number of neighbours discovered by the unordered search method

Impact of model caching and reuse: An analysis of the number of models trained and the usefulness of caching reveals that 525,481 models are trained for the TYT dataset, and 1,019,564 models are trained for the UNITI dataset. This shows a sharp but predictable increase in the number of tested models before the final neighbourhoods are ‘frozen’. For the TYT dataset, there were 28,960 instances of a previously trained models getting reused from the model cache (5.51% hit rate),

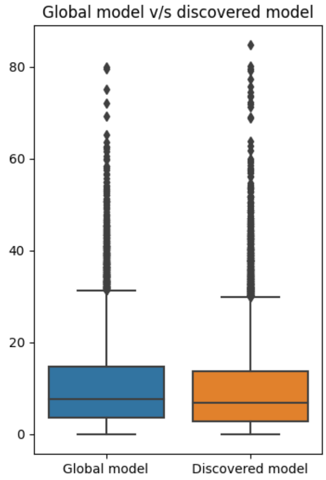


Figure 5.5.: UNITI Dataset: Errors for the global model and unordered neighbourhood discovery model

and the for the UNITI dataset it was 53,888 reuses out of 1,019,564 models trained (5.29%). Since the runtimes for the entire brute-force neighbourhood discovery workflow was <1hr for the UINTI dataset, the benefit of caching is debatable, even thought the hit rate on the cache is higher than in the previous case.

5.5. Conclusion

The first two main methods we describe in Sections 3.2 and 4.2 explored training personalised models for users on the basis of similarity built on static and dynamic data. Our subsequent work learning models separately for users with short sequences shows us that the notion of a neighbourhood can be discovered. Since our early results do not show conclusively that similarity built on either static or dynamic data fully determine the neighbourhood of a user, in this chapter, we explore methods that discover this neighbourhood based on trial and error approaches. Section 5.2.1 explores an approach where the computationally intractable problem of searching every permutation of a user set is simplified by enforcing a similarity-driven search order. We start with the data of a single user, and then iteratively add the data of the next-most-similar user, searching through the entire list of users in this order to keep all users that improve model performance, and discard all users that do not. Our main findings were:

- We tested our similarity-driven neighbourhood discovery workflow against a global baseline model trained on the data of all users and saw that our proposed approach decreases the overall prediction errors by $\approx 13\% - 15\%$.
- We further investigated the extent to which the improvements in prediction quality is distributed through the set of users, and found that 82.8% - 89.6% of the users were better predicted by our proposed method compared to the global baseline model.
- This shows that although the improvement in overall RMSE is not large, the

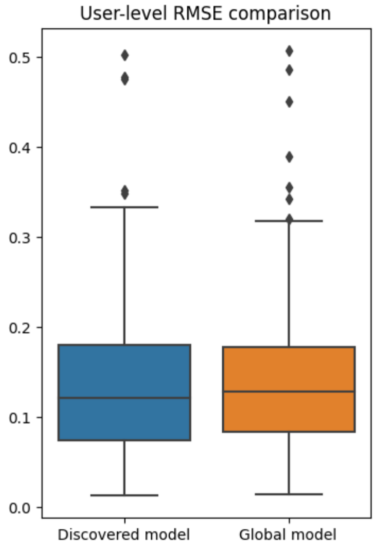


Figure 5.6.: TYT Dataset: User-level RMSEs for the global model and unordered neighbourhood discovery model

benefits are evenly distributed among all users, and not the result of a few long users dominating the RMSE computation. A Wilcoxon-signed rank test of the distribution of user-level RMSEs from the global and neighbourhood discovery models rejected the null hypothesis that the two samples are drawn from the same distribution by p=4.55e-27 and p=3.15e-35

- We investigated the final neighbourhoods discovered by the model by plotting the adjacency matrices as a heatmap, and found two emergent patterns:
  - There was a small group of users in each dataset that were consistently rejected from all other users' neighbourhoods.
  - Conversely, there was also a slightly larger group of users in both datasets that contributed positively to almost all users' neighbourhoods.
- Since our workflow is extremely wasteful in the number of models trained (the vast majority of intermediate trained models are discarded), we implemented a model caching system. However, with hit rates of 0.2% and 0.3% for the two datasets, we find caching to be of little practical use. This might be a consequence of the fact that most users have mostly unique similarity-ordered user lists. Other datasets where a clearer sense of similarity exists might still benefit from the approach. We also expect that the utility of the cached models will increase with the number of users (which will cause exponentially more models to be trained), or if the complexity of training a model increases.

Since our early results suggest that computing the similarity between pairs of users on the basis of their early interactions is not effective, we compared the performance of the similarity-driven neighbourhood discovery method with a more powerful baseline that is less sensitive to the user ordering. This method works by adding the user that results in the largest decrease to the RMSE at every stage, instead of adding the next-most-similar user that decreases the error. This was expected to allow the

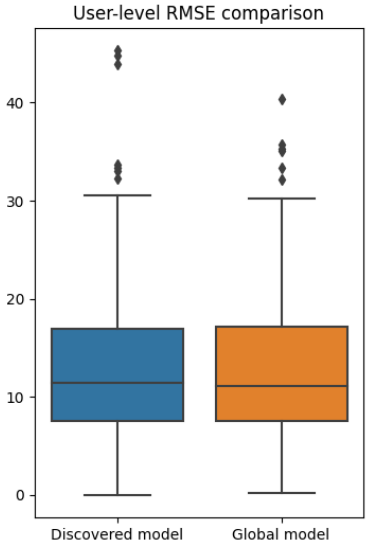


Figure 5.7.: UNITI Dataset: User-level RMSEs for the global model and unordered neighbourhood discovery model

model to overcome its sensitivity to the user ordering, but our experiments revealed that the performance was worse than for the similarity-guided counterpart. Our main experimental findings for this method were:

- Even with commodity hardware and training more than 1 million models, the execution time for the method was <1hr for the larger dataset[1].
- Although the overall RMSE was slightly better than the global model, the brute force neighbourhood discovery method tended to get stuck in local minima, resulting in better models only for ≈ 45% of the users (compared to global). The small neighbourhood sizes for the neighbourhoods discovered by the brute force method suggest that this method is prone to overfitting the neighbourhood.
- The cache hit rate was 5.28% (UNITI) and 5.51% (TYT), but the low overall execution time makes the case for the caching unconvincing.

[1] 50m:57s of runtime on a Core-i7 11th Gen. with 40 GB RAM.

6. Conclusion and Future Work

The primary goal of this thesis was to explore complementary methods to develop personalised models for datasets that include data in two modalities - static and dynamic. Towards this end, we investigated three main approaches:

- training personalised models using neighbourhoods built on static data,
- training personalised models using neighbourhoods built on dynamic data, and
- training personalised models using a supervised notion of similarity, for cases where similarity cannot be reliably measured.

These main approaches form the basis for the three research questions detailed in Section 1.2. We explore our contributions towards each of the questions in Sections 6.1, 6.2, and 6.3 below.

6.1. Personalised models using similarity based on static data

As explained in Section 1, our main assumption is that our data comprises of various 'entities', each of which contributes data in two modalities - one relatively unchanging and 'static', and the other that changes over time, which we call 'dynamic'. We begin our investigation by exploring the degree to which similarity in the static space can be used to create a personalised model. Our two main questions are:

- To what extent can static similarity be exploited to train personalised models?
- To what extent can expert knowledge about the similarity between entities be incorporated into the modelling process?

6.1.1. RQ1.1 Exploiting Static Similarity to Train Personalised Models

In our personalised modelling approach, we train one model for each entity / user in the dataset. This presents some challenges that are not uncommon in panel-like datasets comprising multiple entities, most notably the fact that most entities are too short. Our work in Section 3.2 explores augmenting the personalised models to combat the scarcity at the entity level.

Our proposed method discovered the kNN of each user based on their static data, and explored two ways to combine the data of the discovered neighbours: data augmentation, where the dynamic data of all neighbours are pooled to create a single model, and model augmentation, where one model is trained on each neighbour, and the model parameters are averaged. It was found that data augmentation worked better for 2/3 datasets, and comparably for the third.

Pooling the data of each neighbour presents one further challenge - dealing with timestamps effectively. In our experiments, we found that keeping timestamps as-is resulted in better models than when each user or entity was aligned to their own local clocks beginning at 0.

Our approach to pool the data of the k nearest neighbours of each user while preserving the timestamps assumes that the static data picks good neighbours. We tested this assumption using a proposed baseline which selects the same number of neighbours as the kNN, but selects them randomly (we call this the kRE baseline). This method indirectly measures the degree to which the error drops simply because we added more data to the model, rather than because the kNN selected good neighbours to learn from. Unfortunately, we saw that the kRE baseline shows very similar performance to the kNN model, suggesting that our neighbourhood selection can be improved. However, quite surprisingly, we saw that both the kNN and the kRE model benefited from augmenting the model with a small k. It is our suspicion that this is the result of dataset-level trends leaking into the system because through the timestamp. We therefore avoid using the timestamp in our future modelling efforts along with the other independent variables. We suspect that this indirect leaking of dataset-level tendencies is the reason far-term predictions outperform the near-term predictions for the AQI and Amazon datasets, where the trends towards higher / lower values over time very pronounced.

6.1.2. RQ1.2 Improving Neighbourhoods Using Expert Knowledge

After fixing the static-data-based neighbourhood discovery to use data-augmentation, we explored the degree to which these neighbourhoods can be tweaked to incorporate expert knowledge. We experimentally validate our solution on the TYT dataset, where we know of the two anomalous groups of tinnitus sufferers: (a) a group that suffers from unreasonably high tinnitus distress in spite of relatively low symptom severity, and (b) a group that suffers unexpectedly low tinnitus distress in spite of relatively high symptom severity.

By applying the ‘tinnitus questionnaire’ score of the patients in the static data, we split the patients into 2 distress clusters. The tinnitus volume data was split into three clusters using a similar approach. Since each patient falls into one of three loudness clusters and one of two distress clusters, we assign all patients into one of six groups based on the unique combination of which loudness-distress clusters they fall into. Of these six groups, low-loudness+high-distress and the high-loudness+low-distress patients are known to be anomalous. Our proposed method incorporated this information in to the personalised modelling process by limiting the kNN to discovering in-group (i.e., ‘concordant’) tinnitus patients only. We see that the groups all differ in the degree of predictability, and that 4/6 groups are better predicted than by the unrestricted kNN model, suggesting that including expert knowledge to exclude discordant users from each others’ neighbourhoods does indeed improve performance. Additionally, we also found that high-loudness+low-distress patients are harder to predict than the opposite case of patients who are severely distressed even by low tinnitus volume. This suggests that excluding patients of high heterogeneity (and therefore low predictability given each others’ data) is the reason why the models improve.

6.2. Leveraging Dynamic Data to Build Entity Neighbourhoods

Since using the static data did not sufficiently inform the predictors about the dynamic data, the next step logically is to investigate the extent to which the dynamic data can be leveraged for the same purpose. Since our earlier results show that the timestamp can leak information about the entities' futures, we pivot to a slightly different learning problem of predicting the short term future of the sequence given the current observation. Our main approach towards using the dynamic data to discover user neighbourhoods has two flavours - one that tries to create a summary of the EMA sequence that can be exploited to measure the similarity between EMA sequences of unequal length, and another that exploits the differences in the lengths of the sequences themselves to group users. Our main takeaways for the summarisation approach are listed below:

- Summarising EMA data with Hidden Markov Models:
    - In this approach, we propose summarising the users as "the amount of time they spend in each hidden state". We use the percentage of each sequence generated by a state as a way to handle the different lengths in the sequences. The summary of each user / patient is used as the vector over which the kNN is computed when discovering a personalised neighbourhood.
    - For both datasets we tried, inspecting the model's hidden states and transition matrices did not present a clear 'winner'. Model quality metrics like AIC, BIC and log-likelihood were similarly unhelpful, since they chose either too few states or too many. Evaluating empirically by measuring model quality resulting from 2 to 10 states showed no clearly dominant solution.
    - Although the HMM-derived representations improved on the quality of predictions delivered by the global model by only 2.9% - 9.9%, they did so with only 1.5% - 6.6% of the total number of users in the dataset. This suggests that focusing the personalised predictors do indeed work, but further work needs to rule out that the HMM states are only acting as a proxy in capturing a user's average loudness and distress.
    - By limiting the HMM model to learning only the interplay between tinnitus loudness and distress, we allow the model to be transferred between separate mHealth apps with only partially overlapping questionnaires. Our experiment on learning the HMM on the more regular UNITI dataset and transferring the HMM to the TYT dataset resulted in performances that exceed the global model by 7.5% (instead of 9.9% on the home-grown representations).
    - To summarise, the performance gains from using HMMs to summarise the EMA sequences and discover their neighbourhoods resulted in modest performance improvements, but the low sensitivity of the results to the number of states is a discouraging result. Further investigation is necessary, especially a deeper analysis of the model with an expert is critical to better assess the correct number of states, especially since our intermediate results suggested that 'clones' of certain states exist with the same loudness and

distress, but different covariance matrices that connect them. The need for capturing users that show rare behaviours needs to be balanced with creating representations that don't all trap users into a very small number of states.

Our second approach towards summarising the EMA sequences drew inspiration from some work in our own workgroup [42], which discovered different patients preferentially express different Granger causalities in their EMA data. We therefore investigated the degree to which this information can be exploited to discover patient neighbourhoods:

- We found indeed that different patients express different Granger cusalities, but the process of testing for this in the multivariate EMA sequence is prone to false positives. This problem is compounded when considering lags > 1. An analysis of the discovered relationships between the EMA variables showed that the UNITI dataset (with longer more regular observations) showed stronger differences in the causalities expressed by the users (even though the type I error cannot be avoided, this suggests an underlying signal).

- We investigated two ways to use the discovered Granger causality matrix to discover similar users, one that computes a Jaccard similarity that translates to "percentage of shared Granger causal relationships expressed between the users", and another that focuses the similarity function on only relationships toward one variable. It was found that causalities toward no single variable was more important in terms of model performance. The similarity computed over the full binary Granger causality matrix found better neighbours.

- We see that the performance of the kNN model based on representing users as the Granger causalities they express was able to match (and slightly exceed) the quality of the global model. We saw that the UNITI dataset with more regular observations benefited more, and that while the performance benefit was not large, matching the performance of the global model was possible with a tiny fraction of the users (k=10).

Our third approach partitions the users in the EMA datasets into two groups ('long' and 'short' users) based on the amount of data they contribute. The motivation behind this method is twofold, one is that predictions are possible for users who contribute so little data that they would typically be excluded during preprocessing, and the other is more medically motivated, since it is not clear whether the users who engage little with the system do so because of some underlying similarity in the experience of the disease. We explored the degree to which models trained over users that contribute a lot of data can be transferred to the short users, and the degree to which the predictions generated by this non-personalised transferred model can be adapted to the sequences of each short user. As a next step, we explored the possibility that not all long users might be equally predictive of a short one. The key takeaways are summarised below:

- The model trained on long users do indeed make predictions (albeit with slightly higher error) on data of short users. We found through augmenting the predictions with a kNN regressor focused on the short users' sequence that the transferred models could be personalised to the short users.

- Weighting the errors of the transferred model and the short-user-centred model on the basis of their past errors over predictions for the user decreased the number of extreme errors made by each model separately. Our experiments, however, were restricted to small choices in k, since the sequence for each short user is already small.

- To further personalise the predictors trained for short users, we investigated the degree to which the neighbourhood of long users can be personalised towards each short user - i.e., 'Do all long users contribute equally towards the predictions for a short one?'. Our approach investigated 'growing' a neighbourhood iteratively based on some similarity measure. We found that not all long users are equally important in training personalised models for short users, and that the RMSE gains were 2.5% and 36% for the two datasets we tried, while the models used only 40% - 60% of the training data. Apart from the fact that performance improvements come with fewer data, the result also needs to be placed in the context that the models were not trained on the data of the users in question at all, only the discovered neighbourhoods, which makes it more remarkable.
- We explored two neighbourhood discovery methods, one that fully explores all users in a similarity-ordered list while searching for potential neighbours, and another that terminates the search process upon encountering the first user that increases the validation error. We observed that for one dataset, both methods discovered the same neighbourhood.
- The performance of the models trained using similarity-ordered incremental exploration of neighbourhoods was surprisingly found to not depend too strongly on the exact similarity measure. Our choice to base similarity on first v/s most recent data vs. metadata summaries of interaction intensity did not significantly change the final model quality (or even most users' neighbourhoods!). However, we did find that some long users were more useful than others (they were included more often in short users' neighbourhoods). We take this to mean that we have not found the best measure of similarity between users based on their EMA data, although for the sake of argument we fix the similarity measure as dependent on the EMA data of the first five days.
- All the results, however, are derived from datasets too small to be entirely trustworthy. Replication of these results in other larger datasets is a priority before these results can be fully trusted.

6.3. Supervised Neighbourhood Selection Based on Validation Error

The results from the first and second parts of this work showed us that the static data and the dynamic data do not adequately capture the underlying similarities between users when building personalised predictors. This is because of the unreasonably effective kRE baseline, and the lack of a convincing performance gap (despite much smaller training data), on training personalised models using the neighbourhoods built on static data as well as summaries of EMA data. We therefore took inspiration from the last part of our work in RQ2 and generalise our method of neighbourhood discovery from short users to all users. This effectively turns the question “What can you learn from an entity's neighbourhood?” on its head, and makes it “Can you learn an entity's neighbourhood", which places the emphasis on which neighbourhoods are useful. We investigated the degree to which an 'optimal' neighbourhood can be learned through iterative exploration using the dynamic data of each user. Our main takeaways are listed below:

- Our similarity-driven neighbourhood discovery workflow (inspired by the 'exhaustive search' method developed for short users) was able to deliver ≈ 13% – 15% improvements in the overall RMSE compared to their global counterparts.
- We found that the performance gains were not restricted to a few users, and that 82.8%-89.6% of the users benefited from the personalised neighbourhood models. A comparison of user-level RMSEs showed that the errors from the personalised neighbourhood are statistically significantly different from the RMSE for the same users predicted by the global model, with p=4.55e-27 and p=3.15e-35 for the TYT and UNITI datasets respectively.
- Plotting the discovered neighbourhoods as a heatmap showed the existence of 'celebrity' and 'ostracised' users - users that are included in almost every other user's neighbourhoods, and vice versa. More interestingly, we also saw that the users who were rejected from most other users' models still benefited from including the other users in their neighbourhood (i.e., while other users avoided adding their data, they did not avoid the other users' data from being included in their models)[1].
- Caching the thousands of intermediate models trained during the neighbourhood discovery search process brought almost negligible benefit. Only 0.2%-0.3% of the models were re-used instead of being re-trained from scratch.
- We also proposed a more advanced baseline which removes the effect of ordering the users by similarity. While this exponentially increases the number of models trained, the run times were still within reach of commodity hardware. Unfortunately we see that the brute force search makes the discovered neighbourhoods get stuck in local minima, with only ≈ 45% of the users being better predicted by the neighbourhood models compared to the global model for both datasets. Although the cache hit rate was higher for both models, the low run time makes the case of caching intermediate models less convincing.

The main points highlighted in Sections 6.1 - 6.3 highlight that there are three methods to discover similar entites while training personalised predictors with EMA-like data:

- The static data that describe these entities can be exploited in the building of neighbourhoods. The kNN models, while they result in better performance than the global models trained over much more data, do seem to struggle to beat the performance of simpler methods that select the same number of entities (k) randomly. This shows that the static data is a useful tool in finding useful neighbours to learn from, but it needs to be augmented with more information.
- The dynamic data of the entities can also be leveraged to find the neighbours of an entity. We see that exploiting machine learning techniques to summarise the EMA sequences can help match the performance of the global model with quite small neighbourhoods (<10% of the users). This shows that the dynamic data

[1] A clinician who works on this dataset summed up the situation succinctly: "There are users that behave like universal blood donors, and others that behave like universal acceptors"

also contains valuable information for learning personalised models. However, like in the case of the static data, the dynamic data alone does not compute better neighbourhoods.

- Drawing upon the the insight that the choice of similarity measure used during neighbourhood discovery for short users does not have a big impact on final model quality, we propose a generalised iterative neighbourhood discovery framework to find that each user's personalised model can be trained over an iteratively expanded neighbourhood using the model quality as a way to supervise the notion of similarity. While seemingly a trial and error method like any other, we still believe that the outputs generated by this method are valuable not just because of better-performing personalised predictors, but also the fact that each user gets its own neighbourhood. This additional output (which let us discover that some users are avoided by all others, for example,) can serve as a starting point for clinical investigations. It is even possible that the results of such investigations may eventually improve the notion of similarity in the domain of tinnitus by uncovering hitherto unknown properties of users.

6.4. Limitations and Threats to Validity

As with most unsolved scientific problems with real-world relevance, the approaches presented in this work have limitations and threats to validity, since they make various simplifying assumptions of the world. While we have tried to test the methods against the more obvious threats, the ones that remain open still are listed below:

- Towards personalised predictors using neighbourhoods derived from static data:
  - Only one of three datasets included in the study were EMA datasets, and the Amazon reviews datasets gave no static data to derive our kNN models from. More datasets from the EMA domain that have all the desired properties would be useful to determine the efficacy of our proposed method in training personalised predictors that are more accurate than the global model.
  - The inclusion of the current timestamp was seen to leak dataset tendencies, giving the k-random entities baseline an unfair advantage. While internal experiments showed us that dropping the timestamp removes this disadvantage (and the decision was carried over into our subsequent works), a full re-run of the experiments would help to systematically show the impact of this decision on model quality.
  - Since we train a large number of models (one per entity), we have limited the complexity of the model to linear models, to avoid both runtime and workflow complexity issues (for example, user-level model selection and hyperparameter tuning).
  - The performance can be studied for neighbourhood sizes exceeding 50, since it is a small proportion of the entire dataset. It could be that the trends we observe reverse for larger neighbourhood sizes.
  - It would be useful to introduce EMA-like datasets from a domain where the similarity between entities / users is better known. The results on

    such a dataset can help confirm that the neighbourhood models really do benefit from the computation of neighbourhoods. In our current case, we are only inferring the existence of a neighbourhood from the result.

  - More sophisticated similarity functions can be used that can capture non-linear relationships between features.

- Towards personalised models using neighbourhoods derived from the dynamic data:

  - The HMM models trained to summarise the EMA sequences were not validated by an expert. A more thorough investigation of the models is necessary to know that the summaries of the EMA sequences do indeed capture disease-relevant differences between users. i.e., A model that achieves better performance on the basis of an HMM that contains states that make no sense to a physician is not to be trusted, since our goal is not only to have good personalised predictors, but use the neighbourhoods that are output by personalised predictors as well.

  - Some properties of EMA data are at odds with HMMs. The irregularity violates a core HMM assumption of temporally equidistant observations, and the large differences in length can create an HMM model that is biased to the long sequences it is trained on.

  - Further experiments are necessary to confirm that the HMM-derived neighbourhoods do indeed bring improvements to the predictive quality of the personalised models. It might be that the HMMs are only playing the role of grouping users based on their mean distress. Relatively simple baselines need to be created to assess the utility of adding a complex sequence model like HMMs.

  - Relatively simple preprocessing steps can have huge semantic impact on what the HMM is trained on. More experiments are necessary to determine if sequences that capture patient-level deviations from means are more effective at grouping similar EMA series compared to the current implementation, which is more likely to develop a proxy for average patient loudness and distress.

  - We only explore HMMs trained over the tinnitus loudness and distress dynamics. Including more variables might create models that more accurately group EMA sequences. However, this would make interpreting the HMM states more difficult.

  - The use of Granger causalities to summarise the EMA sequences were prone to false positives, since every permutation of the 2-variable pairs from the EMA features was tested with $1 \dots N_{lags}$ for Granger causal relationships. Tightening the confidence level can help reduce the likelihood of false positives, but when testing for a large number of relationships, it is better to create a model that is robust to false positives.

  - The extracted causalities can be used in a non-binary way, so that relationships with high statistical certainty are given higher weight than those with lower. The same can also be applied at a dataset level, where causalities that are discovered in the dataset at rates exceeding the base expected rate from type 2 errors alone can be weighted higher, since they reflect underlying properties shared by a large number of users in the dataset.

  - The current method gives equal weight to the existence of causalities and their absence. More sophisticated methods can weight the vectors so that two patients that both do *not* show a commonly occurring Granger causal relationship (between loudness and distress, for example) are given higher similarity.

- Towards personalised models for short users learned on users with long sequences:
  - The main and largest threat to validity of these methods is the size of the datasets they are trained on
  - The insensitivity of the model to the definition of the similarity measure suggests that the user neighbourhoods, even when 'useful' (resulting in low RMSE), are still added for the wrong reason. An analysis of the discovered neighbourhoods is necessary, where a clinician validates the discovered neighbourhoods and investigates the shared properties of users in each others' neighbourhoods.
- Towards a supervised notion of similarity for neighbourhoods:
  - The choice of similarity function that guides the neighbourhood selection process (cosine similarity over first five observations) is somewhat arbitrary.
  - The neighbourhoods generated by the supervised neighbour selection framework have not been analysed by a clinician to assess their medical relevance.
  - To truly assess whether the neighbourhood is useful, one can attempt to fit more complex models once the neighbourhood has been discovered. A neighbourhood computed over a well performing neighbourhood selection model should outperform the performance of the intermediate models.

6.5. Future Work

The collection of methods proposed in this work were developed sequentially, with the results from previous efforts directing the design of the following methods. The most promising avenues for further exploration are therefore concentrated on the most mature part of this work, introduced in Section 5.4. However, we will present the most promising ways in which the methods presented in each of the chapters may be extended.

- Building personalised predictors based on static data similarity:
  - The similarity function we applied is linear and cannot take complex interrelationships between entities into account (for example, if a patient has family history AND ear trauma). Substituting the similarity function with more sophisticated methods would be low-hanging fruit for an improvement that can be made independently of the rest of the workflow. Promising options are methods derived from graph similarity, where older

    methods [15] can be compared against their more modern counterparts [75] that train node (patient) embeddings. Another option would be training autoencoders to develop lower dimensional representations of users, so that a similarity function can be applied on the dimensionality-reduced vector [83, 44].

  - More complex models can be trained when the data permit it. A full model selection with hyperparameter tuning at the user level is of course computationally expensive, but necessary for a full evaluation.
  - EMA-like dataset from more domains can be used in the study. Any dataset with a known reference relationship can be investigated to see if the proposed workflow discovers neighbourhoods that share that known relationship.
  - Larger datasets would confirm the degree to which the results replicate, confirming the validity of our workflow. The current workflow could also be repeated using the logic followed in Chapter 4 onwards. This would show that the neighbourhood based models still outperform the global and user-centred models.

- Building personalised predictors based on dynamic data similarity:
  - Including larger and / or reference datasets would confirm the degree to which the proposed methods do indeed capture the underlying similarities between entities. Datasets with known relationships between entities (like in the m5 dataset) can help confirm if our neighbourhood discovery methods truly capture and exploit this known relationship.
  - More sophisticated models can be trained on the dynamic data once the neighbourhood is discovered, if sufficient data is available. Like in the previous case of the static data, user level model selection and hyperparameter optimisation can be performed to always extract the most from the training data. The sensitivity of such a method to overfitting would be valuable as a result in and of itself.
  - The time series summaries can be performed with more sophisticated methods like irregularity-aware t-LSTMs, and the performance of these methods can be compared against our simpler methods. T-LSTMs have already been proved to work for patient subtyping [8].
  - The method of using transferred long user models used to create predictions for short users needs to be validated on larger datasets to ensure the reliability of the result. A larger dataset would also allow us to investigate the effect of parameters that were not practically possible to tune over the small dataset.

- In addition to improvements that can be made to each of the methods above independently, it is also worth studying if all / some combination of all the above methods (including HMMs, Granger causalities, as well as static data) yield better results than relying on one method alone. There is reason to consider this method wlil be effective, since each method beat the global model, but not by a large margin. It is perhaps possible that a well crafted similarity measure that incorporates elements from all three approaches finds better user neighbourhoods.

- Discovering the neighbourhoods in a supervised manner:
  - The neighbourhoods discovered by the models have not been investigated for clinical relevance. This would be a top priority for assessing the utility of our results, since many of of choices are motivated by being easy to interpret. Less focus on interpretability can allow for more complex approaches that optimise for performance alone using as narrow a neighbourhood as possible.
  - Instead of the current similarity measure, the neighbourhood growth model could select the next user to add into the neighbourhood based on the learned representations, like from autoencoders. The effect of this ordering can be evaluated by how quickly the intermediate models achieve their final and best performance. Well selected neighbourhoods should improve models quicker than poorly selected ones. Applying the same method on benchmark datasets (like from the M5 competition [64]) can allow the fitting of more complex models.
  - The utility of the neighbourhood selection process can be investigated by fitting more complex models after the neighbourhood selection process has converged on the 'best' neighbourhood. Unlike in the previous cases where the size of k was fixed, the neighbourhood discovery method stands to gain the most from this because the amount of training data available in each user's neighbourhood can vary more strongly.
  - Much more sophisticated analyses are possible on the final as well as the intermediate neighbourhoods derived during the neighbourhood search process. The order in which the users are added might contain valuable information regarding user similarity, and methods like paragraph embeddings or association rule mining can discover such patterns in our output. This part of our output has not been analysed so far.
  - As with the other neighbourhoods, clinical validation is key to trusting our workflow. Apart from the neighbourhoods of the users themselves, differences between well and poorly predicted users, users with many and few neighbours, users that are used in many other users' training data and those that are not, etc. are all valid questions for a clinician.
  - The brute force method that tended to overfit the neighbourhoods can also be improved to have higher tolerances when neighbourhoods are small, so that users do not get stuck in local minima.
- The notion of supervised similarity might also be extended to a supervised metric learning problem[69], where we learn the similarity between patients based on how useful they prove to be in each others' neighbourhoods.

Appendix

A. Appendix

A.1. Appendix for Chapter 3

A.1.1. The six groups of tinnitus patients as derived by expert knowledge

The table below shows the six groups of tinnitus sufferers as derived from kMeans clustering loudness with $k = 3$ and distress as the total score of the “Tinnitus Fragebogen" (i.e., the "Tinnitus Questionnaire" in English) with $k = 2$. The total range of the scores is 0-24. The clustering does not find well separated clusters, but is intended rather as a way to find data-driven cuts in a continuous distribution. The results were validated by an expert.

Group	Group Description	N	Avg. Distress	Avg. Loudness
Group 1	High Distress, High Loudness	97	18.3	82.2
Group 2	High Distress, Moderate Loudness	168	17.0	54.4
Group 3	High Distress, Low Loudness	52	15.7	26.8
Group 4	Low Distress, High Loudness	35	9.2	77.1
Group 5	Low Distress, Moderate Loudness	83	8.1	53.0
Group 6	Low Distress, Low Loudness	81	7.4	28.1

Table A.1.: The six groups of concordant tinnitus sufferers as discovered by running 2-Means on distress, and 3-Means on Loudness

A.2. Appendix for Chapter 4

A.2.1. HMM state transition matrices

TYT Dataset

The following Figures A.1 - A.8 show the state transition matrices (left image), and the corresponding means for loudness, distress for the TYT dataset.

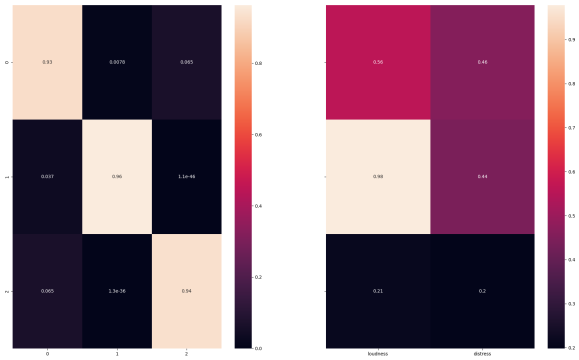


Figure A.1.: The TYT dataset: State means and transition matrices for HMM with 3 states

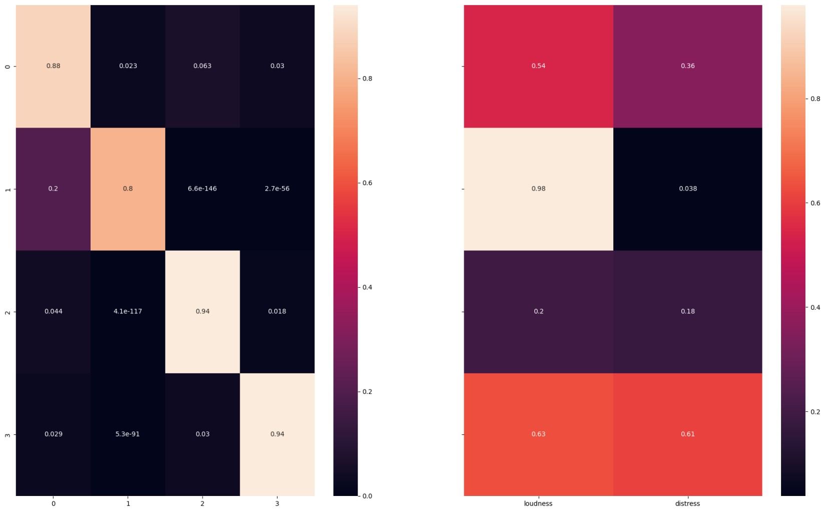


Figure A.2.: The TYT dataset: State means and transition matrices for HMM with 4 states

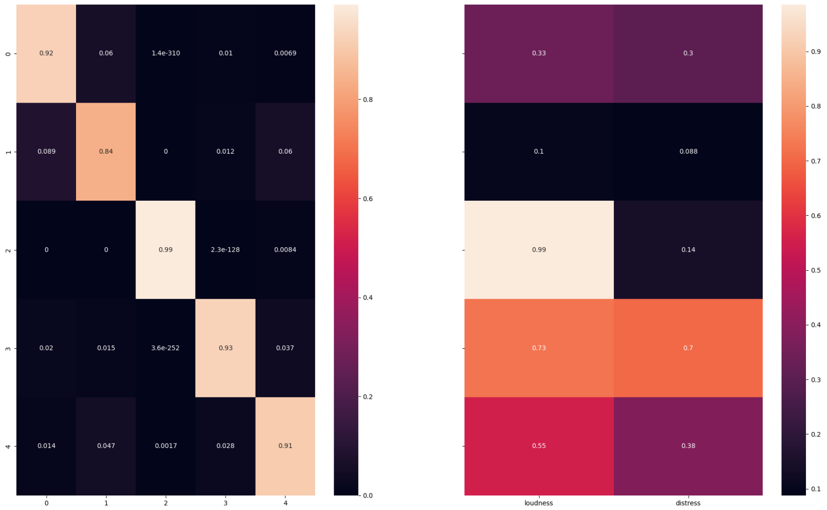


Figure A.3.: The TYT dataset: State means and transition matrices for HMM with 5 states

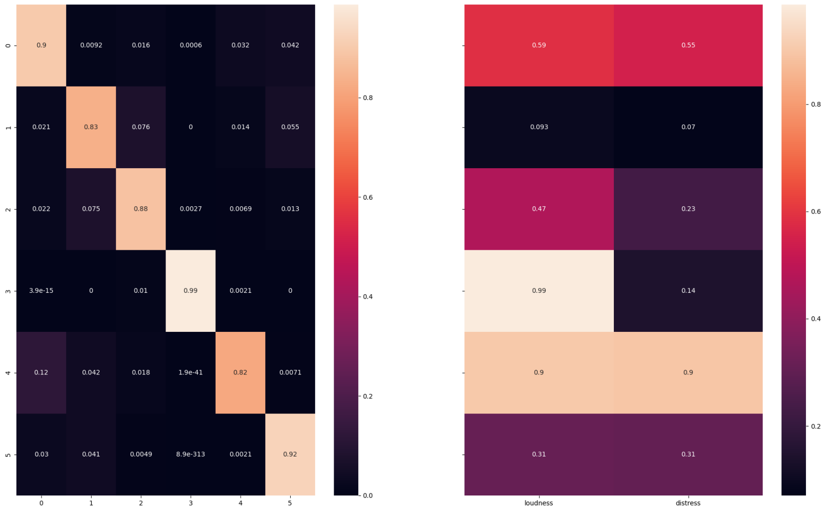


Figure A.4.: The TYT dataset: State means and transition matrices for HMM with 6 states

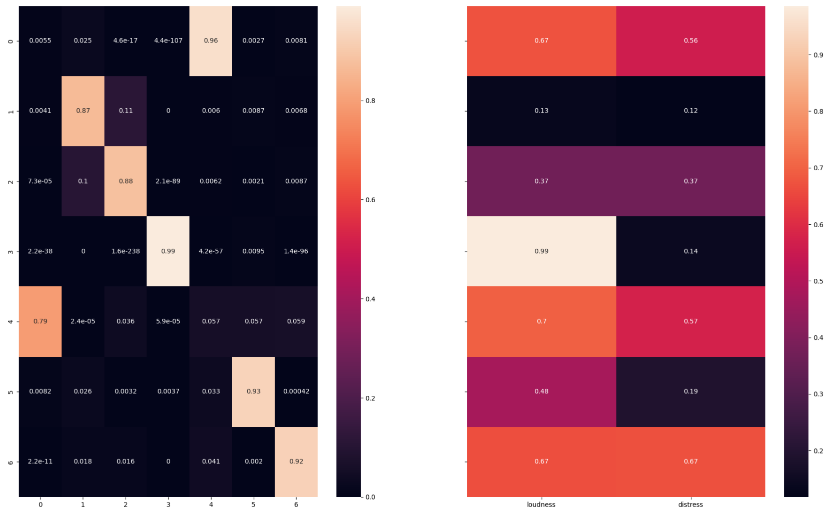


Figure A.5.: The TYT dataset: State means and transition matrices for HMM with 7 states

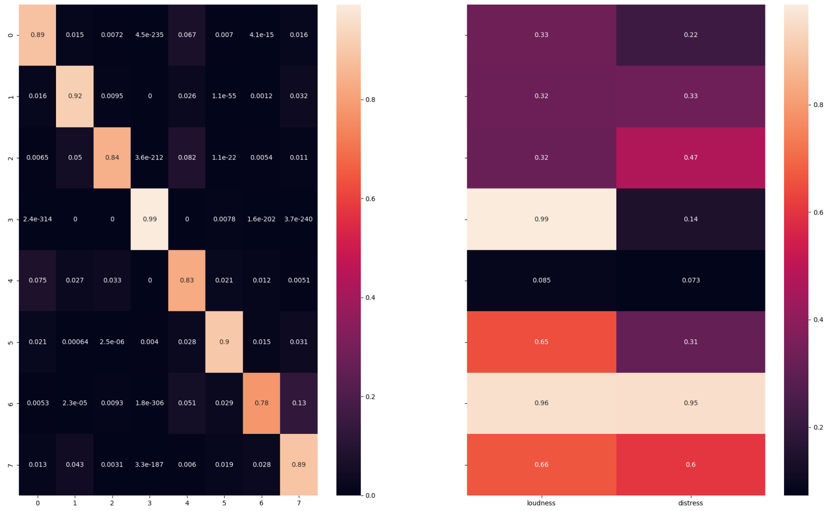


Figure A.6.: The TYT dataset: State means and transition matrices for HMM with 8 states

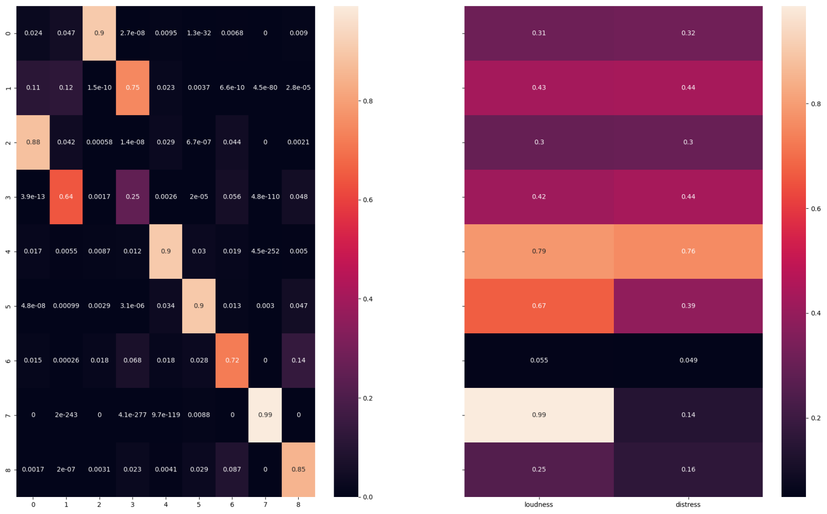


Figure A.7.: The TYT dataset: State means and transition matrices for HMM with 9 states

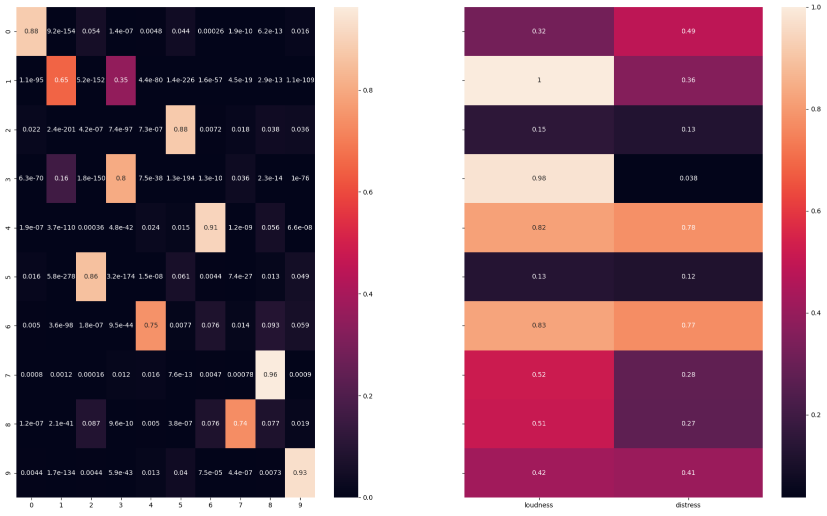


Figure A.8.: The TYT dataset: State means and transition matrices for HMM with 10 states

UNITI Dataset

The following Figures A.9 - A.16 show the state transition matrices (left image), and the corresponding means for loudness, distress for the UNITI dataset.

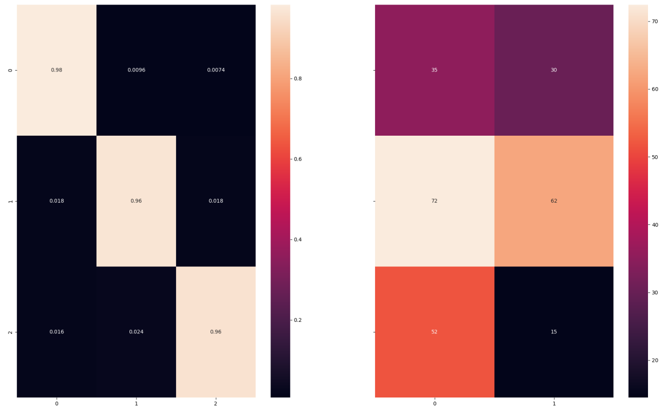


Figure A.9.: The UNITI dataset: State means and transition matrices for HMM with 3 states

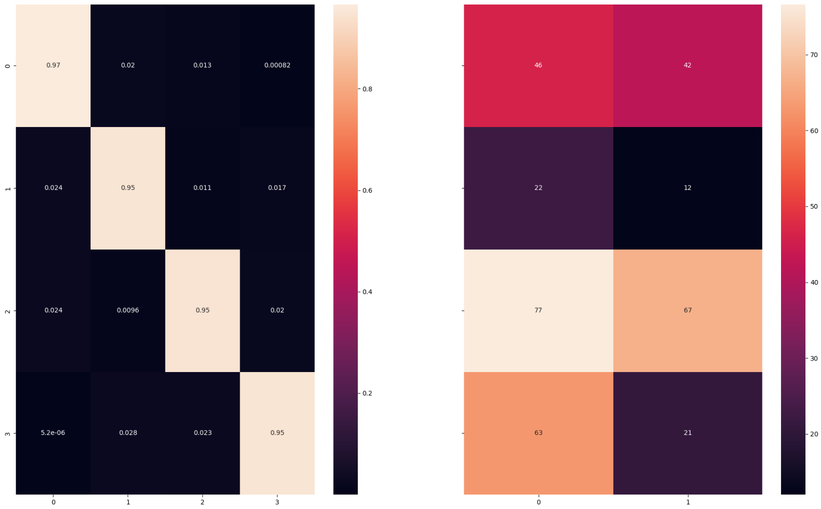


Figure A.10.: The UNITI dataset: State means and transition matrices for HMM with 4 states

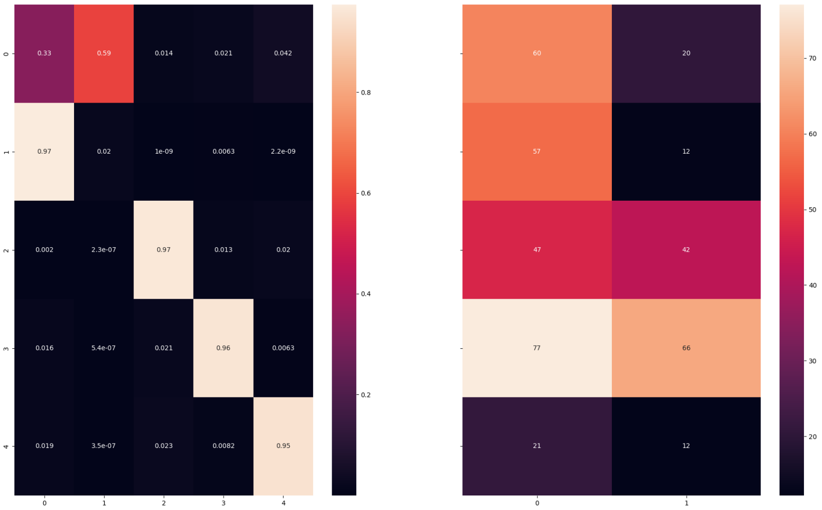


Figure A.11.: The UNITI dataset: State means and transition matrices for HMM with 5 states

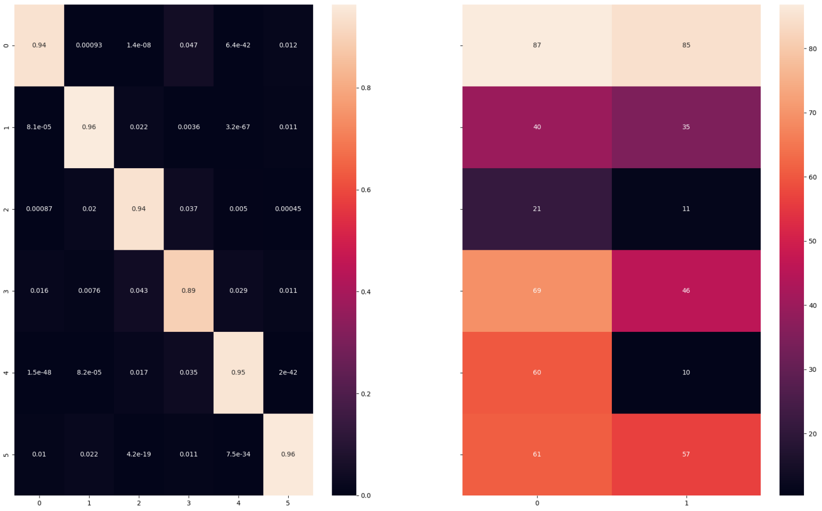


Figure A.12.: The UNITI dataset: State means and transition matrices for HMM with 6 states

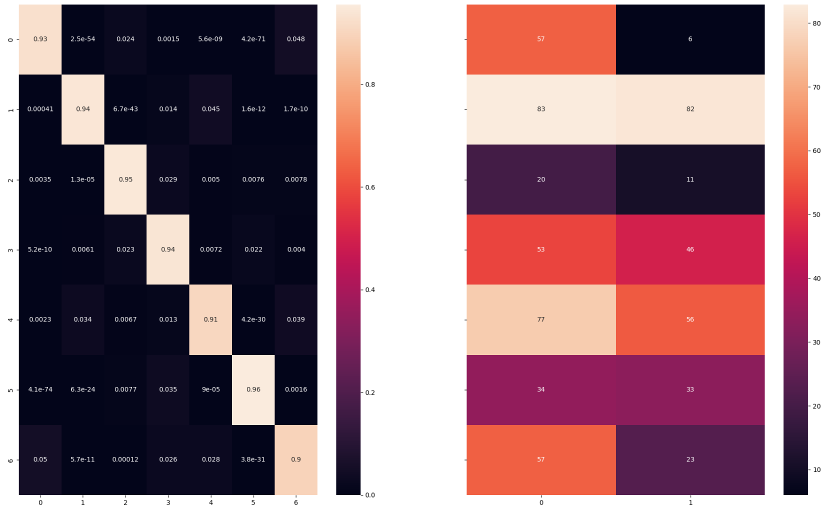


Figure A.13.: The UNITI dataset: State means and transition matrices for HMM with 7 states

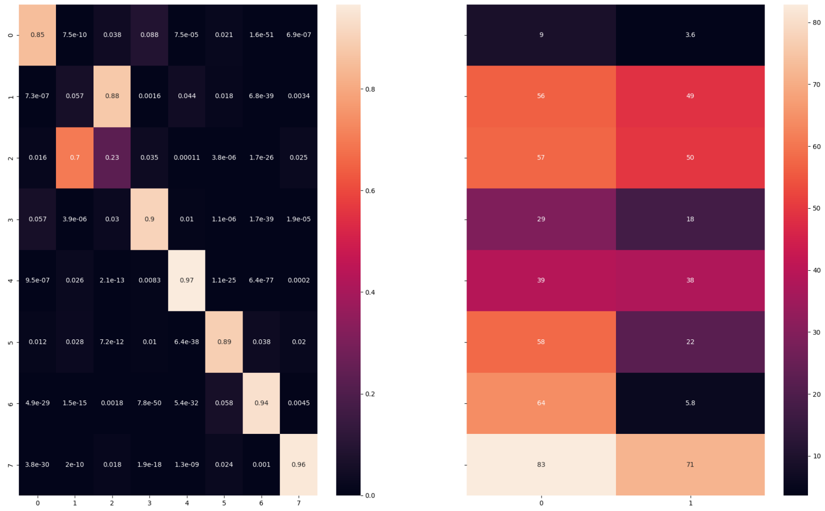


Figure A.14.: The UNITI dataset: State means and transition matrices for HMM with 8 states

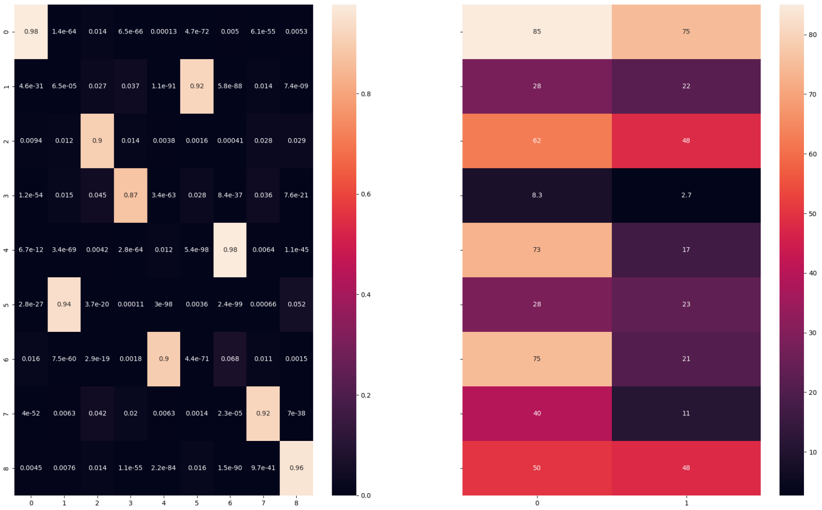


Figure A.15.: The UNITI dataset: State means and transition matrices for HMM with 9 states

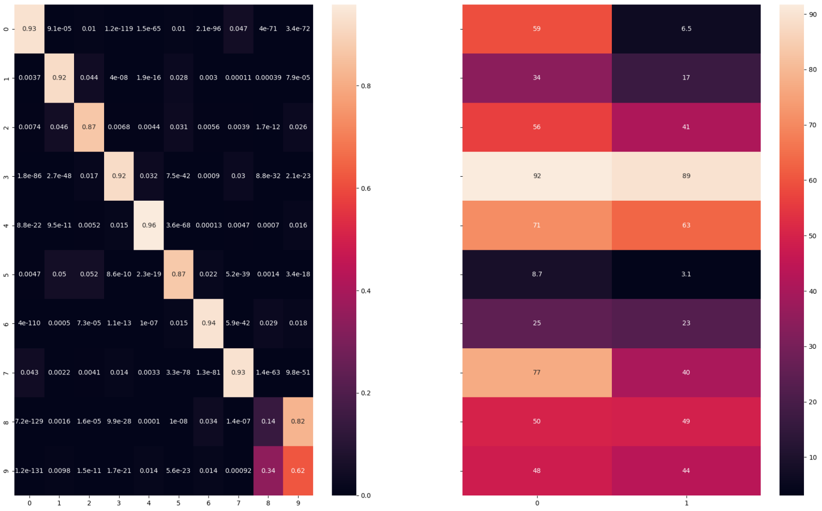


Figure A.16.: The UNITI dataset: State means and transition matrices for HMM with 10 states

A.2.2. Appendix for Section 4.4.2

Results for Granger causality: User-level RMSEs for global vs local model Figure A.17 below shows the boxplot of RMSEs achieved at the user level by the global model versus the personalised (N=1) model that does not use the data of neighbours.

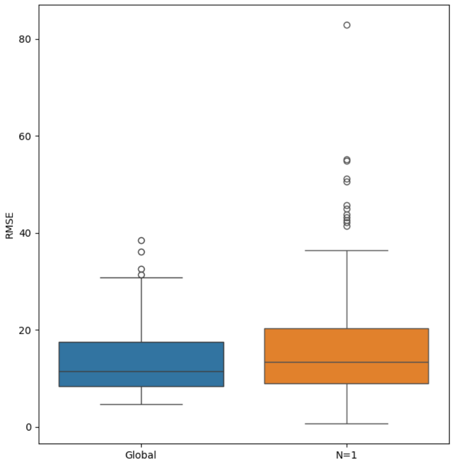


Figure A.17.: The UNITI dataset: Boxplot of user-level RMSEs

Results for Granger causality at the question level: The following two Figures A.18 and A.19 show the results for building the personalised model neighbourhoods on Granger causalities targeted at each variable in the momentary assessment.

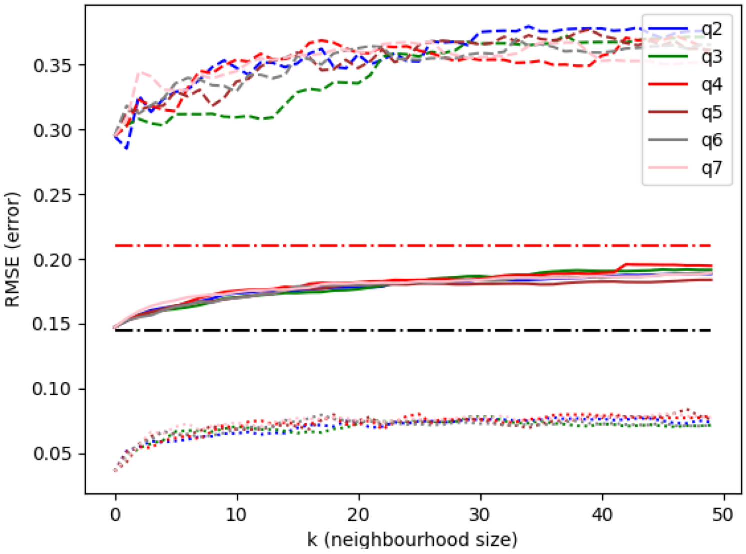


Figure A.18.: The TYT dataset: Restricting the neighbourhood computation to causalities towards a single variable

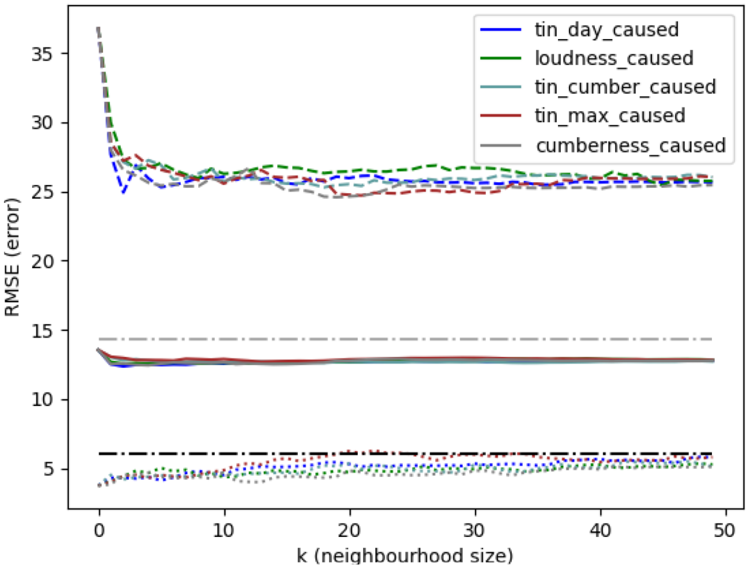


Figure A.19.: The UNITI dataset: Restricting the neighbourhood computation to causalities towards a single variable

A.2.3. Appendix for Section 4.4.4

Similarity based on data v/s metadata: The image A.20 below shows the boxplots for similarities computed for the TYD Spain and Bulgaria datasets. Column on the left shows the cosine similarities over user vectors that describe the number of sessions logged in the first N days. The box plot on the left hand side shows the results for when using the first N days, and the boxplot on the right, for last N days.

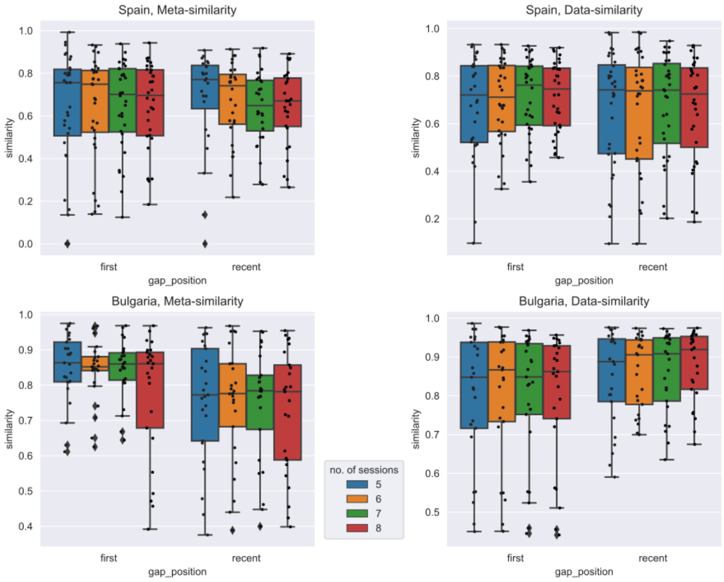


Figure A.20.: Boxplots of similarities measured for user pairs in (U_{long}, U_{short}) for data v/s metadata, first v/s last observations, for N=5 . . . 8.

Model performance for exhaustive search and early termination based on metadata: The following table A.2 shows the results achieved by the metadata-based similarity measure when used by the exhaustive search and early termination logic. It can be seen that the exact choice of data v/s metadata makes little difference to the final performance.

Dataset	Baseline Model (RMSE)	Exhaustive Search (RMSE)	Early Termination (RMSE)
BG	24.300	22.972	24.178
ES	23.165	14.016	14.016

Table A.2.: TYD Bulgaria (BG) and TYD Spain (ES) datasets: Errors achieved by the exhaustive search and early termination method compared to the baseline model

Detailed neighbourhood information for exhaustive search and early termination based on metadata: The following table shows the results for the early termination and exhaustive search for similarity based on metadata (number of user interactions in first 5 days). The results are very similar those found by the data-based similarity methods, although some of the user-level neighbourhoods are different.

Dataset	User_ID	B (RMSE)	ES (RMSE)	ET (RMSE)	|B|	|ES|	|ET|
BG	1	36.26	24.69	31.58	316	145	100
	2	30.07	31.04	31.04		265	265
	3	19.38	20.15	20.15		171	171
	4	23.24	25.39	25.39		171	171
	5	23.47	23.47	23.47		316	316
ES	1	32.14	16.07	16.07	505	190	105
	2	42.82	35.93	35.93		314	90
	3	30.36	7.11	7.11		90	90
	4	23.82	11.3	11.3		84	84
	5	18.38	6.78	6.78		280	280
	6	23.92	11.36	11.36		280	280
	7	21.97	14.52	14.52		195	195
	8	11.77	12.23	12.23		195	195

Table A.3.: User-level RMSEs and training data size for baseline model, exhaustive search, early termination using metadata-guided similarity

A.3. Appendix for Chapter 5.3

	TYT	UNITI
N	227	222
Min. length	30	30
Max. length	841	263
Avg. length	90	64.8
Std. dev.	104.9	37
Date Start	05.2014	04.2021
Date End	01.2022	04.2022

Table A.4.: Properties of the TYT and UNITI datasets used in Chapter 5.3

Bibliography

[1] Amornbunchornvej, Chainarong, Zheleva, Elena, and Berger-Wolf, Tanya. “Variable-lag granger causality and transfer entropy for time series analysis”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15.4 (2021), pp. 1–30 (cit. on p. 54).

[2] Amornbunchornvej, Chainarong, Zheleva, Elena, and Berger-Wolf, Tanya Y. “Variable-lag granger causality for time series analysis”. In: *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2019, pp. 21–30 (cit. on p. 54).

[3] Atlas, Les, Cohn, David, and Ladner, Richard. “Training connectionist networks with queries and selective sampling”. In: *Advances in neural information processing systems* 2 (1989) (cit. on p. 94).

[4] Baguley, David, McFerran, Don, and Hall, Deborah. “Tinnitus”. In: *The Lancet* 382.9904 (2013), pp. 1600–1607 (cit. on pp. 10, 11).

[5] Baguley, David M and Atlas, Marcus D. “Cochlear implants and tinnitus”. In: *Progress in brain research* 166 (2007), pp. 347–355 (cit. on p. 11).

[6] Ban, Tao et al. “Referential k NN Regression for Financial Time Series Forecasting”. In: *Neural Information Processing: 20th International Conference, ICONIP 2013, Daegu, Korea, November 3-7, 2013. Proceedings, Part I 20*. Springer. 2013, pp. 601–608 (cit. on pp. 25, 26).

[7] Bar-Hillel, Aharon et al. “Learning a Mahalanobis metric from equivalence constraints.” In: *Journal of machine learning research* 6.6 (2005) (cit. on p. 31).

[8] Baytas, Inci M et al. “Patient subtyping via time-aware LSTM networks”. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017, pp. 65–74 (cit. on p. 118).

[9] Beukes, Eldré W et al. “Exploring tinnitus heterogeneity”. In: *Progress in brain research* 260 (2021), pp. 79–99 (cit. on p. 58).

[10] Beyer, Christian et al. “Predicting Polarities of Entity-Centered Documents without Reading Their Contents”. In: *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. SAC ’18. Pau, France: Association for Computing Machinery, 2018, pp. 525–528. ISBN: 9781450351911. DOI: `10.1145/3167132.3172870`. URL: `https://doi.org/10.1145/3167132.3172870` (cit. on pp. 26, 36, 37, 43, 61).

[11] Bijnen, EJ et al. “Cluster analysis [electronic resource]: Survey and evaluation of techniques”. In: () (cit. on p. 52).

[12] Bonilla-Escribano, Pablo et al. “Multidimensional variability in ecological assessments predicts two clusters of suicidal patients”. In: *Scientific reports* 13.1 (2023), p. 3546 (cit. on pp. 31, 32).

[13] Bradley, Paul S, Bennett, Kristin P, and Demiriz, Ayhan. “Constrained k-means clustering”. In: *Microsoft Research, Redmond* 20.0 (2000) (cit. on p. 31).

[14] Buechi, Rahel et al. “Evidence assessing the diagnostic performance of medical smartphone apps: a systematic review and exploratory meta-analysis”. In: *BMJ open* 7.12 (2017) (cit. on p. 14).

[15] Chartrand, Gary, Kubicki, Grzegorz, and Schultz, Michelle. “Graph similarity and distance in graphs”. In: *Aequationes Mathematicae* 55.1 (1998), pp. 129–145 (cit. on p. 118).

[16] Cima, Rilana FF et al. “Specialised treatment based on cognitive behaviour therapy versus usual care for tinnitus: a randomised controlled trial”. In: *The Lancet* 379.9830 (2012), pp. 1951–1959 (cit. on pp. 11, 12).

[17] Crummer, Richard W and Hassan, Ghinwa A. “Diagnostic approach to tinnitus”. In: *American family physician* 69.1 (2004), pp. 120–126 (cit. on p. 10).

[18] Csikszentmihalyi, Mihaly and Larson, Reed. “Validity and reliability of the experience-sampling method”. In: *The Journal of nervous and mental disease* 175.9 (1987), pp. 526–536 (cit. on p. 15).

[19] Czyz, Ewa K et al. “Ecological Momentary Assessments and Passive Sensing in the Prediction of Short-Term Suicidal Ideation in Young Adults”. In: *JAMA Network Open* 6.8 (2023), e2328005–e2328005 (cit. on p. 51).

[20] Davis, Paul B, Paki, Bardia, and Hanley, Peter J. “Neuromonics tinnitus treatment: third clinical trial”. In: *Ear and hearing* 28.2 (2007), pp. 242–259 (cit. on p. 11).

[21] De Ridder, Dirk et al. “Transcranial magnetic stimulation and extradural electrodes implanted on secondary auditory cortex for tinnitus suppression”. In: *Journal of neurosurgery* 114.4 (2011), pp. 903–911 (cit. on p. 11).

[22] Deng, Zhenyun et al. “Efficient kNN classification algorithm for big data”. In: *Neurocomputing* 195 (2016), pp. 143–148 (cit. on p. 31).

[23] El Aboudi, Naoual and Benhlima, Laila. “Review on wrapper feature selection approaches”. In: *2016 International Conference on Engineering & MIS (ICEMIS)*. IEEE. 2016, pp. 1–5 (cit. on p. 95).

[24] Flores Mateo, Gemma et al. “Mobile phone apps to promote weight loss and increase physical activity: a systematic review and meta-analysis”. In: *Journal of medical Internet research* 17.11 (2015), e253 (cit. on p. 14).

[25] Fredrickson, Barbara L. “Extracting meaning from past affective experiences: The importance of peaks, ends, and specific emotions”. In: *Cognition & Emotion* 14.4 (2000), pp. 577–606 (cit. on p. 15).

[26] Freeman, John R. “Granger causality and the times series analysis of political relationships”. In: *American Journal of Political Science* (1983), pp. 327–358 (cit. on p. 53).

[27] Fujita, André et al. “Functional clustering of time series gene expression data by Granger causality”. In: *BMC systems biology* 6 (2012), pp. 1–12 (cit. on pp. 53, 54).

[28] Geler, Zoltan et al. “Dynamic time warping: Itakura vs sakoe-chiba”. In: *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*. IEEE. 2019, pp. 1–6 (cit. on p. 52).

[29] Ghassempour, Shima, Girosi, Federico, and Maeder, Anthony. “Clustering multivariate time series using hidden Markov models”. In: *International journal of environmental research and public health* 11.3 (2014), pp. 2741–2763 (cit. on p. 53).

[30] Goetz, Laura H and Schork, Nicholas J. “Personalized medicine: motivation, challenges, and progress”. In: *Fertility and sterility* 109.6 (2018), pp. 952–963 (cit. on p. 3).

[31] Granger, Clive WJ. "Investigating causal relations by econometric models and cross-spectral methods". In: *Econometrica: journal of the Econometric Society* (1969), pp. 424–438 (cit. on p. 53).

[32] Gunopulos, Dimitrios and Das, Gautam. "Time series similarity measures (tutorial pm-2)". In: *Tutorial notes of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2000, pp. 243–307 (cit. on p. 52).

[33] Henry, James A et al. "Guide to conducting tinnitus retraining therapy initial and follow-up interviews." In: *Journal of Rehabilitation Research & Development* 40.2 (2003) (cit. on p. 11).

[34] Henry, James A et al. "Pilot study to evaluate ecological momentary assessment of tinnitus". In: *Ear and hearing* 32.2 (2012), p. 179 (cit. on p. 15).

[35] Hiller, Wolfgang and Goebel, Gerhard. "Rapid assessment of tinnitus-related psychological distress using the Mini-TQ". In: *Int J Audiol* 43.10 (2004), pp. 600–604 (cit. on p. 17).

[36] Hiller, Wolfgang and Goebel, Gerhard. "When tinnitus loudness and annoyance are discrepant: audiological characteristics and psychological profile". In: *Audiology and Neurotology* 12.6 (2007), pp. 391–400 (cit. on pp. 32, 33, 46).

[37] Hoare, Derek J et al. "Systematic review and meta-analyses of randomized controlled trials examining tinnitus management". In: *The Laryngoscope* 121.7 (2011), pp. 1555–1564 (cit. on p. 11).

[38] Hosenfeld, Bettina et al. "Major depressive disorder as a nonlinear dynamic system: bimodality in the frequency distribution of depressive symptoms over time". In: *Bmc psychiatry* 15.1 (2015), pp. 1–9 (cit. on p. 53).

[39] Huddar, Vijay et al. "Predicting complications in critical care using heterogeneous clinical data". In: *IEEE Access* 4 (2016), pp. 7988–8001 (cit. on p. 3).

[40] Hulme, William J et al. "Cluster hidden markov models: An application to ecological momentary assessment of schizophrenia". In: *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE. 2019, pp. 99–103 (cit. on p. 53).

[41] Hulme, William J et al. "Adaptive symptom monitoring using hidden markov models–an application in ecological momentary assessment". In: *IEEE Journal of Biomedical and Health Informatics* 25.5 (2020), pp. 1770–1780 (cit. on p. 53).

[42] Jamaludeen, Noor et al. "Circadian conditional granger causalities on ecological momentary assessment data from an mhealth app". In: *2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE. 2021, pp. 354–359 (cit. on pp. 54, 58, 112).

[43] Jastreboff, Pawel J. "Tinnitus retraining therapy". In: *Textbook of tinnitus* (2011), pp. 575–596 (cit. on p. 12).

[44] Jia, Yao, Zhou, Chongyu, and Motani, Mehul. "Spatio-temporal autoencoder for feature learning in patient data with missing observations". In: *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE. 2017, pp. 886–890 (cit. on p. 118).

[45] Jung, Martin and Zscheischler, Jakob. "A guided hybrid genetic algorithm for feature selection with expensive cost functions". In: *Procedia Computer Science* 18 (2013), pp. 2337–2346 (cit. on p. 95).

[46] Kabbur, Santosh, Ning, Xia, and Karypis, George. "Fism: factored item similarity models for top-n recommender systems". In: *Proceedings of the 19th*

ACM SIGKDD international conference on Knowledge discovery and data mining. 2013, pp. 659–667 (cit. on p. 24).

[47] Kalle, Sven. “A systematic overview of internet-based and smartphonebased services for tinnitus diagnostics and treatment”. In: (2016) (cit. on p. 16).

[48] Kennel, Matthew B, Brown, Reggie, and Abarbanel, Henry DI. “Determining embedding dimension for phase-space reconstruction using a geometrical construction”. In: *Physical review A* 45.6 (1992), p. 3403 (cit. on p. 44).

[49] Keogh, Eamonn et al. “Dimensionality reduction for fast similarity search in large time series databases”. In: *Knowledge and information Systems* 3 (2001), pp. 263–286 (cit. on p. 53).

[50] Kini, B Venkataramana and Sekhar, C Chandra. “Large margin mixture of AR models for time series classification”. In: *Applied Soft Computing* 13.1 (2013), pp. 361–371 (cit. on p. 53).

[51] Knitza, Johannes et al. “Mobile health usage, preferences, barriers, and eHealth literacy in rheumatology: patient survey study”. In: *JMIR mHealth and uHealth* 8.8 (2020), e19661 (cit. on p. 2).

[52] Kraft, Robin et al. “Combining mobile crowdsensing and ecological momentary assessments in the healthcare domain”. In: *Frontiers in neuroscience* 14 (2020), p. 164 (cit. on p. 19).

[53] Krauss, Tom et al. *Explainable AI - Requirements, Use Cases and Solutions.* Tech. rep. Steinplatz 1, 10623 Berlin, Germany: Technology Programme AI Innovation Competition of the Federal Ministry for Economic Affairs and Climate Action Accompanying research, 2022. URL: `https://www.digitale-technologien.de/DT/Redaktion/EN/Downloads/Publikation/KI_Inno_Xai_Studie.pdf` (cit. on p. 1).

[54] Langguth, Berthold et al. “Consensus for tinnitus patient assessment and treatment outcome measurement: Tinnitus Research Initiative meeting, Regensburg, July 2006”. In: *Progress in brain research* 166 (2007), pp. 525–536 (cit. on p. 17).

[55] Langguth, Berthold et al. “Tinnitus: causes and clinical management”. In: *The Lancet Neurology* 12.9 (2013), pp. 920–930 (cit. on pp. 10–12).

[56] Le, Quoc and Mikolov, Tomas. “Distributed Representations of Sentences and Documents”. In: *Proceedings of the 31st International Conference on Machine Learning.* Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, June 2014, pp. 1188–1196. URL: `https://proceedings.mlr.press/v32/le14.html` (cit. on p. 37).

[57] Lewicki, Michael and Sejnowski, Terrence J. “Coding time-varying signals using sparse, shift-invariant representations”. In: *Advances in neural information processing systems* 11 (1998) (cit. on p. 52).

[58] Lin, Jessica, Khade, Rohan, and Li, Yuan. “Rotation-invariant similarity in time series using bag-of-patterns representation”. In: *Journal of Intelligent Information Systems* 39 (2012), pp. 287–315 (cit. on p. 53).

[59] Lin, Jessica et al. “A symbolic representation of time series, with implications for streaming algorithms”. In: *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery.* 2003, pp. 2–11 (cit. on p. 53).

[60] Linden, Greg, Smith, Brent, and York, Jeremy. “Amazon. com recommendations: Item-to-item collaborative filtering”. In: *IEEE Internet computing* 7.1 (2003), pp. 76–80 (cit. on pp. 23, 24).

[61] Liu, Zitao and Hauskrecht, Milos. "A personalized predictive framework for multivariate clinical time series via adaptive model selection". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management.* 2017, pp. 1169–1177 (cit. on pp. 3, 26).

[62] Lora, Alicia Troncoso et al. "Electricity market price forecasting: Neural networks versus weighted-distance k nearest neighbours". In: *Database and Expert Systems Applications: 13th International Conference, DEXA 2002 Aix-en-Provence, France, September 2–6, 2002 Proceedings 13.* Springer. 2002, pp. 321–330 (cit. on p. 26).

[63] Lora, Alicia Troncoso et al. "Electricity market price forecasting based on weighted nearest neighbors techniques". In: *IEEE Transactions on Power Systems* 22.3 (2007), pp. 1294–1301 (cit. on p. 44).

[64] Makridakis, Spyros, Spiliotis, Evangelos, and Assimakopoulos, Vassilios. "The M5 competition: Background, organization, and implementation". In: *International Journal of Forecasting* 38.4 (2022), pp. 1325–1336 (cit. on pp. 1, 119).

[65] McAuley, Julian. *Personalized machine learning.* Cambridge University Press, 2022 (cit. on pp. 2, 23).

[66] McAuley, Julian and Yang, Alex. "Addressing complex and subjective product-related queries with customer reviews". In: *Proceedings of the 25th International Conference on World Wide Web.* 2016, pp. 625–635 (cit. on p. 36).

[67] McInnes, Leland, Healy, John, and Melville, James. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018) (cit. on p. 59).

[68] Monarch, Robert Munro. *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI.* Simon and Schuster, 2021 (cit. on p. 94).

[69] Moutafis, Panagiotis, Leng, Mengjun, and Kakadiaris, Ioannis A. "Regression-based metric learning". In: *2016 23rd International Conference on Pattern Recognition (ICPR).* IEEE. 2016, pp. 2700–2705 (cit. on p. 119).

[70] Mueen, Abdullah et al. "AWarp: Fast warping distance for sparse time series". In: *2016 IEEE 16th International Conference on Data Mining (ICDM).* IEEE. 2016, pp. 350–359 (cit. on p. 52).

[71] Narendra and Fukunaga. "A branch and bound algorithm for feature subset selection". In: *IEEE Transactions on computers* 100.9 (1977), pp. 917–922 (cit. on p. 95).

[72] Nayak, Neha, Angeli, Gabor, and Manning, Christopher D. "Evaluating word embeddings using a representative suite of practical tasks". In: *Proceedings of the 1st workshop on evaluating vector-space representations for nlp.* 2016, pp. 19–23 (cit. on p. 42).

[73] Ng, Nathan H et al. "Predicting surgery duration with neural heteroscedastic regression". In: *Machine Learning for Healthcare Conference.* PMLR. 2017, pp. 100–111 (cit. on p. 25).

[74] Ni, Jianmo, Muhlstein, Larry, and McAuley, Julian. "Modeling heart rate and activity data for personalized fitness recommendation". In: *The World Wide Web Conference.* 2019, pp. 1343–1353 (cit. on p. 25).

[75] Nikolentzos, Giannis, Meladianos, Polykarpos, and Vazirgiannis, Michalis. "Matching node embeddings for graph similarity". In: *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 31. 1. 2017 (cit. on p. 118).

[76] Ning, Xia and Karypis, George. “Slim: Sparse linear methods for top-n recommender systems”. In: *2011 IEEE 11th international conference on data mining*. IEEE. 2011, pp. 497–506 (cit. on p. 24).

[77] Nondahl, David M et al. “Tinnitus and its risk factors in the Beaver Dam offspring study”. In: *International journal of audiology* 50.5 (2011), pp. 313–320 (cit. on p. 10).

[78] Okamoto, Hidehiko et al. “Listening to tailor-made notched music reduces tinnitus loudness and tinnitus-related auditory cortex activity”. In: *Proceedings of the National Academy of Sciences* 107.3 (2010), pp. 1207–1210 (cit. on p. 11).

[79] Oreshkin, Boris N et al. “N-BEATS: Neural basis expansion analysis for interpretable time series forecasting”. In: *arXiv preprint arXiv:1905.10437* (2019) (cit. on p. 1).

[80] Organization, World Health et al. “mHealth: new horizons for health through mobile technologies.” In: *mHealth: new horizons for health through mobile technologies.* (2011) (cit. on pp. 13, 14).

[81] Pedregosa, F. et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on p. 35).

[82] Phillips, John S and McFerran, Don. “Tinnitus retraining therapy (TRT) for tinnitus”. In: *Cochrane database of systematic reviews* 3 (2010) (cit. on p. 12).

[83] Pratella, David et al. “A survey of autoencoder algorithms to pave the diagnosis of rare diseases”. In: *International journal of molecular sciences* 22.19 (2021), p. 10891 (cit. on p. 118).

[84] Probst, Thomas et al. “Emotion dynamics and tinnitus: Daily life data from the “TrackYourTinnitus” application”. In: *Scientific reports* 6.1 (2016), p. 31166 (cit. on p. 15).

[85] Probst, Thomas et al. “Emotional states as mediators between tinnitus loudness and tinnitus distress in daily life: Results from the “TrackYourTinnitus” application”. In: *Scientific reports* 6.1 (2016), p. 20382 (cit. on p. 15).

[86] Probst, Thomas et al. “Does tinnitus depend on time-of-day? An ecological momentary assessment study with the “TrackYourTinnitus” application”. In: *Frontiers in aging neuroscience* 9 (2017), p. 253 (cit. on pp. 15, 58).

[87] Pryss, Rüdiger. “Mobile Crowdsensing in Healthcare Scenarios: Taxonomy, Conceptual Pillars, Smart Mobile Crowdsensing Services”. In: *Digital Phenotyping and Mobile Sensing: New Developments in Psychoinformatics* (2019), pp. 221–234 (cit. on p. 19).

[88] Pryss, Rüdiger. “Mobile crowdsensing in healthcare scenarios: Taxonomy, conceptual pillars, smart mobile crowdsensing services”. In: *Digital Phenotyping and Mobile Sensing: New Developments in Psychoinformatics*. Springer, 2022, pp. 305–320 (cit. on pp. 14, 15).

[89] Pryss, Rüdiger et al. “Mobile crowd sensing services for tinnitus assessment, therapy, and research”. In: *2015 IEEE international conference on mobile services*. IEEE. 2015, pp. 352–359 (cit. on pp. 16, 17).

[90] Al-Qahtani, Fahad H and Crone, Sven F. “Multivariate k-nearest neighbour regression for time series data—A novel algorithm for forecasting UK electricity demand”. In: *The 2013 international joint conference on neural networks (IJCNN)*. IEEE. 2013, pp. 1–8 (cit. on p. 26).

[91] Qin, Yue et al. “Research progress on semi-supervised clustering”. In: *Cognitive Computation* 11 (2019), pp. 599–612 (cit. on p. 31).

[92] Rajaraman, Anand and Ullman, Jeffrey David. “Data Mining”. In: *Mining of Massive Datasets*. Cambridge University Press, 2011, pp. 1–17 (cit. on p. 60).

[93] Ramos, Patrícia and Oliveira, José Manuel. “Robust Sales forecasting Using Deep Learning with Static and Dynamic Covariates”. In: *Applied System Innovation* 6.5 (2023), p. 85 (cit. on p. 1).

[94] Reddy, Chandan K and Aggarwal, Charu C. *Healthcare data analytics*. Vol. 36. CRC Press, 2015 (cit. on p. 3).

[95] Ren, Pengzhen et al. “A survey of deep active learning”. In: *ACM computing surveys (CSUR)* 54.9 (2021), pp. 1–40 (cit. on p. 94).

[96] Rivera-Romero, Octavio et al. “Designing personalised mHealth solutions: An overview”. In: *Journal of Biomedical Informatics* (2023), p. 104500 (cit. on p. 2).

[97] Roglic, Gojka. “WHO Global report on diabetes: A summary”. In: *International Journal of Noncommunicable Diseases* 1.1 (2016), pp. 3–8 (cit. on pp. 12, 13).

[98] Rowland, Simon P et al. “What is the clinical value of mHealth for patients?” In: *NPJ digital medicine* 3.1 (2020), p. 4 (cit. on pp. 2, 13, 14).

[99] Saadatfar, Hamid et al. “A new K-nearest neighbors classifier for big data based on efficient data pruning”. In: *Mathematics* 8.2 (2020), p. 286 (cit. on p. 31).

[100] Sahraeian, Sayed Mohammad Ebrahim and Yoon, Byung-Jun. “A novel low-complexity HMM similarity measure”. In: *IEEE Signal Processing Letters* 18.2 (2010), pp. 87–90 (cit. on p. 53).

[101] Sakoe, Hiroaki and Chiba, Seibi. “Dynamic programming algorithm optimization for spoken word recognition”. In: *IEEE transactions on acoustics, speech, and signal processing* 26.1 (1978), pp. 43–49 (cit. on p. 52).

[102] Schaette, Roland et al. “Acoustic stimulation treatments against tinnitus could be most effective when tinnitus pitch is within the stimulated frequency range”. In: *Hearing Research* 269.1-2 (2010), pp. 95–101 (cit. on p. 11).

[103] Schlee, Winfried et al. “Measuring the moment-to-moment variability of tinnitus: the TrackYourTinnitus smart phone app”. In: *Frontiers in aging neuroscience* 8 (2016), p. 294 (cit. on p. 15).

[104] Schlee, Winfried et al. “Momentary assessment of tinnitus—How smart mobile applications advance our understanding of tinnitus”. In: *Digital Phenotyping and Mobile Sensing: New Developments in Psychoinformatics*. Springer, 2022, pp. 285–303 (cit. on p. 15).

[105] Schleicher, Miro et al. “Understanding adherence to the recording of ecological momentary assessments in the example of tinnitus monitoring”. In: *Scientific Reports* 10.1 (2020), p. 22459 (cit. on p. 14).

[106] Schleicher, Miro et al. “Expect the gap: A recommender approach to estimate the absenteeism of self-monitoring mHealth app users”. In: *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2022, pp. 1–10 (cit. on p. 14).

[107] Schleicher, Miro et al. “When can I expect the mHealth user to return? Prediction meets time series with gaps”. In: *International Conference on Artificial Intelligence in Medicine*. Springer. 2022, pp. 310–320 (cit. on p. 14).

[108] Schneider, Stefan et al. “II. Indices of pain intensity derived from ecological momentary assessments and their relationships with patient functioning: an individual patient data meta-analysis”. In: *The journal of pain* 22.4 (2021), pp. 371–385 (cit. on p. 32).

[109] Schoisswohl, Stefan et al. “Unification of Treatments and Interventions for Tinnitus Patients (UNITI): a study protocol for a multi-center randomized clinical trial”. In: *Trials* 22 (2021), pp. 1–16 (cit. on p. 18).

[110] Schwenker, Friedhelm and Trentin, Edmondo. “Pattern classification and clustering: A review of partially supervised learning approaches”. In: *Pattern Recognition Letters* 37 (2014), pp. 4–14 (cit. on p. 31).

[111] Seabold, Skipper and Perktold, Josef. “statsmodels: Econometric and statistical modeling with python”. In: *9th Python in Science Conference.* 2010 (cit. on p. 70).

[112] Semigran, Hannah L et al. “Evaluation of symptom checkers for self diagnosis and triage: audit study”. In: *bmj* 351 (2015) (cit. on p. 14).

[113] Settles, Burr. “Active learning literature survey”. In: (2009) (cit. on p. 94).

[114] Settles, Burr and Craven, Mark. “An analysis of active learning strategies for sequence labeling tasks”. In: *proceedings of the 2008 conference on empirical methods in natural language processing.* 2008, pp. 1070–1079 (cit. on p. 94).

[115] Sharabiani, Anooshiravan et al. “Asymptotic dynamic time warping calculation with utilizing value repetition”. In: *Knowledge and Information Systems* 57 (2018), pp. 359–388 (cit. on p. 52).

[116] Smit, AC and Helmich, MA. “Personalized detection of impending symptom transitions in depression using early warning signals during and shortly after antidepressant discontinuation”. In: *What’s in a mood?* (2021), p. 181 (cit. on p. 51).

[117] Smith, Brent and Linden, Greg. “Two decades of recommender systems at Amazon. com”. In: *Ieee internet computing* 21.3 (2017), pp. 12–18 (cit. on p. 24).

[118] Strasser, Ben, Botea, Adi, and Harabor, Daniel. “Compressing optimal paths with run length encoding”. In: *Journal of Artificial Intelligence Research* 54 (2015), pp. 593–629 (cit. on p. 52).

[119] Tass, Peter A et al. “Counteracting tinnitus by acoustic coordinated reset neuromodulation”. In: *Restorative neurology and neuroscience* 30.2 (2012), pp. 137–159 (cit. on p. 11).

[120] Triguero, Isaac, García, Salvador, and Herrera, Francisco. “Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study”. In: *Knowledge and Information systems* 42 (2015), pp. 245–284 (cit. on p. 95).

[121] Troncoso Lora, Alicia et al. “Influence of kNN-based load forecasting errors on optimal energy production”. In: *Progress in Artificial Intelligence: 11th Portuguese Conference on Artificial Intelligence, EPIA 2003, Beja, Portugal, December 4-7, 2003. Proceedings 11.* Springer. 2003, pp. 189–203 (cit. on p. 26).

[122] Troncoso Lora, Alicia et al. “Time-series prediction: Application to the short-term electric energy demand”. In: *Current Topics in Artificial Intelligence: 10th Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2003, and 5th Conference on Technology Transfer, TTIA 2003, San Sebastian, Spain, November 12-14, 2003. Revised Selected Papers.* Springer. 2004, pp. 577–586 (cit. on p. 26).

[123] Unnikrishnan, V. et al. “Entity-Level Stream Classification: Exploiting Entity Similarity to Label the Future Observations Referring to an Entity”. In: *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA).* 2018, pp. 246–255. DOI: 10.1109/DSAA.2018.00035 (cit. on pp. 23, 37).

[124] Unnikrishnan, V. et al. “Love thy Neighbours: A Framework for Error-Driven Discovery of Useful Neighbourhoods for One-Step Forecasts on EMA data”. In: *2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS)*. 2021, pp. 295–300. DOI: `10.1109/CBMS52027.2021.00080` (cit. on p. 51).

[125] Unnikrishnan, Vishnu et al. “Entity-level stream classification: exploiting entity similarity to label the future observations referring to an entity”. In: *International Journal of Data Science and Analytics* 9.1 (2020), pp. 1–15. ISSN: 2364-4168. DOI: `10.1007/s41060-019-00177-1`. URL: `https://doi.org/10.1007/s41060-019-00177-1` (cit. on pp. 23, 32, 37, 45).

[126] Unnikrishnan, Vishnu et al. “Predicting the Health Condition of mHealth App Users with Large Differences in the Number of Recorded Observations - Where to Learn from?” In: *Discovery Science*. Ed. by Annalisa Appice et al. Cham: Springer International Publishing, 2020, pp. 659–673 (cit. on pp. 20, 51, 67, 82, 84, 85).

[127] Unnikrishnan, Vishnu et al. *The Effect of Non-Personalised Tips on the Continued Use of Self-Monitoring mHealth Applications*. 2020. DOI: `10.3390/brainsci10120924` (cit. on pp. 17, 18).

[128] Unnikrishnan, Vishnu et al. “A Similarity-Guided Framework for Error-Driven Discovery of Patient Neighbourhoods in EMA Data”. In: *Advances in Intelligent Data Analysis XXI*. Cham: Springer Nature Switzerland, 2023, pp. 459–471 (cit. on p. 93).

[129] Van Engelen, Jesper E and Hoos, Holger H. “A survey on semi-supervised learning”. In: *Machine learning* 109.2 (2020), pp. 373–440 (cit. on p. 94).

[130] Vanneste, Sven et al. “Does enriched acoustic environment in humans abolish chronic tinnitus clinically and electrophysiologically? A double blind placebo controlled study”. In: *Hearing Research* 296 (2013), pp. 141–148 (cit. on p. 11).

[131] Vogel, Carsten et al. “UNITI Mobile—EMI-Apps for a Large-Scale European Study on Tinnitus”. In: *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE. 2021, pp. 2358–2362 (cit. on p. 18).

[132] Wagstaff, Kiri et al. “Constrained k-means clustering with background knowledge”. In: *Icml*. Vol. 1. 2001, pp. 577–584 (cit. on p. 31).

[133] Wang, Shirley B et al. “A pilot study using frequent inpatient assessments of suicidal thinking to predict short-term postdischarge suicidal behavior”. In: *JAMA network open* 4.3 (2021), e210591–e210591 (cit. on p. 32).

[134] Wichers, Marieke, Smit, Arnout C, and Snippe, Evelien. “Early warning signals based on momentary affect dynamics can expose nearby transitions in depression: A confirmatory single-subject time-series study”. In: *Journal for person-oriented research* 6.1 (2020), p. 1 (cit. on p. 51).

[135] Wu, Yuan et al. “Mobile app-based interventions to support diabetes self-management: a systematic review of randomized controlled trials to identify functions associated with glycemic efficacy”. In: *JMIR mHealth and uHealth* 5.3 (2017), e6522 (cit. on p. 14).

[136] Xiong, Yimin and Yeung, Dit-Yan. “Time series clustering with ARMA mixtures”. In: *Pattern Recognition* 37.8 (2004), pp. 1675–1689 (cit. on p. 53).

[137] Yang, Dandan et al. “Granger causality for multivariate time series classification”. In: *2017 IEEE international conference on big knowledge (ICBK)*. IEEE. 2017, pp. 103–110 (cit. on p. 54).

[138] Yang, Jaewon et al. “Finding progression stages in time-evolving event sequences”. In: *Proceedings of the 23rd international conference on World wide web*. 2014, pp. 783–794 (cit. on pp. 24, 27).

[139] Yarowsky, David. “Unsupervised word sense disambiguation rivaling supervised methods”. In: *33rd annual meeting of the association for computational linguistics*. 1995, pp. 189–196 (cit. on p. 95).

[140] Zhang, Zheng et al. “Dynamic time warping under limited warping path length”. In: *Information Sciences* 393 (2017), pp. 91–107 (cit. on p. 52).

[141] Zhao, Feng et al. “A similarity measurement for time series and its application to the stock market”. In: *Expert Systems with Applications* 182 (2021), p. 115217 (cit. on p. 52).

[142] Zhu, Xiaofeng, Zhang, Lei, and Huang, Zi. “A sparse embedding and least variance encoding approach to hashing”. In: *IEEE transactions on image processing* 23.9 (2014), pp. 3737–3750 (cit. on p. 31).

Ehrenerklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; verwendete fremde und eigene Quellen sind als solche kenntlich gemacht. Insbesondere habe ich nicht die Hilfe eines kommerziellen Promotionsberaters in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Ich habe insbesondere nicht wissentlich:

- Ergebnisse erfunden oder widersprüchliche Ergebnisse verschwiegen,
- statistische Verfahren absichtlich missbraucht, um Daten in ungerechtfertigter Weise zu interpretieren,
- fremde Ergebnisse oder Veröffentlichungen plagiiert,
- fremde Forschungsergebnisse verzerrt wiedergegeben.

Mir ist bekannt, dass Verstöße gegen das Urheberrecht Unterlassungs- und Schadensersatzansprüche des Urhebers sowie eine strafrechtliche Ahndung durch die Strafverfolgungsbehörden begründen kann. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Magdeburg, den 20.02.2024

Vishnu Mazhuvancherry Unnikrishnan