EARLY DETECTION OF HEALTH CHANGES IN THE ELDERLY USING IN-HOME MULTI-SENSOR DATA STREAMS

A Dissertation

presented to

the Faculty of the Graduate School

at the University of Missouri

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

---

by

WENLONG WU

Dr. James M. Keller, Dissertation Supervisor

May 2022

The undersigned, appointed by the dean of the Graduate School have examined the Dissertation entitled

EARLY DETECTION OF HEALTH CHANGES IN THE ELDERLY USING IN-HOME MULTI-SENSOR DATA STREAMS

presented by Wenlong Wu

candidate for the degree of Doctor of Philosophy

and hereby certify that, in their opinion, it is worthy of acceptance.

James M. Keller

Mihail Popescu

Marjorie Skubic

Marilyn Rantz

ACKNOWLEDGMENTS

I would like to express my gratitude towards my advisor, Dr. James Keller, for his continuous support of my Ph.D. study and research. His guidance has always helped me throughout the time of researching and writing this dissertation. I cannot imagine having a better advisor and mentor for my Ph.D. study.

I would like to thank my committee members Dr. Mihail Popescu, Dr. Marjorie Skubic, and Dr. Marilyn Rantz for their help during my graduate career. Our conversations have been insightful and full of valuable advice. I would also like to thank my lab mates at the Center for Eldercare and Rehabilitation Technology who shared their thoughts and ideas along the way during my Ph.D. research. Thanks also go out to my girlfriend Sophie Wang for her love and support throughout my graduate school career.

A special thanks to my parents Guotang Wu and Qine Duan, who went beyond their way to raise me and support my overseas study.

TABLE OF CONTENTS

ACKNOWLEDGMENTS ii
LIST OF FIGURES v
LIST OF TABLES ix
ABSTRACT x
CHAPTER 1. INTRODUCTION 1
CHAPTER 2. SYSTEM ARCHITECTURE 7
2.1. Monitoring sensor networks 7
2.2. Feature extraction 9
CHAPTER 3. LITERATURE REVIEW 10
3.1. Health monitoring system for eldercare 10
3.2. Clustering 12
3.2.1. Fuzzy C-Means 13
3.2.2. Possibilistic C-Means 14
3.2.3. Sequential Possibilistic One-Means 15
3.3. Streaming clustering 17
CHAPTER 4. LEARNING DATA DISTRIBUTIONS FROM STREAMING DATA 20
4.1. Streaming Gaussian mixture approach 23
4.1.1. Gaussian Mixture Model 23
4.1.2. Sequential Possibilistic Gaussian Mixture Model 24
4.2. Streaming neural gas approach 25
4.2.1. Neural Gas 26
4.2.2. Possibilistic K-Nearest Neighbors 26
4.2.3. Streaming Soft Neural Gas 27
CHAPTER 5. TRAJECTORY ANALYSIS 32
5.1. Maximum typicality continuous decline measurement 32
5.2. Trend value measurement 33
5.3. Final SPGMM algorithm with trajectory analysis 35
CHAPTER 6. ALGORITHM VALIDATION 36
6.1. SPGMM validation 36
6.1.1. Validation on the synthetic dataset 36

6.1.2. Validation on the LG weather dataset 38
6.2. StreamSoNG validation 41
6.2.1. Parameter setting 42
6.2.2. Synthetic datasets 42
6.2.3. Comparison of different neuron update mechanisms 43
6.2.4. The effect of permuting the streaming data 46
6.2.5. Visualization of typicality value changes in streaming data 47
6.2.6. Validation of a real-world texture image dataset 48
CHAPTER 7. APPLICATION ON ELDERCARE 52
7.1. Case study I – Resident I 53
7.2. Case study II – Resident II 57
7.3. Case study III – Resident III 61
7.4. Case study IV – Resident IV 65
7.5. Discussion 68
7.6. Automatic ground truth via CLAMP 72
7.6.1. Introduction to Clinical Language Annotation, Modeling, and Processing (CLAMP) 73
7.6.2. Results 76
CHAPTER 8. ALGORITHM EXPLAINABILITY 80
8.1. Linguistic summarization for warnings 80
8.2. Annotation distribution of clusters 83
8.3. SHAP 84
8.4. Functional health score 90
CHAPTER 9. CONCLUSIONS AND FUTURE WORK 93
9.1. Conclusions 93
9.2. Future directions 94
9.3. Publications 95
REFERENCES 96
VITA 103

LIST OF FIGURES

Figure 1. (a) Trajectory of typical functional decline and the goal with in-home sensors and early detection of health changes [10]; (b) Principal Component Analysis (PCA) reduction of sensor data collected on a TigerPlace resident with 20 features extracted from the motion, bed, and depth sensor data. Each point represents one day. 2

Figure 2. The architecture of health alert system in TigerPlace 7

Figure 3. Typical apartment floor map with sensor deployment where "M" is motion sensor, "B" is bed sensor, and "D" is depth sensor 8

Figure 4. (a) Distance histogram in one class (b) η circle plot on the initialization set of synthetic dataset 29

Figure 5. The trend value illustration.....34

Figure 6. (a) The synthetic dataset scatter plot and (b) the result of SPGMM where the red x denotes "outlier," the blue diamond denotes "weak warning," the cyan diamond denotes "medium warning," the red diamond denotes "strong warning," and the blue x denotes recovered normal point. 37

Figure 7. The LG weather dataset scatter plot (a) of the original and (b) the labeled (Blue represents sunny days; green represents cold windy days; red represents snowy days) ...38

Figure 8. SPGMM results on the LG weather dataset: (a) warning system result where the blue diamond denotes "weak warning," the cyan diamond denotes "medium warning," the red diamond denotes "strong warning," and the blue x denotes recovered normal point; (b) the final clustering result.....39

Figure 9. The strong warning point analysis (zoomed in) 40

Figure 10. The maximum typicality value plot and trend value plot of the LG weather dataset 41
Figure 11. The scatter plot of three synthetic datasets (initialization and streaming sets) 43
Figure 12. Precision score on the three datasets with different neuron updating mechanisms (x-axis is the k value by updating k nearest prototype) 44
Figure 13. Precision scores of high typicality samples on the three datasets with different values of k in the PKNN using updating k nearest prototypes mechanism (x-axis is the k value by updating k nearest prototype).....45
Figure 14. Precision score on the three shuffled datasets with different values of k in the PKNN using updating k nearest prototypes mechanism (x-axis is the k value by updating k nearest prototype).....46
Figure 15. Four data samples' typicality value changes plots with time on the three synthetic datasets 47
Figure 16. Three examples of each image type: (a) pebbles, (b) bricks, and (c) plaid.....48
Figure 17. Typicality value plot for (a) a streaming sample from the pebbles class, (b) a streaming sample from the bricks class, (c) a streaming sample from the new plaid class 49
Figure 18. A sequence of transition images from the brick region (class 2) to the pebble region (class 1).....50
Figure 19. Typicalities of the sequence of transition images to the pebble class (class 1) and the brick class (class 2) 50
Figure 20. Retrospective data visualization (projections from the original 20-dimensional data) for Resident I labeled by year using (a) PCA, (b) t-SNE.....54

Figure 21. SPGMM clustering results of Resident I using retrospective data visualization (projections from the original 20-dimensional data): (a) PCA and (b) t-SNE.....55
Figure 22. Warnings generated by SPGMM on Resident I. Black represents negative electronic health records (EHR); blue represents a weak warning; magenta, a medium warning; and red, strong warning 56
Figure 23. Retrospective data visualization (projections from the original 20-dimensional data) for Resident II manually labeled by year using (a) PCA, (b) t-SNE. 58
Figure 24. SPGMM clustering results of Resident II using retrospective data visualization (projections from the original 20-dimensional data): (a) PCA, (b) t-SNE. 59
Figure 25. Warnings generated by SPGMM on Resident II. (black: negative EHR notes; blue: weak warning; magenta: medium warning; red: strong warning) 59
Figure 26. Retrospective data visualization (projections from the original 19-dimensional data) for Resident III manually labeled by year using (a) PCA, (b) t-SNE. 62
Figure 27. SPGMM clustering results of Resident III using retrospective data visualization (projections from the original 19-dimensional data): (a) PCA, (b) t-SNE. ..63
Figure 28. SPGMM warnings on Resident III. Black represents negative EHR notes; 63
Figure 29. Retrospective data visualization (projections from the original 15-dimensional data) for Resident IV manually labeled by year using (a) PCA and (b) t-SNE. 66
Figure 30. SPGMM clustering results of Resident IV using retrospective data visualization ((projections from the original 15-dimensional data)): (a) PCA, (b) t-SNE. 67

Figure 31. Warnings generated by SPGMM on Resident IV (black shows negative EHR notes; blue shows weak warning; magenta represents a medium warning, and red is a strong warning). 67
Figure 32. F1 score for each days' validated warnings generated by current 1-D alerts (blue), SPGMM (red), DenStream (green), and iForest (cyan) 71
Figure 33. The user interface of CLAMP 74
Figure 34. The interface in CLAMP for annotating entities and relationships.....76
Figure 35. Comparison between two clinicians' ratings and CLAMP labeling 77
Figure 36. The warnings and the negative EHR notes for resident III with three medium warning dates are marked (1: 11/5/2016; 2: 9/30/2017; 3: 12/25/2017) 81
Figure 37. Annotation type statistics of each cluster for Resident III 83
Figure 38. The distribution of annotation types in each cluster for Resident III 84
Figure 39. SHAP explanation for cluster 1 of Resident III.....86
Figure 40. Feature distribution of (a)-(c) important features, and (e)-(f) unimportant features, in cluster 1 vs. non-cluster 1, where orange histogram is cluster 1, blue histogram is non-cluster 1, blue vertical line is the mean feature value of cluster 1, and red vertical line is the mean feature value of non-cluster 1 87
Figure 41. SHAP explanation for cluster 2 of Resident III.....88
Figure 42. SHAP explanation for cluster 3 of Resident III.....88
Figure 43. SHAP explanation for cluster 4 of Resident III.....89
Figure 44. SHAP explanation of which cluster the day, 03/26/2015, belongs to 90
Figure 45. Functional health trajectory with SPGMM clustering result.....91
Figure 46. Functional health score with warnings generated by SPGMM 92

LIST OF TABLES

Table I. Sensor features used by our method 9

Table II. SP1M pseudocode 16

Table III. STREAMING SOFT NEURAL GAS PSEUDOCODE 31

Table IV. SPGMM pseudo code 35

Table V. Precision score of kNN, DenStream, and StreamSoNG 45

Table VI. Case study residents overview 53

Table VII. Feature list for Resident I 54

Table VIII. Warning dates and related EHR notes of Resident I 56

Table IX. Feature list for Resident II 58

Table X. Warning dates and related EHR notes of Resident II 60

Table XI. Feature list for Resident III 61

Table XII. Warning dates and related EHR notes of Resident III 63

Table XIII. Feature list for Resident IV 66

Table XIV. Warning dates and related EHR notes of Resident IV 67

Table XV. Evaluation of SPGMM warnings 69

Table XVI. F1 scores of different algorithms 70

Table XVII. Cohen's kappa score of three raters (two clinicians and CLAMP) on Resident II, III, IV 78

Table XVIII. The interpretation of the Cohen's kappa coefficient 79

ABSTRACT

The rapid aging of the population worldwide requires increased attention from health care providers and the entire society. For the elderly to live independently, many health issues related to old age, such as frailty and risk of falling, need increased attention and monitoring. When monitoring daily routines for older adults, it is desirable to detect the early signs of health changes before serious health events, such as hospitalizations, happen, so that timely and adequate preventive care may be provided. By deploying multi-sensor systems in homes of the elderly, we can track trajectories of daily behaviors in a feature space defined using the sensor data. In this work, we investigate a methodology for learning data distribution from streaming data and tracking the evolution of the behavior trajectories over long periods (years) using high dimensional streaming clustering and provide very early indicators of changes in health. If we assume that habitual behaviors correspond to clusters in feature space and diseases produce a change in behavior, albeit not highly specific, tracking trajectory deviations can provide hints of early illness. Retrospectively, we visualize the streaming clustering results and track how the behavior clusters evolve in feature space with the help of two dimension-reduction algorithms, Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE). Moreover, our tracking algorithm in the original high dimensional feature space generates early health warning alerts if a negative trend is detected in the behavior trajectory. We validated our algorithm on synthetic data, real-world data and tested it on a pilot dataset of four TigerPlace residents monitored with a collection of motion, bed, and depth sensors over ten years. We used the TigerPlace electronic health records (EHR) to understand the residents' behavior patterns and to evaluate and explain the health warnings generated by

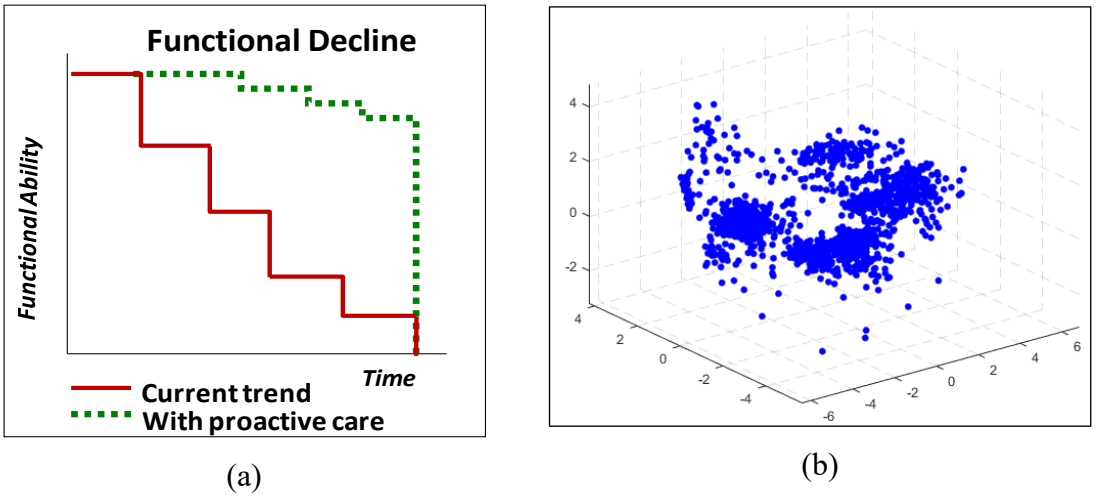
our algorithm. The results obtained on the TigerPlace dataset show that most of the warnings produced by our algorithm can be linked to health events documented in the EHR, providing strong support for a prospective deployment of the approach.

CHAPTER 1. INTRODUCTION

The population of older adults (age 65 and older) in the United States is expected to grow significantly over the next 30 years due to improved healthcare [1]. This population segment is expected to continue to increase as the baby boomer generation, born after the post-world War II years, reach retirement age. The increase in the older adult population is not just a U.S. trend, but also extends worldwide. Older adults are living longer than before, and they want to live as independently as possible in their homes. However, independent lifestyles can be complicated for those who have chronic illness and impairments in mobility or cognition. Late health assessment is an aggravating risk factor that often occurs due to a fear of institutionalization and the failure of timely physician assessments [2]. A possible solution to unrecognized health problems is to utilize an automatic health monitoring system that can detect and report signs of early illness [3]. Applying sensor network technology to the healthcare system presents a promising opportunity to address these problems.

"Aging in place" is a term used to describe a person living in the residence of their choice, for as long as they are able, as they age [4]. Technology that can help older adults "age in place" has been explored in recent years. A growing number of projects have deployed embedded sensor networks to monitor the health of older adults [5][6]. TigerPlace is a senior housing complex that enables residents to age in place and is a successful example of sensor networks embedded in the home [7][8][9]. TigerPlace offers various kinds of services as needed, promoting independence and helping residents maintain an active lifestyle by providing ongoing assessments for early illness recognition. An essential component in the health-monitoring system, installed at TigerPlace, is the

ability to continuously and unobtrusively collect sensor data captured from residents. The system then processes the acquired sensor data, infers the activities and behaviors of the residents, and detects changes in their health status. Once changes have been noted, an "alert" is sent to the clinical staff at TigerPlace, to inform them that a change may have occurred in the residents' health. The staff are able to assess the residents further, and inform a physician as necessary.



(a) (b)

Figure 1. (a) Trajectory of typical functional decline and the goal with in-home sensors and early detection of health changes [10]; (b) Principal Component Analysis (PCA) reduction of sensor data collected on a TigerPlace resident with 20 features extracted from the motion, bed, and depth sensor data. Each point represents one day.

Figure 1 (a) conceptualizes the motivation of a health-monitoring system, which was first hypothesized in [10] and later validated in multiple studies [8][11]. The solid line shows a typical trajectory of decline in functional ability based on research and practice with older adults. This trajectory includes plateaus with little decay and abrupt step-downs, illustrating dramatic functional decline, often the result of a significant health event or health change. Our goal with sensor-based monitoring and health alerts is to reduce the decline through very early recognition of health problems. The desired trend (the dotted line) extends the length of the plateaus and reduces the depth of the steps. The key is to

identify the small health problems before they become big problems and offer timely interventions designed to change the trajectory in functional decline. The result is increased functional ability, quality of life, and independence. The aim is to make a person's life better, not necessarily longer.

After examining years of in-home sensor data, we find that the in-home sensor data for "normal" days tend to form clusters in the feature space, as shown in a 3-D projection in Figure 1 (b); abnormal days appear as outliers [9][12]. "Normal" is defined as a typical behavior pattern that a resident shows in his/her current (stable) state. Outliers in the sensor data may be caused by health problems, sensor failure, visitor activity, or extended absence. Therefore, outlier detection models that are based on clustering algorithms can be applied to detect the non-normal days in an older adult's daily routine. We have seen cases in which a new cluster forms due to a sudden change in the older adult's baseline health condition [13]. For example, in Figure 1, the five steps from Figure 1 (a) might correspond to the 5 clusters from Figure 1 (b). However, note that after a health incident, the older adult may return to his previous behavior pattern (and the previous cluster) or a new cluster may form, corresponding to a new baseline health condition. Therefore, temporal analysis is required to track different patterns in an individual's various life stages, including small gradual changes, large sudden changes, temporary health changes, and more significant changes resulting in a new baseline health condition.

Temporal analysis for health change detection has been actively studied in recent years and can be applied for monitoring older adults' health. In our previous study [9], we developed a one-dimensional (1-D) algorithm using mean and standard deviation over a short temporal window (14 days in our case) to calculate potential alerts from changes in

the data for each sensor reading. In that case, each sensor feature is independent of the others. This simple 1-D alert algorithm was used in TigerPlace to test the approach of providing early interventions based on early signs of health changes found through the in-home sensing system, i.e., testing the approach motivated by Figure 1 (a). TigerPlace residents with the early alert system had, on average, an extended stay at TigerPlace of 1.7 years longer than TigerPlace residents without the sensor-based alerts [8]. That is, the sensor-based alerts facilitated earlier treatments, thereby resulting in better functional ability so that residents could remain at TigerPlace longer. The test also illustrated shortcomings of the simplistic 1-D algorithm. For example, it used a small set of features and a fixed temporal window for all older adults, and the algorithm was not able to identify shifts in the baseline health condition, which resulted in a new cluster. The algorithm also produced a high number of false alarms. Other previous work has investigated multi-dimensional algorithms and different machine learning methods, such as comparing a fuzzy pattern tree with a support vector machine [9]; both resulted in fewer false alarms. In other related work [14], the SEP change point detection algorithm is proposed which supports multi-dimensional data, compares distributions of windows and is effective for detecting abrupt changes. However, none of this previous work is designed to detect very early, subtle health changes or to track changes across different health clusters. In reality, older adults are all different, often with multiple chronic health conditions that present in unique ways. Each home is also different, and the set of sensors may be different depending on the home layout and the unique needs of the individual older adult. Because of these wide variations, supervised learning of a generalized model is not practical. What is needed is an algorithm that can adapt to a variety of conditions and environments and can recognize

a change in baseline health condition, as well as very early subtle changes that correspond to changing health. We have a live system based on depth videos installed at TigerPlace that focuses on immediate detections of abrupt changes such as falls [15][16]. The work proposed here, cast as streaming clustering, addresses the shortcomings of this previous work and offers an adaptive, flexible approach to recognizing very early signs of health changes, often before the older adult even recognizes the change.

In this work, I designed two multi-dimensional (multi-D) streaming processing algorithms: (i) Sequential Possibilistic Gaussian Mixture Model (SPGMM) that extends the Gaussian Mixture Model (GMM) into a temporal framework to track the pattern changes in data streams [17] and (ii) Streaming Soft Neural Gas (StreamSoNG) that extends the Neural Gas (NG) and Possibilistic K-Nearest Neighbors (PKNN) into a temporal framework to track the pattern changes in data streams [18]. A major contribution of this work is that trajectory analysis on data streams is defined and conducted upon SPGMM to offer earlier, more sensitive health alerts that are customized to the individual's health trajectory, with fewer false alarms. A warning system has been incorporated into the trajectory analysis to produce warnings if data streams have a negative trend. That is, an early warning can be generated if the trajectory through the feature space is consistently headed toward the cluster boundary but has not yet become an actual outlier. In this work, SPGMM and StreamSoNG are shown to have an excellent performance on synthetic and real-world datasets. The SPGMM algorithm with trajectory analysis is tested on the in-home multi-sensor data from TigerPlace. It is our first attempt to incorporate motion, bed, and depth sensors together for tracking health trajectories. Four case studies of TigerPlace residents are conducted to show how clusters (behaviors) in feature space evolve in their

various life stages, and how the warnings generated by the SPGMM algorithm correlate to health events extracted from the electronic health record (EHR) notes. In addition to accurate warning predictions, four algorithm explainability approaches are studied in this work to provide transparency to older residents and clinicians.

The rest of the report is organized as follows. Chapter 2 introduces our TigerPlace monitoring sensor network and features. Chapter 3 briefly reviews related work. Chapter 4 introduces two approaches to learn data distribution from streaming data: SPGMM and StreamSoNG. Chapter 5 discusses trajectory analysis in data streams. Chapter 6 validates SPGMM and StreamSoNG on synthetic and real-world datasets. Chapter 7 shows case studies on four residents' sensor data in TigerPlace and introduces an automatic ground truth labeling method via the CLAMP toolkit. Chapter 8 presents four algorithm explainability approaches. Chapter 9 summarizes conclusions and planned future directions.

CHAPTER 2. SYSTEM ARCHITECTURE

2.1. Monitoring sensor networks

Technology has a tremendous impact on the elderly by offering them full productive and independent lives. In TigerPlace, sensor technology has been utilized to help elderly residents not only manage their illness but also stay as healthy and independent as possible. An integrated monitoring system was installed in TigerPlace apartments with the University of Missouri IRB approval. Most residents in TigerPlace have at least one chronic disease, such as heart disease, diabetes, osteoarthritis, or early-stage dementia.

After focus groups with TigerPlace residents early in our research [19], we decided to use only non-wearable sensors for monitoring, since they are unobtrusive, reliable, user friendly, able to detect a range of health changes, and require no action on the part of the user. Most importantly, we found out the non-wearable sensors are readily accepted by older adults and by the TigerPlace clinical personnel.

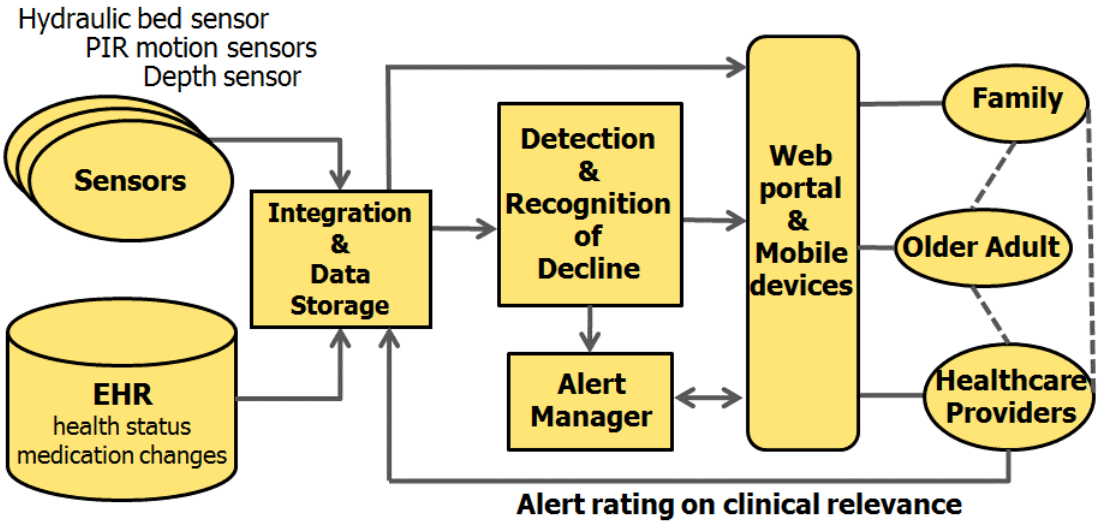


Figure 2. The architecture of health alert system in TigerPlace

The architecture of our health alert system is shown in Figure 2. The system includes a collection of in-home sensors, a data logger computer, a database for storing sensor data, a health alert manager, a web interface for viewing sensor data and alerts, and a database of clinical relevance ratings to inform algorithm development.

The sensor network employs a variety of sensors to capture various aspects of the residents' behavior. Figure 3 displays a floor map of a typical apartment that shows the placement of some of the sensors.

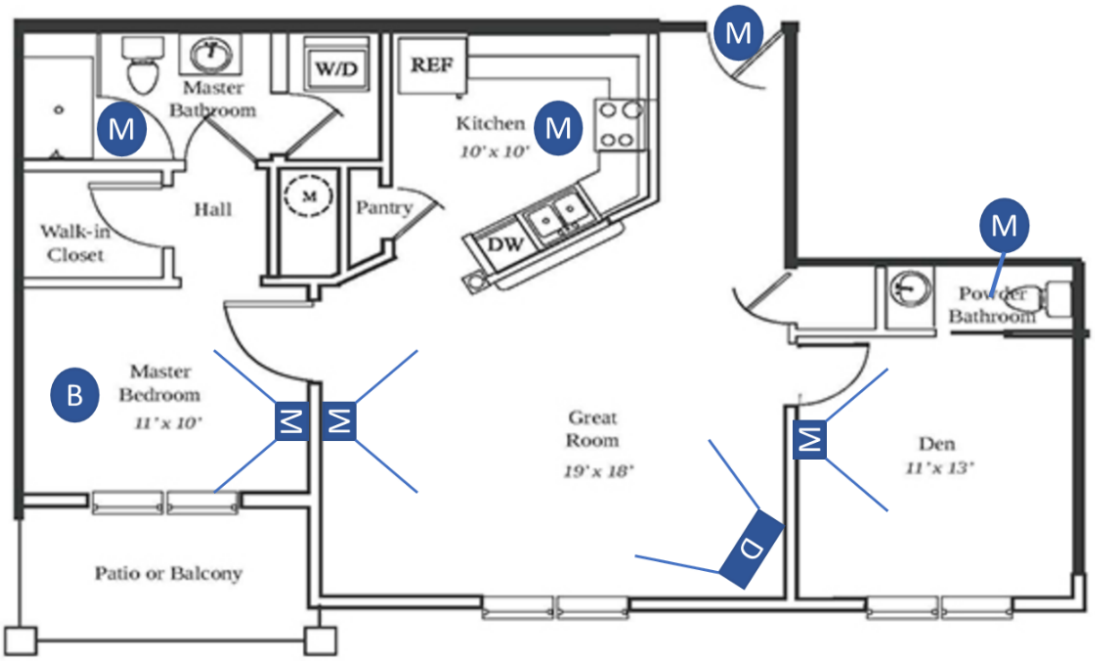


Figure 3. Typical apartment floor map with sensor deployment where "M" is motion sensor, "B" is bed sensor, and "D" is depth sensor

The system is unobtrusive, exploiting low-cost sensor technologies and specialized data processing. The network currently installed in TigerPlace consists of a set of passive infrared (PIR) motion sensors, hydraulic bed sensors, and depth cameras. Motion sensors are installed to detect presence in a particular room as well as for specific activities [20]. For example, a motion sensor mounted on the ceiling above the shower detects showering activity. The hydraulic bed sensor is a collection of four transducers (flexible tubes of

water) positioned under the mattress with electronics that capture quantitative pulse, respiration, and bed restlessness without direct contact with the individual [21][22]. The depth sensor is mounted in the living area to observe walking paths. It is not necessary to capture all possible walks; instead, we look for purposeful, relatively straight steps to establish a model of each resident's walking gait [23]. The technique tracks individual residents over time to capture changes in gait patterns and accommodates multi-resident homes and visitors.

2.2. Feature extraction

A list of the features computed from motion, bed, and depth sensors is given in Table I. The features generated for each resident are slightly different because of different sensor placement. The sensor placement is customized for each resident according to their apartment structure. Moreover, some sensors (bed, PIR, depth camera) have been improved over 10 years of monitoring and the data provided by the updated hardware was also incorporated in our study. However, the hardware change created extra challenges that we had to accommodate in our methodology.

Table I. Sensor features used by our method

Sensor Type	Measured Features
Motion	Aggregated firings of the bathroom, bedroom, living room, and kitchen.
Bed	Pulse, respiration, restlessness, and total time in bed.
Depth	Walking speed, stride time, stride length, stride density, and resident walking height

CHAPTER 3. LITERATURE REVIEW

3.1. Health monitoring system for eldercare

Rapid aging of population in the US and whole world requires increased attention from health care providers and from the entire society. The elderly population (aged over 65) is expected to grow significantly in the next 30 years [1]. In the same time, older adults prefer to live independently, many of the health issues associated with old age such as frailty and falling require increased attention. Many technologies regarding eldercare were developed to support older people's safety and independent living. The basic idea of many systems is to alert caregivers when emergencies happen. For example, some eldercare systems have a pull cord attached to the wall which generates an alert. However, the system will not work as expected if an older person is unable to initiate an alert when he/she is distant from the pull cord during an incident. To overcome this problem, monitoring sensors are introduced to monitor older people's activities at home to track behavior patterns and generate alerts automatically.

One of the early smart eldercare projects was conducted at the University of New South Wales, Australia. This project tested whether changes in functional health status among the elderly could be accurately determined remotely by continuously monitoring simple parameters such as mobility, sleep patterns and utilization of cooking, washing and toilet facilities [24]. In [25], the authors proposed the use of motion sensors to infer people's daily activities. They introduced a system that allowed routine, continuous, non-obtrusive monitoring of selected activities of daily living.

The Independent Life Style Assistant (ILSA) developed by Honeywell was another example of an early system that incorporated passive monitoring (mobility, sleeping and

toilet usage) and environmental preferences (room temperature) [26]. A field study was conducted in eleven elderly homes for six months, focusing on monitoring of mobility and medication compliance.

In other related work, Barnes et al. used motion and door sensors to extract a 24 hours activity profile and sent out alerts if newly logged data deviated significantly from the stored profile [27]. In [28], the authors described a well-being monitoring system for elderly residents with passive sensing from motion sensors. The motion sensor data are classified into one of six activities (sleeping, eating and so on) using fuzzy rules.

Besides the passive motion sensor, wearable health monitoring devices have also been investigated for homecare systems. In [29], the authors proposed a system that supported various wearable sensors for individual event detection and alerts. In [30], the authors introduced a model that used wrist-worn sensors to measure health and wellness status of an individual. However, the wearable sensors are not readily accepted by older people because they may not wear them consistently, for example, they may forget to wear them after charging. These disadvantages make wearable sensors impractical for daily monitoring.

Other types of sensors such as camera sensors, microphone sensors, laser scanning sensors and others have also been investigated by different research groups worldwide. These innovations have shown that the cost and acceptance of sensors are also crucial. Some people may feel uncomfortable living with camera sensors and microphone sensors. Laser scanning sensors are anonymous but are not affordable for many older people. So, the tradeoff between sensor efficiency and sensor cost needs to be considered during deployment in the real world.

In recent work, Cook et al. [31] evaluated the possibility of using unobtrusively collected activity-aware smart home behavioral data to detect one of the most common consequences of aging: functional health decline. The authors gathered the longitudinal smart home data of 29 older adults for an average of >2 years, and automatically labeled the data with corresponding activity classes and extracted time-series statistics containing 10 behavioral features. Using the collected sensor data and labels, they created regression models to predict absolute and standardized functional health scores, as well as classification models to detect reliable absolute change and positive and negative fluctuations in everyday functioning.

Many health monitoring systems have been piloted to track older people's daily routes. For example, MIT's PlaceLab, Georgia Tech's aware House, Honeywell's Independent Lifestyle Assistant, and University of Missouri's TigerPlace have demonstrated possible approaches for activity monitoring [19][32][33]. A key feature of these health-monitoring systems is the ability to continuously and unobtrusively collect information about daily activities and behaviors of older people and detect their health changes. With the rapid development of the Internet of Things (IoT) and 5G network recently, we believe the passive monitoring research will continue to rise in the next few decades.

3.2. Clustering

Clustering algorithms are powerful and useful techniques and have been popularly used in artificial intelligence. Cluster analysis research has a long history and has been combined with several research tools, including neural networks and probabilistic reasoning. Cluster analysis has a prominent role in data mining, pattern recognition and other search-related techniques, which helps researchers find the inner structure of data. It

has become a vital methodology in the discovery of data distribution and underlying patterns. A fundamental aspect of any clustering algorithm is its ability to construct meaningful partitions that organize objects from a dataset based on specific criteria. A clustering algorithm can also explore and discover as it uses its connectivity and density functions. It can define multiple-level granularity structures and hypothesize models for each cluster, then find the best fit of each model as it relates to another.

In this subsection, I will briefly review three clustering algorithms: Fuzzy C-Means, Possibilistic C-Means, and Sequential Possibilistic One-Means. Fuzzy C-Means (FCM) [34] is an effective method to find cluster structures in data. FCM works well on data that has overlapping clusters but it is sensitive to outliers. Possibilistic C-Means (PCM) [35] was then introduced by abandoning the membership sum-to-one constraint in FCM and has been shown to be robust against outliers. The Sequential Possibilistic One-Means (SP1M) [36][37][38] is a sequential version of PCM that iteratively hunts for one possibilistic cluster at a time, which is an effective mechanism to handle coincident clusters in PCM.

3.2.1. Fuzzy C-Means

The Fuzzy C-Means (FCM) [34] is defined as the minimization of the following objective function:

$$J_{FCM}(U, V; X) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m} \left|\left| v_i - x_k \right|\right|^2 \quad (1)$$

with the constraints

$$\sum_{i=1}^{c} u_{ik} = 1 \quad (2)$$

for all $k = 1, \dots, n$. Here, the fuzzifier parameter $m \in (1, \infty)$. Optimization of the FCM model can be performed by randomly initializing V and then alternatively updating U and V (alternating optimization) using the necessary conditions for extrema of J_{FCM},

$$u_{ik} = \left(\sum_{j=1}^{c} \left(\frac{\left|\left|v_i - x_k\right|\right|^2}{\left|\left|\left|v_j - x_k\right|\right|\right|^2} \right)^{\frac{1}{m-1}} \right)^{-1} \tag{3}$$

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^m x_k}{\sum_{k=1}^{n} u_{ik}^m} \tag{4}$$

until a suitable termination criterion holds, for example when successive estimates of V change less than a threshold ε.

$$\max_{i=1,\dots,c} \left|\left|v_i - v_i'\right|\right| < \varepsilon \tag{5}$$

In this work, we will always consider the Euclidean distance $||\cdot||$, the fuzziness index $m = 2$, and the threshold $\varepsilon = 0.01$. FCM may work well on many general datasets but cannot perform well on outliers and noise due to the constraint given by equation (2).

3.2.2. Possibilistic C-Means

The Possibilistic C-Means (PCM) algorithm [35] is a generalization of the Fuzzy C-Means (FCM) algorithm, which abandons the membership sum-to-one constraint in FCM (equation (2)). The objective function of PCM is as follows,

$$J_{PCM}(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^m \left|\left|v_i - x_k\right|\right|^2 + \sum_{i=1}^{c} \eta_i \sum_{k=1}^{n} (1 - u_{ik})^m \tag{6}$$

The PCM task is to minimize (6) subject to

$$u_{ik} \in [0, 1] \text{ for all } i \text{ and } k$$

$$0 < \sum_{k=1}^{n} u_{ik} < n \text{ for all } i$$

$$\max_i u_{ik} > 0 \text{ for all } k$$

The typicality update in the PCM is:

$$u_{ik} = \frac{1}{1 + \left(\frac{\left|\left|v_i - x_k\right|\right|^2}{\eta_i} \right)^{\frac{1}{m-1}}} \tag{7}$$

The parameter η_i determines the distance at which the typicality degree equals to 0.5. As was suggested in [35], η_i can be determined as follows:

$$\eta_i = P \frac{\sum_{k=1}^{N} u_{ik}^m \left|\left|v_i - x_k\right|\right|^2}{\sum_{k=1}^{N} u_{ik}^m} \tag{8}$$

Usually, *P* is set as one. The cluster center update is still the same as the cluster center update in equation (4).

PCM performs well against outliers but it may yield coincident cluster centers. Some PCM research has been done to deal with the coincident cluster center problem such as AMPCM [39] and PFCM [40].

3.2.3. Sequential Possibilistic One-Means

In PCM, each cluster is independent of the others and, effectively, is found separately. Hence PCM can be computed for $c = 1$. The Sequential Possibilistic One-Means (SP1M) algorithm [36][37][38] was created to overcome the coincident cluster drawback by generating one cluster at a time using Possibilistic One-Mean (P1M) and stopping when all the "dense" regions are found.

Table II. SP1M pseudocode

01: Input X, c, ε	
02: Initialize U, V as empty, $i = 1$	
03: Do {	
04: ---- Repeat <**loop to find a suitable cluster**>	
05: ---- ---- Pick $v \in X$ with probabilities	(9)
06: ---- ---- Repeat <**loop to execute P1M**>	
07: ---- ---- ---- Compute $\eta(i)$ dynamically	(*)
08: ---- ---- ---- Compute $u(v, X)$	(7)
09: ---- ---- ---- Compute $v(u, X)$	(4)
10: ---- ---- Until termination	(5)
11: ---- Until termination	(10)
12: ---- Append u to U	
13: ---- Append v to V	
14: } While (i ++ < c && # (P1M) < K)	
15: Output U, V	

The pseudo-code for the latest version of SP1M with dynamic η is shown in Table II, where X is the dataset, c is the input for maximum cluster number, ε is the threshold, K is defined to be the number of points whose maximum typicality is smaller than 0.5. Note that the (*) details of dynamic η computation in Table II are discussed in [38].

In SP1M, the cluster centers are not initialized randomly. They are initialized from probabilities based on the typicalities of the previously found clusters. The initial cluster centers are picked from dataset X with probabilities

$$p(x_k) = \begin{cases} \dfrac{1}{n} & if\ i = 1 \\ 0 & if\ \max\limits_{j=1,\dots,i} u_{jk} > 0.5 \\ \dfrac{1 - \max\limits_{j=1,\dots,i} u_{jk}}{n - \sum_{s=1}^{n} \max\limits_{j=1,\dots,i} u_{js}} & otherwise \end{cases} \tag{9}$$

When P1M has found a new cluster center v, v may be very close to one of the cluster centers in V found so far. We only consider v a new cluster center if it has a distance of at least 2η from each cluster center in V, that is, if

$$\min_{w \in V} \big| ||v - w|| \big| \geq 2\eta \tag{10}$$

If this condition does not hold, then v is discarded and P1M runs again to find a non-coincident cluster center. SP1M, as shown in Table II has two termination criteria. It can stop if/when c non-coincident clusters have been found. Recall that c is an input parameter from the user. If there are more than c clusters, this criterion would cause SP1M to miss legitimate structures (why we recommend setting c large). But that is only half of the story, since too large a value of c can result in an infinite search for non-existent clusters. In [38], a termination criterion was inserted to avoid the algorithm being trapped in an endless loop when no more new clusters can be found. Consequently, less than c non-coincident clusters may have been found in this way. Note that $\#(\mathrm{P1M}) < K$ means on the k^{th} execution of P1M (the process of searching for the k^{th} cluster), if the times of abandoning coincident cluster are greater than a threshold K, the algorithm stops, where K is the number of points whose maximum typicality is smaller than 0.5. The points whose maximum typicality is larger than 0.5 are likely to be those points strongly identified in an already found cluster. The value of K decreases at each P1M run.

3.3. Streaming clustering

Streaming data, such as social network click information or daily sensor firing information, are generated every day. Conventional clustering and classification models use static (batch) data and hence, are not directly applicable to data streams. Therefore, alternate strategies are required to incrementally update models as new feature vectors become available. There has been some research on classification in data streams [41], such as adapting CluStream as an on-line classifier [42], Very Fast Decision Trees [43][44], rule-based classifier [45], and a nearest neighbor technique [46]. However, for the most

part, these approaches are collectively referred to as streaming clustering, and there are a number of ways to organize them into taxonomies [47][48].

The multitude of streaming clustering algorithms have several things in common. First, they do not retain the entire dataset. They maintain cluster "footprints" that summarize the clusters discovered and have a mechanism of incrementally updating those footprints as new vectors arrive. In [49][50][17][51][52], underlying probabilistic models are used and the footprint contains probability distribution parameters. Many of the density-based streaming models also contain footprint entries that allow calculation of basic summary statistics [53][54][55][56][57]. The footprints can additionally contain the structure of fuzzy rules [58][59]. These streaming clustering methodologies generally maintain only summary information and a means to incrementally update it.

Classical cluster analysis seeks to answer three basic questions: (1) does a data set contain clusters, and if so, how many are there?; (2) how do you find the clusters?; and (3) are those clusters meaningful (cluster validity)? For streaming data, there is no way to answer the first question because the samples must be processed sequentially. The papers above are all examples of the quest to answer the second question, i.e., to find clusters. The third question is also problematic in the streaming mode. It is not possible to analyze the goodness of a partition of the data because, in any realistic case, no partition is ever maintained. Incremental versions of some standard cluster validity indices, iCVIs [60][61][62][63][64], have been developed. Close inspection of iCVIs, which we call incremental stream monitoring functions (iSMFs), show that they can be used to monitor what is happening during the streaming process, but do not in fact provide the kind of validity information as their classical ancestors on complete data sets [60][61][62][63][64].

The other basic trait of these algorithms is that, in fact, they assign labels to each point as it appears. For the most part, the labels are crisp – a point is assigned completely to an existing or new structure. Density-based algorithms put the point into a micro cluster, and underlying probability models, like Gaussian mixtures, use likelihoods to make the assignment. Once the points are assigned to a structure, their labels cannot change, as only the footprints are preserved. In fact, crisp labels are desirable because each incoming point is used to update the appropriate footprint. There is no iteration over the actual data, though some algorithms can form larger clusters from multiple footprints. Hence, what we call streaming clustering is actually classification, and should be thought of in that way. Streaming algorithms are almost always validated based upon the concept of "purity," or label accuracy and tested on labeled data.

Most approaches to streaming data analysis, after an initialization phase, assign a crisp label to incoming data. This implies that each point is assigned to one and only one existing structure (call it a cluster or class as you wish). The newly labeled point is used to update the structure footprint. The exception to this rule is when an incoming sample is judged by the algorithm to be an outlier with respect to previously seen data. The point doesn't receive a current class label, nor is it used to update the footprint. Different approaches deal with outliers in different ways, but they are the keys to discovery of new structure in the stream.

CHAPTER 4. LEARNING DATA DISTRIBUTIONS FROM STREAMING DATA

The motivating streaming algorithm in this work is a variant of the Missouri University (MU) streaming clustering algorithm (MUSC). Our first proposed streaming algorithm, SPGMM [17], is a modified form of MUSC found in [49] and [50] in which the underlying footprints are modeled by the parameters of the components of a Gaussian Mixture. The novelty of this approach is in the initialization phase and discovery of new structures from the outlier list. In the SPGMM, initialization, and the search for new structures, are done using the sequential possibilistic one-means with dynamic eta (SP1M-DE) algorithm [38]. In that variant, crisp label vectors (including an "outlier" label) are assigned to each incoming point. The outliers are saved and examined frequently to detect new structures. The iCVIs [60][61][62][63][64] have been used to provide visual insights into how the stream structures are evolving.

There is no fundamental reason that crisp label vectors need to be assigned to incoming data. Some applications show the desirability of soft labels in an object detection system where different classifiers are used in different scenarios. In one such scenario, an Unmanned Underwater Vehicle (UUV) traveling over different environments may employ different classifiers for detecting objects based on the observed seafloor type, like sand ripple, seagrass, etc. A baseline application of this type is described in [65], but without a streaming environment module. The environment-detecting component can be initialized with suspected known backgrounds. As new environment imagery is acquired, it must be matched to the known backgrounds to determine the appropriate classifier to use. However, the environment in a particular SAS image isn't always an exact match to the known

backgrounds; it can either be partially representative of more than one background, or something completely new. This is where possibilistic labels are particularly valuable.

To be precise, if $X = \{x_1, \cdots, x_n\}$ represents a set of objects, or a set of feature vectors for the objects, then a collection of subsets $\{A_1, \cdots, A_C\}$ along with a membership function μ is called a

1. Crisp partition if $\mu: X \to \{0,1\}$ and

$$\sum_{i=1}^{C} \mu_i\,(x_k) = 1\ for\ k = 1, \cdots, n.$$

2. Fuzzy or Probabilistic partition if $\mu: X \to [0,1]$ and

$$\sum_{i=1}^{C} \mu_i\,(x_k) = 1\ for\ k = 1, \cdots, n.$$

3. Possibilistic partition if $\mu: X \to [0,1]$ and

$$0 < \sum_{k=1}^{n} \mu_i\,(x_k) \leq n\ for\ i = 1, \cdots, C.$$

The vector $(\mu_1(x_k), \cdots, \mu_C(x_k))$ of values of the function μ is called the label vector for x_k. For a crisp label vector, x_k belongs to one and only one of the subsets; for a fuzzy label vector, the membership of x_k is shared among the subsets (though constrained to sum to 1); but for a possibilistic label vector, the membership of x_k is unconstrained and reflects the typicality of each of the subsets. Hence, in a possibilistic framework, a particular patch of seafloor can have a high typicality to more than one known seafloor type (when there is a blend) or it can have a very low typicality to all of the currently known seafloor types, signaling an outlier or new seafloor type.

If a streaming vector (new image) can be assigned typicalities in the known backgrounds, then classifiers can be blended. Images with low typicality in all known backgrounds are listed as outliers and some standard classifier fusion is used. When a new

structure (background class) is found in the data stream, either an automatically generated label can be assigned, or an active learning phase can be initiated in which a human will need to interact with the system to give the new class a label and perhaps a new classifier inserted.

We are mainly interested in the environment recognition portion of such a system. In addition to SPGMM, we propose a new streaming classification algorithm called Streaming Soft Neural Gas (StreamSoNG). The StreamSoNG algorithm uses the neural gas algorithm (NG) [66] during initialization to find sparse data representations for known classes, i.e., to generate the class footprints. A modified version of the possibilistic k-nearest neighbors algorithm (PKNN) [67] is employed as the streaming classifier to assign soft (possibilistic) labels to data stream vectors. The typicality vector can be used directly in a classification scheme. Based on maximum typicality, class footprints are incrementally updated. This updating allows for class definitions to vary from their initial topologies. An incoming sample that has low typicality values in all K classes is marked as an outlier and saved to an anomaly list. The sequential possibilistic one-means algorithm (SP1M) [38] is run the anomaly list to identify a potential new class.

In this chapter, I will introduce the two approaches to learn data distribution from streaming data: streaming Gaussian mixture approach (SPGMM) and streaming neural gas approach (StreamSoNG). Before processing with data streams, the model needs an initialization set to learn data distribution as a baseline so that it can update the data distribution with data streams. Gaussian mixture model (GMM) and Neural Gas (NG) are employed as the baseline model and are modified to update the model parameters with data streams.

4.1. Streaming Gaussian mixture approach

Streaming clustering is an effective tool to recognize normal baseline and to detect outliers in sequentially presented data. Perhaps more importantly would be the ability to predict that incoming data indicates movement towards a likely anomaly. In this subsection, a Gaussian Mixture Model (GMM) is employed to represent different patterns in the data stream. The Sequential Possibilistic One-Means (SP1M) is used for initialization, and is incorporated into the GMM framework to recognize new mixture components in the data stream. The new proposed algorithm is called Sequential Possibilistic Gaussian Mixture Model (SPGMM). I will review GMM first in section 4.1.1 and introduce SPGMM in section 4.1.2.

4.1.1. Gaussian Mixture Model

Given a *d*-dimensional vector *x*, a Gaussian mixture probability density function can be written as:

$$p(x) = \sum_{i=1}^{c} w_i p_i(x) \tag{11}$$

where *c* represents the number of mixture components, the mixture weights w_i satisfy $\sum_{i=1}^{c} w_i = 1$ and $w_i > 0$. Each component density $p_i(x), i = 1,2, \dots, c$ is the probability density function of Gaussian distribution parameterized by a $d \times 1$ mean vector u_i and a $d \times d$ covariance matrix Σ_i. The probability density function is shown as follows,

$$p_i(x) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp\left\{\frac{1}{2}(x - u_i)^T(\Sigma_i)^{-1}(x - u_i)\right\} \tag{12}$$

It has shown that any continuous probability density function can be approximated with high accuracy by using a sufficient number of Gaussian mixture components and by adjusting their means and covariance as well as the weights [68]. For video background

subtraction, it is suggested in [69] to use a fixed number of Gaussian mixture components, usually between 3 and 5, to model the data. However, the number of Gaussian components in different situations should change to describe different data patterns. Too many or too few Gaussian components may decrease the generalization or the accuracy of the algorithm. Thus, determining a suitable number of Gaussian components is of vital importance to the mixture modeling. Clustering algorithms are often used to help determine the number of Gaussian components in the data.

4.1.2. Sequential Possibilistic Gaussian Mixture Model

The heart of the basic Sequential Possibilistic Gaussian Mixture Model (SPGMM) follows the MUSC algorithm in [49][50], but instead of using PCM and AMPCM to initialize the GMM in [49][50], we use SP1M that can automatically determine the number of clusters, that is, the number of Gaussian mixture components. When a new streaming data point x_t comes in at time *t*, the Mahalanobis distance is calculated to each Gaussian mixture and then the algorithm determines if this new incoming point belongs to any of mixture model by comparing it with a pre-defined threshold.

If the new streaming data belongs to one of the Gaussian components, that is, one of the clusters is "firing" due to the activation of the new streaming data point, then the algorithm will update the Gaussian mixture model by updating the mean and covariance of the "firing" Gaussian by

$$u_{new} = u_{old} + \frac{x_t - u_{old}}{|u_{new}|} \tag{13}$$

$$\Sigma_{new} = \frac{(|u_{new}| - 1) * \Sigma_{old} + (x_t - u_{old})^T (x_t - u_{old})}{|u_{new}|} \tag{14}$$

where $|u_{new}|$ is the cardinality of the "firing" cluster, u and Σ are mean and covariance of that Gaussian.

Otherwise, the system will make an alert and the new streaming data point will be marked as an outlier. All the data points that are marked as outliers will be recorded in an anomaly list for further analysis. We note that the outlier points may not be all necessarily indicative of pattern changes. We also need to check the anomaly list to see if a new cluster forms due to the new pattern possibility. If enough data marked as outliers form a cluster, then we remove them from the anomaly list and build a new Gaussian component. SP1M is run in the anomaly list to hunt for one cluster at a time and return an empty set if it fails to recognize a cluster. The full pseudo code of the SPGMM is shown in chapter 5.3.

4.2. Streaming neural gas approach

In addition to SPGMM, we propose a new streaming classification algorithm called Streaming Soft Neural Gas (StreamSoNG). The StreamSoNG algorithm uses the neural gas algorithm (NG) [66] during initialization to find sparse data representations for known classes, i.e., to generate the class footprints. A modified version of the possibilistic k-nearest neighbors algorithm (PKNN) [67] is employed as the streaming classifier to assign soft (possibilistic) labels to data stream vectors. The typicality vector can be used directly in a classification scheme. Based on maximum typicality, class footprints are incrementally updated. An incoming sample that has low typicality values in all *K* classes is marked as an outlier and saved to an anomaly list. The sequential possibilistic one-means algorithm (SP1M) [38] is run the anomaly list to identify a potential new class. I will review NG in section 4.2.1, PKNN in section 4.2.2, and introduce StreamSoNG in section 4.2.3.

4.2.1. Neural Gas

The neural gas algorithm (NG) [66] is a competitive-learning neural network algorithm in the same family as the self-organizing feature map algorithm (SOFM) [70]. The NG algorithm aims to optimally describe the topology of data vectors using a fixed number of prototypes.

In the standard NG, given a set of vectors $X = \{x_t | t \in \mathbb{N}\}$ and a finite number of prototype vectors $w_i, i = 1, \dots, N$, a data vector x_t at timestamp *t* is randomly chosen from *X*. The distance order of the prototype vectors to the chosen data vector x_t is computed. Let i_0 denote the index of the closest prototype vector, i_1 denote the index of the second closet prototype vector, and i_{N-1} denote the index of the prototype vector most distant to x. Then each prototype vector is adapted according to

$$w_{i_k}^{t+1} = w_{i_k}^{t} + \varepsilon e^{-\frac{k}{\lambda}}\left(x_t - w_{i_k}^{t}\right), k = 0, 1, \dots, N-1 \tag{15}$$

where ε is the adaptation step size and λ is the neighborhood range. Both ε and λ are reduced with increasing iterations. After sufficiently many epochs using randomly sampled data vectors, the prototype vectors cover the data space with minimum representation error.

4.2.2. Possibilistic K-Nearest Neighbors

The possibilistic K-nearest neighbors algorithm (PKNN) extends the crisp KNN algorithm that first assigns a fuzzy membership (between 0 and 1) to each training pattern rather than using a binary class membership. The membership is assigned as described in [71] using

$$\mu^i(p) = \begin{cases} 0.51 + \left(\frac{n_i}{K}\right) \times 0.49, & if\ i = j \\ \left(\frac{n_i}{K}\right) \times 0.49, & if\ i \neq j \end{cases} \tag{16}$$

where n_i denotes the number of neighbors that belong to the i^{th} class, i.e., $\sum_{i=1}^{C} n_i = K$ and j is the actual class label of training sample p. Note, for StreamSoNG, the "training data" are the NG prototypes that make up class footprints. The pattern's fuzzy membership $\mu^i(p)$ controls its contribution during the classification process.

The version of the PKNN proposed in [67] assigns membership values (typicalities) $t^i(x)$ of a data vector x to class i using

$$t^i(x) = \sum_{k=1}^{K} \mu^i(p_k) w(x, p_k) \tag{17}$$

where p_k is the k^{th} nearest prototype to x and

$$w(x, p_k) = \frac{1}{1 + \left[\max\left(0, ||x - p_k|| - \frac{\eta_1}{\eta_2}\right)\right]^{\frac{2}{m-1}}} \tag{18}$$

In equation (18), η_1 and η_2 are constants that are estimated from the training data and $m > 1$ is a "fuzzifier" parameter. One approach is to identify the five nearest prototypes to each training sample and construct a histogram containing all associated distances. Then, we take $\eta_1 = \mu_H$ and $\eta_2 = 3 \times \sigma_H$, where μ_H and σ_H are the mean and standard deviation of the histogram of distances [67].

The first expression of a possibilistic KNN, shown in equation (18), worked well for a two-class application. In the streaming scenario, the number of classes is unknown. We will formulate a version more closely aligned to the original PCM that naturally extends to any number of classes.

4.2.3. Streaming Soft Neural Gas

In this work, we propose a new algorithm called the streaming soft neural gas algorithm (StreamSoNG) to classify streaming data vectors. The novel aspects of this

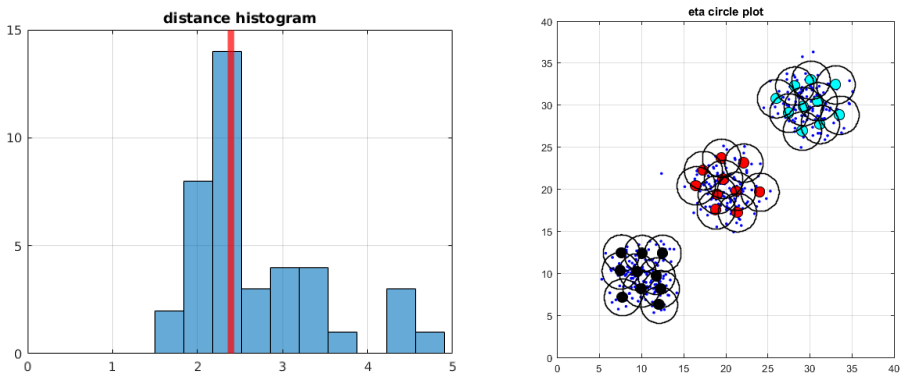
technique include using NG prototypes as class footprints, a different PKNN formulation both in the initialization phase and in the streaming possibilistic label assignment, and the incremental update of class footprints. During initialization, we use NG to learn representations (prototypes) of the initial C classes, $\{p_{ij}|i = 1, \dots, C; j = 1, \dots, n_i\}$, and only save the learned prototypes along with their labels as the class footprints.

We compute each prototype's distance to its K nearest prototypes (here, K is 5) in each existing class, then build a histogram of these distances for each class as shown in Figure 4 (a). The region in feature space that each prototype's region of influence (the parameter η_i) is estimated as the mean of the distance histogram using equation (19).

$$\eta_i = \frac{1}{n * K} \sum_{j=1}^{n} \sum_{k=1}^{K} \|p_{kj} - p_j\| \tag{19}$$

where n is the number of neurons in each class, K is the number of nearest neighbors, i is the cluster number index.

Figure 4 (b) shows the value of η plotted on the initialization set of the synthetic dataset. The η value in a class is the radius of the small circles in Figure 4 (b). As we see, each prototype covers a potentially overlapping sub-area of the data distribution in each class. The clusters in the example dataset in Figure 4 (b) were created by equal size circular Gaussians, so the values of η_i turn out to be the same for each class in this example. However, that need not be the case for more complex class definitions.



(a) (b)

Figure 4. (a) Distance histogram in one class (b) η circle plot on the initialization set of synthetic dataset

When streaming data x_t arrives, we first get its K nearest prototypes and compute the prototypes' fuzzy label memberships using equation (16) and the typicalities of streaming data to its K nearest prototypes using equation (20) below. The typicality value $t_{ik}(x_t)$ of the streaming data x_t to its k^{th} nearest prototype is computed as in the original PCM using

$$t_{ik}(x_t) = \frac{1}{1 + \left(\frac{\|x_t - p_{ik}\|^2}{\eta}\right)^{\frac{1}{m-1}}} \tag{20}$$

Here, i is the class label for the k^{th} nearest prototype, and η is estimated from the histogram of the distance between prototypes within a class, as shown in Figure 4. Normally, and for the experiments in this work, the Euclidean norm is used. Then, we multiply the fuzzy label memberships and typicalities to get the typicalities with fuzzy labels, $t'_{ik}(x_t)$, using Equation (21).

$$t'_{ik}(x_t) = \mu^i(p_{ik}) * t_{ik}(x_t) \tag{21}$$

In other words, we compute the scaled typicality $t'_{ik}(x_t)$ to each of the k prototypes using the typicality from equation (21) and the fuzzy labels $\mu^i(p_{ik})$ from equation (16) for

the prototypes of each class to get $t'_{ik}(x_t)$, and then compute the class average typicality using

$$\bar{t}_i = \frac{1}{K}\sum_{k=1}^{K} t'_{ik}(x_t) \tag{22}$$

and pass them to a scaling function, equation (23) as its class typicality.

$$T_i(x_t) = \begin{cases} 0, & x < 0 \\ 2 * x - x^2, & 0 \le x \le 1 \\ 1, & x > 1 \end{cases} \tag{23}$$

Now we have the class typicality vector, $T(x_t) = \left(T_1(x_t), \cdots, T_C(x_t)\right)^T$ of the streaming data, and use the maximum class typicality to represent the typicality of the streaming data to its closest class. If the maximum class typicality value is larger than a preset threshold, we assign the label of its closest class to this streaming data for footprint update. At that time, we update the prototypes that are in the same class of the current streaming data point according to

$$p_{ik}^{t+1} = p_{ik}^{t} + \alpha * T\left(t'_{ik}(x_t)\right) * \ e^{-\frac{k}{\lambda}}(x_t - p_{ik}^{t}) \tag{24}$$

where α is a learning rate (we use 0.1 in this work); p_{ik}^{t} is the k^{th} closet prototype (neuron) to data vector x_t at time t; λ is a neighborhood range parameter (we use 2 in this work). The typicality value $t'_{ik}(x_t)$ measures the typicality of a streaming data vector x_t to the prototype p_{ik}^{t}. If x_t has a high typicality to a given neighbor prototype, meaning that it is a good representation of that class, then we update the k^{th} nearest prototype with a large step; otherwise, we only update the kth nearest prototype by a small amount.

If the maximum class typicality value is smaller than a preset threshold, then the streaming data point has a low connection to any class. In this case, we mark the streaming data as an unseen class (outlier) and save it to the outlier list *O* for future analysis.

If the streaming data point is marked as an outlier, we run P1M on the outliers list *O* to search for a new class. If P1M finds a cluster for which the number of points with typicality bigger than 0.5 is larger than a minimum cluster-formed threshold *M*, we identify this new cluster as a new class, run NG on it, and remove the points from the outlier list *O*. The newly generated prototypes will be appended to the current learned prototypes and represent the new class. At this point, or actually at any time there is an outlier, the system can ask a human to provide a semantic class label or can reject an outlier completely.

The pseudocode of the streaming soft neural gas algorithm is shown in Table III.

Table III. STREAMING SOFT NEURAL GAS PSEUDOCODE

Initialization
Input: initialization set *X_init* and class label *y_init*;
Output: prototypes *P*;
01: for *i* in each class *y_init*:
02: ---- run NG on class(*i*) in *X_init*;
03: ---- save neurons of each class(*i*) into prototypes$p_{ik}, k = 1, \dots, n_i$;
04: end for

Stream Processing
Input: streaming set *X*, initial prototypes *P*, typicality threshold *t*, minimum number of points *M* to form a new class;
Output: streaming set class label vector *L* and class typicality vector *T*
01: initial PKNN model with *P* (declare PKNN);
02: for *x* in streaming set *X*:
03: ---- compute *K* nearest prototypes' fuzzy label memberships using eq. (16)
04: ---- compute typicalities $t_{ik}(x)$ of *x* to its *K* nearest prototypes in *P* using eq. (20)
05: ---- multiply typicalities with fuzzy label memberships to get the typicalities with fuzzy label $t'_{ik}(x)$ using eq. (21)
06: ---- compute the class typicalities of *x* by taking the average of typicalities with fuzzy label in each class using eq. (22) and apply scaling function using eq. (23)
07: ---- predict class label $L^i(x)$ and class typicality $T^i(x)$ using the largest class typicality
08: ---- if (class typicality $T^i(x) > t$):
09: ---- ---- update *P* for class *i* with *x* incrementally using eq. (24);
10: ---- else:
11: ---- ---- mark *x* as an outlier and save to outliers list *O*;
12: ---- ---- run P1M on *O* to search for a new cluster *C'*;
13: ---- ---- if (# of points with typicality>0.5 in *C'* > *M*):
14: ---- ---- ---- run NG on *C'*, and add the new prototypes to *P*;
15: ---- ---- ---- remove the points with typicality>0.5 in *C'* out of *O* and reset the outlier label in class label vector *L* with the new class;
16: ---- ---- end if
17: ---- end if
18: end for

CHAPTER 5. TRAJECTORY ANALYSIS

One characteristic of the data stream is that it forms a time series making neighboring data points related to each other. So, we can analyze the trajectory of the data stream as a time series to see if it has trends or an indication of forming an outlier (important to issue a preemptive alert). In this chapter, two methods of trajectory analysis in the data stream are introduced. The first method is to check the maximum typicality continuous decline in the data stream. The other method is to check if the data stream has the trend of going towards the cluster boundary.

5.1. Maximum typicality continuous decline measurement

In possibilistic clustering, the typicality value measures how "typical" a particular data point is to each cluster. If a data point has a large typicality value (maximum is 1), it means that data point strongly belongs to that cluster. A low typicality value indicates a weak cluster membership (minimum is 0).

The continuous decline of the maximum typicality values in a data stream suggests that the data stream gradually decreases its belonging to any cluster. This is a measure of trajectory analysis that can predict an early sign of pattern changes in the data stream, and can be used in an early warning system where an alert is triggered if consecutive declines of the maximum typicality value are detected. For example, if '*Few_Num*' consecutive declines of the maximum typicality are detected, a "weak warning" will be triggered; if '*Med_Num*' consecutive declines of the maximum typicality are detected, a "medium warning" will be issued; if '*High_Num*' consecutive declines of the maximum typicality are detected, a "strong warning" will be sent. For this work, the value of '*Few_Num*',

'*Med_Num*' and '*High_Num*' are set to 3, 5 and 7 accordingly. The decline threshold could be adjusted for different situations or applications.

5.2. Trend value measurement

In the data stream, the neighboring data points in time are correlated with each other. The data point in the current timestamp x_t could be influenced by $x_{t-1}, x_{t-2}, \ldots, x_{t-s}$ where *s* is a local window size. By looking at the past streaming data values, we may determine if the current data point has a trend of going towards the outside of the cluster. Two new vectors are defined below to compute the trend value. The first is computed from the current steaming data point and the past data stream in a window size *s*. This vector vec_1 is defined as

$$vec_1 = \frac{1}{s}\sum_{i=1}^{s}(x_{t-i+1} - x_{t-i}) \tag{25}$$

The second vector is computed from the current streaming data point and its closest cluster center. This vector is defined as

$$vec_2 = v_{close} - x_t \tag{26}$$

The trend value is decided by these two defined vectors and computed from the cosine function of these two vectors, as shown below,

$$cos\,\alpha = \frac{vec_1 \cdot vec_2}{|vec_1| * |vec_2|} \tag{27}$$

where $vec_1 \cdot vec_2$ is the dot product of vector 1 and vector 2. To illustrate the idea, an example is shown in Figure 5. The big red dot is the closest cluster center. The local window size $s = 3$ in this example. The blue dots are other data points inside of the cluster.

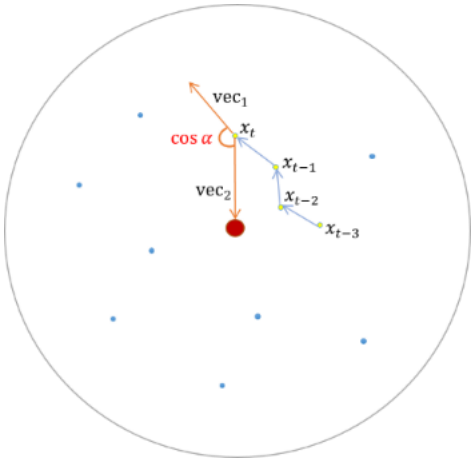


Figure 5. The trend value illustration

The range of the trend value is between -1 and 1. As we can see above, the trend value $\cos\alpha$ is smaller than 0 in this example, which indicates the current time stamp value x_t has a trend to go towards the boundary of the cluster. Thus, we can determine early signs of pattern changes by checking the symbol sign of the trend value. If the trend value $\cos\alpha$ is positive, x_t has the trend to go towards the center of the cluster. Otherwise, x_t has the trend to go towards the boundary of the cluster. The early signs of pattern changes usually happen when the trend value $\cos\alpha$ is negative, that is, when the current time stamp value x_t has a trend to go towards the boundary of the cluster.

5.3. Final SPGMM algorithm with trajectory analysis

Table IV. SPGMM pseudo code

Initialization:

Input:
X - Existing dataset; T_d - distance threshold;
M - Minimum number of data points in one Gaussian mixture

Run Sequential Possibilistic One-Means (SP1M) on X;
Find all possible Gaussian mixtures (means and covariance) and anomaly list

Stream processing:

01: For each data stream x_t:
02: ---- Calculate Mahalanobis distance to each Gaussian Mixtures and find the minimum
03: ---- If the minimal distance < T_d: **<inside of the Gaussian mixtures>**
04: ---- ---- Incrementally update the mean and covariance of the "firing" Gaussian mixture using equations (13) and (14)
05: ---- ---- Examine the anomaly list to check if any anomaly falls into the updated Gaussian mixtures and update them
06: ---- ---- If the symbol sign of the trend value is negative:
07: ---- ---- ---- Check the maximum typicality value continuous decline
08: ---- ---- ---- If '*High_Num*' continuous decline is detected:
09: ---- ---- ---- ---- Trigger "Strong warning"
10: ---- ---- ---- Elf '*Med_Num*' continuous decline is detected:
11: ---- ---- ---- ---- Trigger "Medium warning"
12: ---- ---- ---- Elf '*Few_Num*' continuous decline is detected:
13: ---- ---- ---- ---- Trigger "Weak warning"
14: ---- ---- ---- Else:
15: ---- ---- ---- ----"No warning"
16: ---- ---- ---- End if
17: ---- ---- End if
18: ---- Else: **<outside of the Gaussian mixtures>**
19: ---- ---- Alert "This may be an outlier in the data stream!"
20: ---- ---- Log x_t into anomaly list
21: ---- ---- Examine the anomaly list for an emergent new behavior pattern by running SP1M on anomaly list
22: ---- ---- If a new cluster is found && number of data points > M:
23: ---- ---- ---- Update the Gaussian mixtures and delete data points of the new cluster from anomaly list
24: ---- ---- End if
25: ---- End if
26: End for

CHAPTER 6. ALGORITHM VALIDATION

In order to ensure the proposed SPGMM and StreamSoNG algorithms work as expected, we created synthetic datasets and used real-world datasets to validate the algorithms. In section 6.1, SPGMM is validated on the synthetic Gaussian datasets and real-world weather datasets. In section 6.2, StreamSoNG is validated on the synthetic Gaussian datasets and real-world texture image datasets.

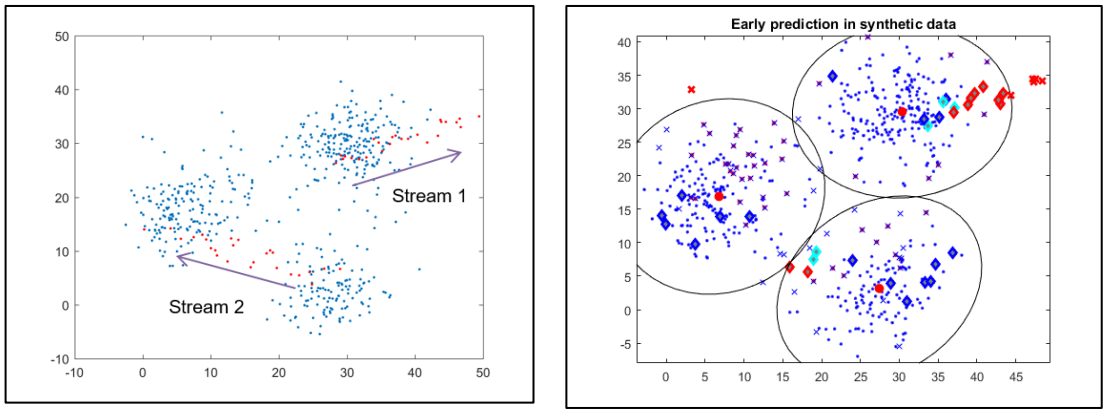
6.1. SPGMM validation

In this section, we used two datasets to test the SPGMM. The first dataset is a synthetic dataset that has three clusters, with vectors presented sequentially. Two trajectory paths were added to the data stream to test the algorithm. The second dataset is a real-world dataset, which is a collection of weather station nodes in the Le Genepi (LG) region in Switzerland [72]. We used two weeks of data at node 18 starting from October 10, 2007, to October 24, 2007. More details for each dataset are discussed in the sections below.

6.1.1. Validation on the synthetic dataset

In this experiment, we run the algorithm on a synthetic dataset that has three Gaussian clusters. In the Gaussian dataset, the mean values of three clusters are (30, 30), (28, 2), and (5, 16). The covariance matrix of three clusters are [18, 2; 2, 16], [13, 3; 3, 11] and [10, 0; 0, 16]. Each cluster has 200 data points. Two trajectory paths were added to the data stream. The scatter plot of the full synthetic data is shown in Figure 6 (a). Our synthetic dataset (including initialization dataset and streaming dataset) has three clusters. The initialization takes place when there is only one cluster present. That cluster is detected by the initialization procedure of SPGMM (the cluster is captured by the possibilistic

clustering algorithm in the initialization phase). The vectors from the remaining two clusters are streamed into the algorithm sequentially. Those clusters are automatically detected by SPGMM. To demonstrate our new algorithm for trajectory analysis, path 1 and path 2 in Figure 6 (a) show two trajectory paths added in the data stream that move out of an established cluster.



(a) (b)

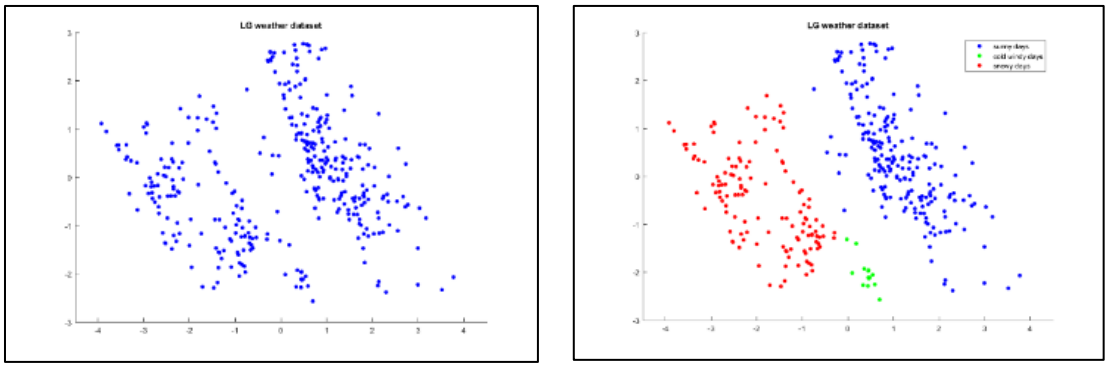
Figure 6. (a) The synthetic dataset scatter plot and (b) the result of SPGMM where the red x denotes "outlier," the blue diamond denotes "weak warning," the cyan diamond denotes "medium warning," the red diamond denotes "strong warning," and the blue x denotes recovered normal point.

The SPGMM is run on this synthetic dataset. Three clusters were captured and the cluster centers are marked as red dots as shown in Figure 6 (b). As we can see in the result, many warnings (marked with diamonds) were fired in both stream 1 and stream 2. For example, the data in stream 1 triggered the weak warning (blue diamond) at the beginning. As the data stream goes towards the outside of the cluster, the medium warning (cyan diamond) and the strong warning (red diamond) were triggered gradually. In the end, the data were marked as outliers (red x) because the data is "outside" of the cluster, which means this data point does not belong to any of the Gaussian components. For the trajectory in stream 2, the weak warning (blue diamond), the medium warning (cyan diamond) and the strong warning (red diamond) were triggered gradually. However, the data stream

becomes normal when it shifts to another cluster, which is a good indication since the data shifts to a different pattern instead of becoming an outlier. The SPGMM showed good performance in capturing the trajectory of moving towards the boundary of the cluster in this experiment.

6.1.2. Validation on the LG weather dataset

In August 2007, a small Sensor Scope network was deployed on the rock glacier located at 2500 meters on the top of Le Genepi above Martigny in Switzerland. A two-week period starting from October 10, 2007, to October 24, 2007, was used in this experiment. Each data point's time gap is one hour and there are a total of 356 records in this dataset. This dataset contains five features: ambient temperature, surface temperature, relative humidity, rain meter and wind speed. To better visualize the clusters and see how the trajectory goes, Principal Component Analysis (PCA) was used to reduce the dimension of the dataset from R^5 to R^2. A scatter plot of the reduced dimension LG weather dataset is shown in Figure 7 (a).



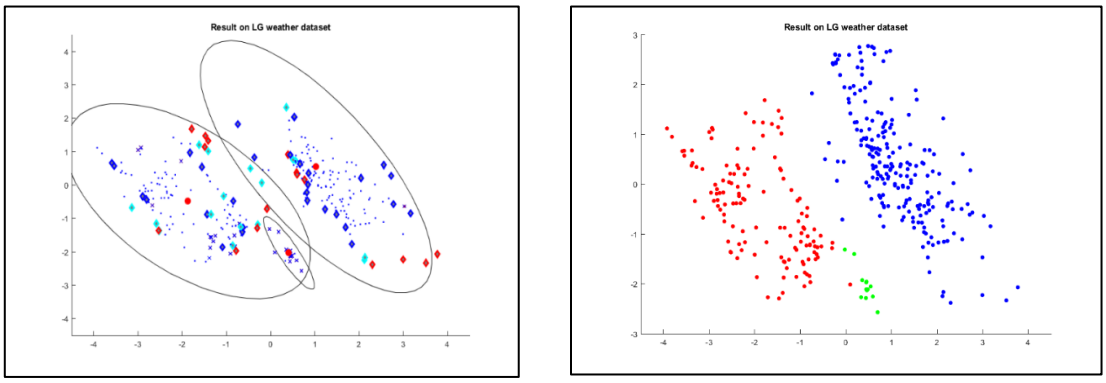
(a) (b)

Figure 7. The LG weather dataset scatter plot (a) of the original and (b) the labeled (Blue represents sunny days; green represents cold windy days; red represents snowy days)

By looking at the scatter plot, it does not provide clear visual evidence about how many clusters are in the LG dataset and where the clusters are located. Therefore, the

imagery information from the site is used to show that there is a snowy day during the two weeks. A windy and cold day precedes the snow. So, we consider the LG dataset to have three different types of weather: sunny days, cold windy days and snowy days as in the "ground truth" seen in Figure 7 (b). For the streaming order of the two-weeks data, the sunny days data (blue points) is streamed first, then the cold windy days data (green points), and finally the snowy days data (red points).

We run SPGMM on the LG weather dataset. One-fifth of the dataset is used to initialize the Gaussian mixture prototype. The other four-fifths of the dataset is a data stream that is fed into the algorithm. The final result of SPGMM on this dataset is given in Figure 8 where Figure 8 (a) shows the warning system to predict the early signs of the pattern changes and (b) shows the final clustering results of each cluster.



(a) (b)

Figure 8. SPGMM results on the LG weather dataset: (a) warning system result where the blue diamond denotes "weak warning," the cyan diamond denotes "medium warning," the red diamond denotes "strong warning," and the blue x denotes recovered normal point; (b) the final clustering result

As we can see above, three types of warnings occur inside of the cluster but are detected sequences to have trends that go towards the boundary of the cluster. There are many warnings that are not so close to the cluster boundary, but they have trends of shifting out of the cluster in the data stream sequence. In Figure 9, we analyze the trajectory path

where the sunny days cluster shifts to the cold windy days cluster. On the current timestamp x_t, a strong warning (the red diamond in Figure 9) is fired. Before x_t, weak warnings (the blue diamond in Figure 9) and medium warnings (the cyan diamond in Figure 9) are also fired. After x_t, the weather becomes cold and windy and shifts to the cold and windy days cluster, as created by SPGMM. Therefore, the strong warning at the current time stamp x_t predicted the early sign of weather changes successfully.

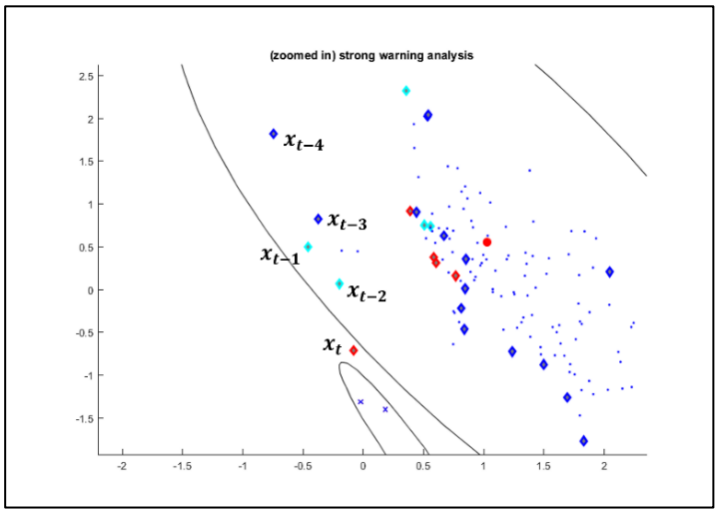


Figure 9. The strong warning point analysis (zoomed in)

The successful prediction of the early sign of weather changes is due to the trajectory analysis that is introduced in the last chapter: the maximum typicality continuous decline measurement and the trend value measurement. The maximum typicality plot and the trend value plot of the data stream for this dataset are shown in Figure 10. By looking at these two plots, we find that the maximum typicality value tends to decline continuously, and the symbol sign of the trend value trends to be negative when one cluster stream shifts to the outliers or to other clusters.

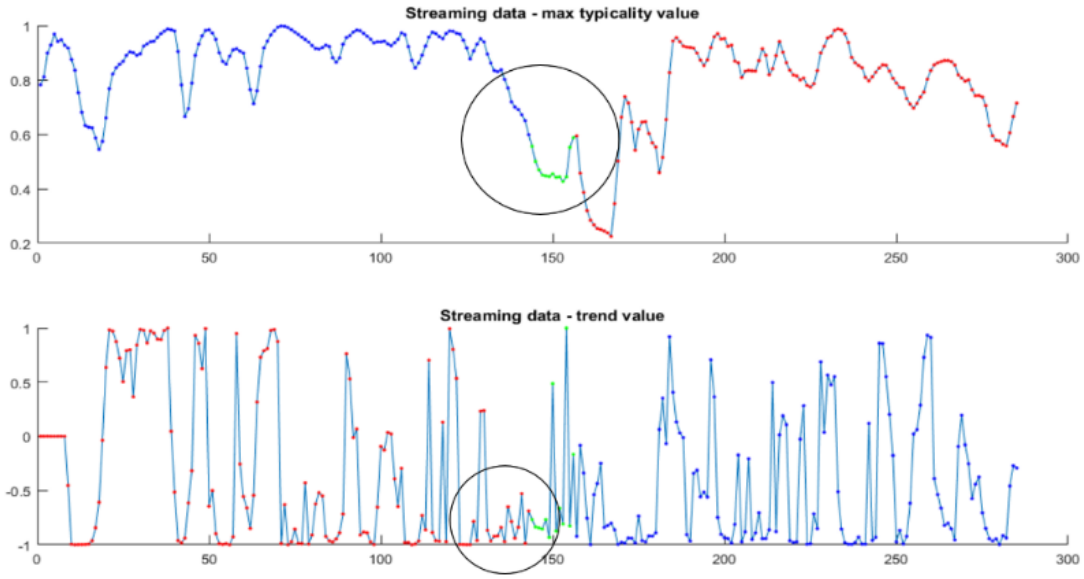


Figure 10. The maximum typicality value plot and trend value plot of the LG weather dataset

6.2. StreamSoNG validation

In this section, we run four experiments to test the StreamSoNG algorithm. The first experiment compares different neuron (prototype) update mechanisms on three synthetic datasets. The second experiment studies the effect of permuting the presentation order of streaming data on the algorithm. The third experiment visualizes how the typicality value of a specific data sample changes as the model updates with streaming data. The last experiment tests the StreamSoNG algorithm on a real-world texture image dataset. We use the precision score to evaluate the models. The precision score compares the prediction with the ground truth, and is defined as

$$precision(A, B) = \frac{\sum_{i=1}^{N}(A_i = B_i)}{N} \tag{28}$$

where A is streaming prediction set, B is streaming ground truth set, and N is the samples number of the streaming set. The precision score is computed on the whole streaming set. Note that this measure is computed after hardening the possibilistic labels assigned to each

streaming input vector. This makes our soft streaming classifier comparable to other crisp models.

6.2.1. Parameter setting

There are some user-specified choices in Table III that need to be made for the implementation of StreamSoNG. We used the number of neurons in each class $n = 10$, typicality threshold $t = 0.1$, minimum number of points to form a new class $M = 30$, epsilon $\varepsilon = 0.01$, fuzzifier $m = 1.5$, learning rate $\alpha = 0.1$ (in equation (24)), neighborhood range lambda $\lambda = 2$ (in equation (24)).

6.2.2. Synthetic datasets

To test the StreamSoNG algorithm, we first created four synthetic datasets. In the first dataset, the mean values of three clusters in initialization set are (10, 10), (20, 20), and (30, 30). The mean values of two clusters in streaming set are (40, 40) and (50, 50). The covariance matrix in both initialization and streaming sets is [4, 0; 0, 4]. In the second dataset, the mean values of three clusters in initialization set are (10, 10), (20, 20), and (30, 30). The mean values of two clusters in the streaming set are (40, 40) and (50, 50). The covariance matrix in both the initialization and the streaming set is [15, 0; 0, 15]. In the third dataset, the mean values of three clusters in the initialization set are (10, 20), (20, 30), and (30, 20). The mean values of two clusters in the streaming set are (20, 10) and (20, 20). The covariance matrix in both the initialization and the streaming set is [5, 0; 0, 5]. The scatter plot of the three datasets is shown in Figure 11. The top three scatter plots, (a) – (c), represent the initialization sets and the bottom three plots, (d) – (f), show the temporal sequence of the stream. The arrows in streaming sets show how the streaming data evolves over time.

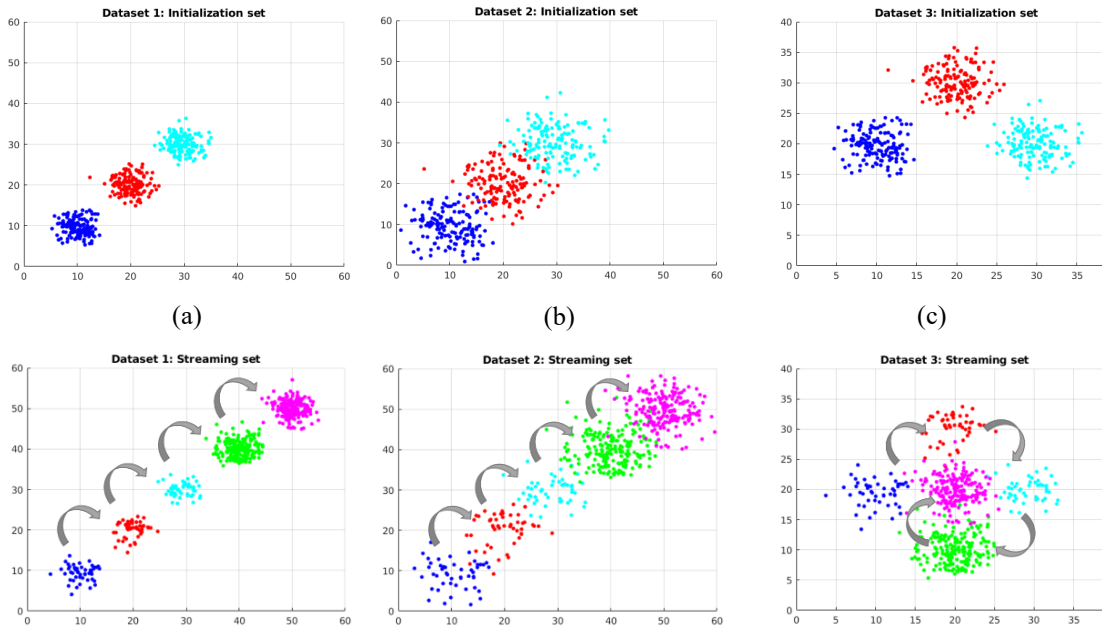
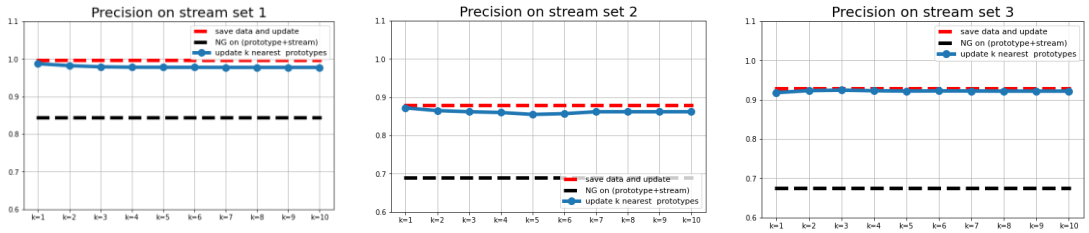


Figure 11. The scatter plot of three synthetic datasets (initialization and streaming sets)

6.2.3. Comparison of different neuron update mechanisms

Three neuron updating mechanisms can be used in data stream processing. The first one is to save all data samples and rerun the NG algorithm on the updated data samples to get new data representations (neurons). The second method is to update only the k nearest neurons using equation (24). The last method is to rerun the NG algorithm on the prototypes and the new streaming data sample, using it, in effect, as a potential prototype. Figure 12 shows the precision scores of the three neuron updating mechanisms on the three synthetic datasets. The red dotted line is the first update mechanism that saves all data samples and reruns the NG algorithm on the entire updated samples. The blue line is the second update mechanism that updates the k nearest neurons. The black dotted line is the third update mechanism that reruns the NG algorithm on the neurons and the new streaming data sample.

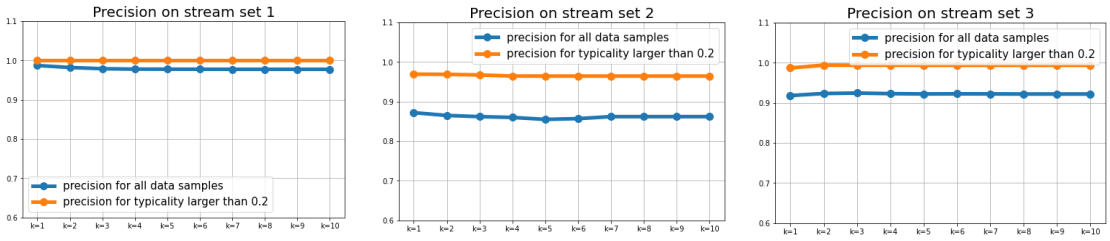


(a) (b) (c)

Figure 12. Precision score on the three datasets with different neuron updating mechanisms (x-axis is the *k* value by updating *k* nearest prototype)

As we see in Figure 12, the first method has the highest precision score but it requires more computation and data storage because it saves all data samples in both initialization and streaming sets. This method represents an upper bound but goes against the spirit of streaming data processing and does not scale up efficiently in a Big Data environment. The second method that updates the *k* nearest neurons performs well compared to the first approach and clearly outperforms the third method that reruns the NG algorithm on the neurons and new streaming data sample. The third neuron updating mechanism can easily forget the learned representations. Therefore, updating *k* nearest prototypes with streaming data is an accurate and efficient method to incrementally adjust the prototypes in a class. The selection of *k* value is robust as the precision scores of using different *k* values are close.

Furthermore, the StreamSoNG algorithm can not only produce a class label for a data sample, but also a typicality matrix that measures how well a data sample belongs to a specific class. If the typicality value of a streaming data sample is high, the algorithm is more confident about its prediction. In this experiment, we compute the precision scores for the predictions only where typicality values are higher than 0.2.



(a) (b) (c)

Figure 13. Precision scores of high typicality samples on the three datasets with different values of *k* in the PKNN using updating *k* nearest prototypes mechanism (x-axis is the *k* value by updating k nearest prototype)

As we see in Figure 13, the precision scores for typicality values higher than 0.2 are higher than the precision score for all streaming data samples using updating *k* nearest prototypes mechanism. That is, the StreamSoNG algorithm has a higher precision score for its confident predictions.

In this experiment, we also run the *K-Nearest Neighbor* (KNN) algorithm [73], *Adaptive Random Forest* (ARF) classifier [74], *Very Fast Decision Rules* (VFDR) classifier [75], and DenStream algorithm [53] on the same synthetic datasets. The results are listed in Table V.

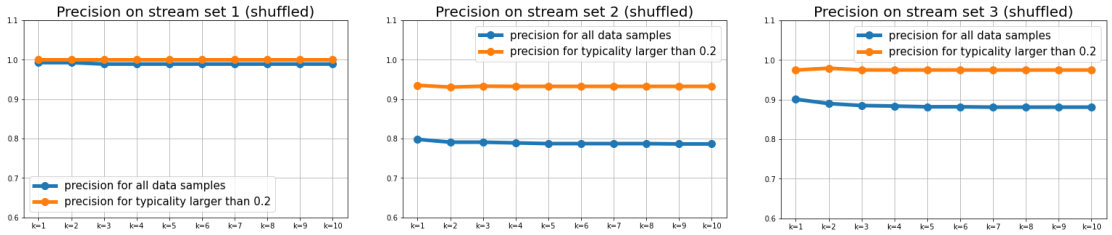
Table V. Precision score of kNN, DenStream, and StreamSoNG

	kNN (k=3)	ARF [74]	VFDR [75]	DenStream	StreamSoNG (k=3)
Dataset 1	0.271	**0.182**	**0.273**	**1**	0.979
Dataset 2	0.259	0.117	0.266	0.073	**0.862**
Dataset 3	0.271	0.095	0.273	0.627	**0.924**

As we can see in Table V, the KNN, ARF, and VFDR schemes have low overall precision scores because they cannot detect the new classes in data streams. DenStream has a perfect score on the well separated dataset (dataset 1) but has a poor score on the overlapping cluster dataset (dataset 2) because it merges close structures. The StreamSoNG algorithm has the highest overall precision score because it can not only detect new classes but also works well on the overlapping cluster dataset.

6.2.4. The effect of permuting the streaming data

In real-world data, the streaming data might not follow a specific pattern arriving at the model as we assumed in experiment 1. Here, we shuffle the order of streaming set and re-run the StreamSoNG algorithm on the shuffled streaming set. The precision scores using the *k* nearest neurons update mechanism on three synthetic datasets are shown in Figure 14 as a function of *k*.



(a) (b) (c)

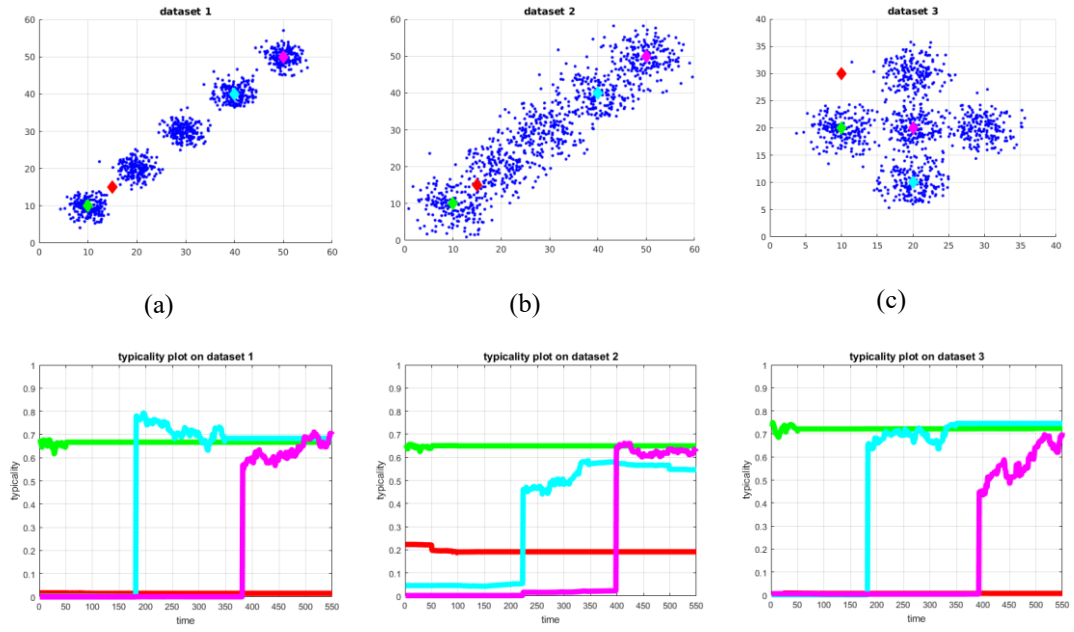
Figure 14. Precision score on the three shuffled datasets with different values of *k* in the PKNN using updating *k* nearest prototypes mechanism (x-axis is the *k* value by updating *k* nearest prototype)

As we see in Figure 14, the precision scores on dataset 1 stay very close to the precision scores on the unshuffled streaming set using updating k nearest prototypes mechanism because the classes in dataset 1 are well separated. The precision scores on datasets 2 and 3 decrease on the shuffled streaming sets compared to the precision scores on the unshuffled streaming set. This is because the two clusters in streaming set are very close to each other, and it is hard to distinguish them at the beginning with randomly presented vectors. In addition, we compute the precision scores for the streaming samples with typicality values higher than 0.2. As before, the precision scores with confident predictions are higher than the precision score for all streaming data samples.

6.2.5. Visualization of typicality value changes in streaming data

In this experiment, we track the typicalities of four data samples and see how they change as the model updates with streaming data. It is as if these points are presented repeatedly, after each real sample of the data stream. They are not used to update the class footprints, but only to monitor changes in maximal typicality throughout the process.

Figure 15 shows the final maximum typicality plot of the four points in the three datasets. In Figure 15 (a) – (c), 4 diamond symbols (in green, red, cyan, magenta color) in each dataset are studied. Their maximum typicality value plots with respect to time are shown in Figure 15 (d) – (f).



(a) (b) (c)

(d) (e) (f)

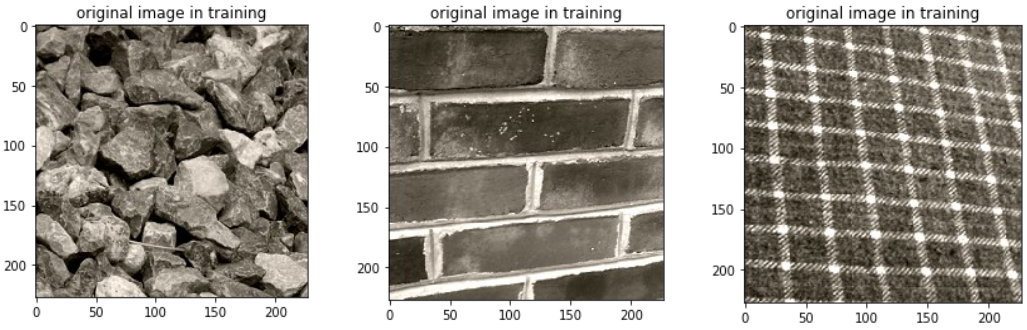
Figure 15. Four data samples' typicality value changes plots with time on the three synthetic datasets

As we see in Figure 15, the green diamond symbol always has a high typicality value as the model updates with streaming data because it is always in the middle of a class. The red diamond symbol has a low typicality value all the time because it is always in the sparse area. The cyan diamond symbol and the magenta diamond symbol are two interesting

cases. Their typicality values are low at the beginning, then become high as more streaming data form a new cluster around their regions and the algorithm creates a new class around them.

6.2.6. Validation of a real-world texture image dataset

In this experiment, we run the StreamSoNG algorithm on the UMD texture dataset [76]. We used 400 images as the initialization set, which included pebble- and brick-type images. We used another 600 images as a streaming set that included pebble-, brick- and plaid-type images. The plaid type image is new in the streaming set and is not included in the initialization set. Three examples of each type of image are shown in Figure 16. This is a surrogate test for the intended use of determining typicalities of seafloor textures displayed in SAS imagery.

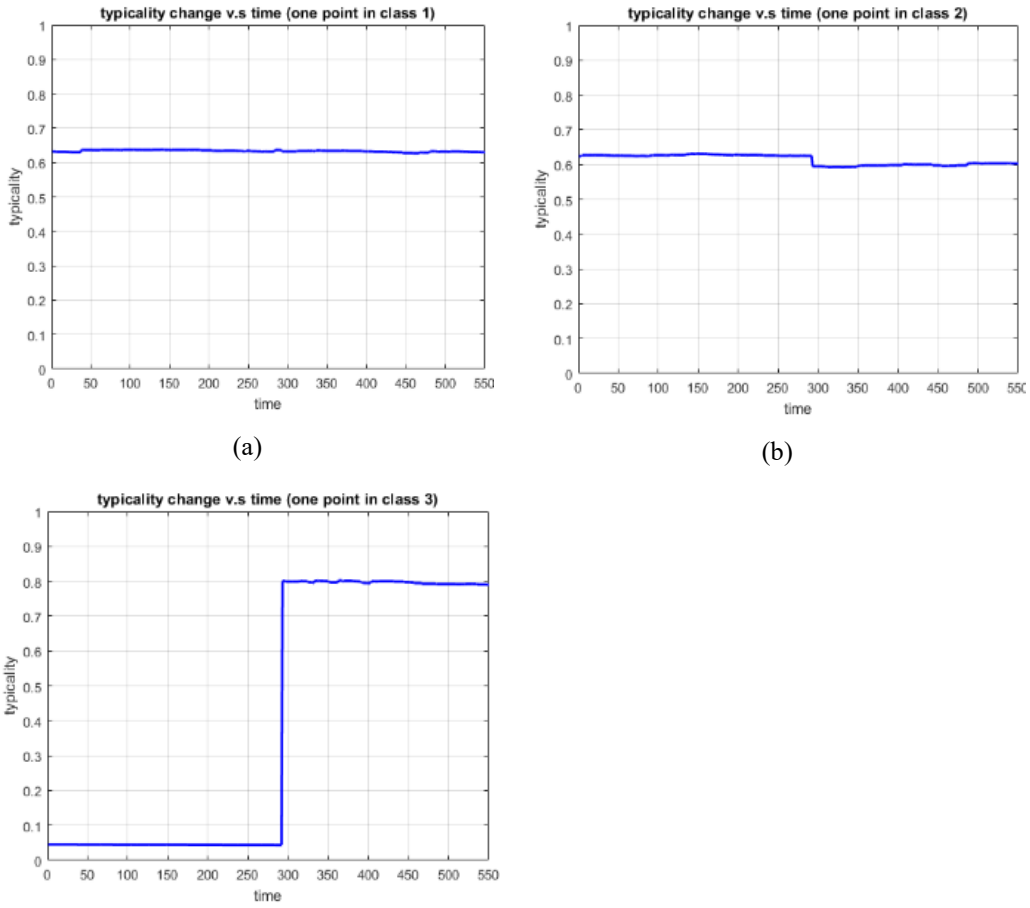


(a) (b) (c)

Figure 16. Three examples of each image type: (a) pebbles, (b) bricks, and (c) plaid

First, we used the Resnet18 pre-trained model [77] with classification layers removed to extract 512 features from these images. Then we train an autoencoder model with three hidden layers on the 512 features of the initialization set. The Resnet18 pre-trained model and encoder in the autoencoder are used together to process the streaming images to get 16 features. Then we run the StreamSoNG algorithm on the extracted 16 features. StreamSoNG achieves 81.3% precision on the entire streaming set and 95.7% precision on

the streaming samples that have maximum typicality value higher than 0.2. StreamSoNG detects the plaid class in the streaming set and produces a new class label for the plaid type of image. Figure 14 shows three examples of typicality changes in each class.



(a) (b)

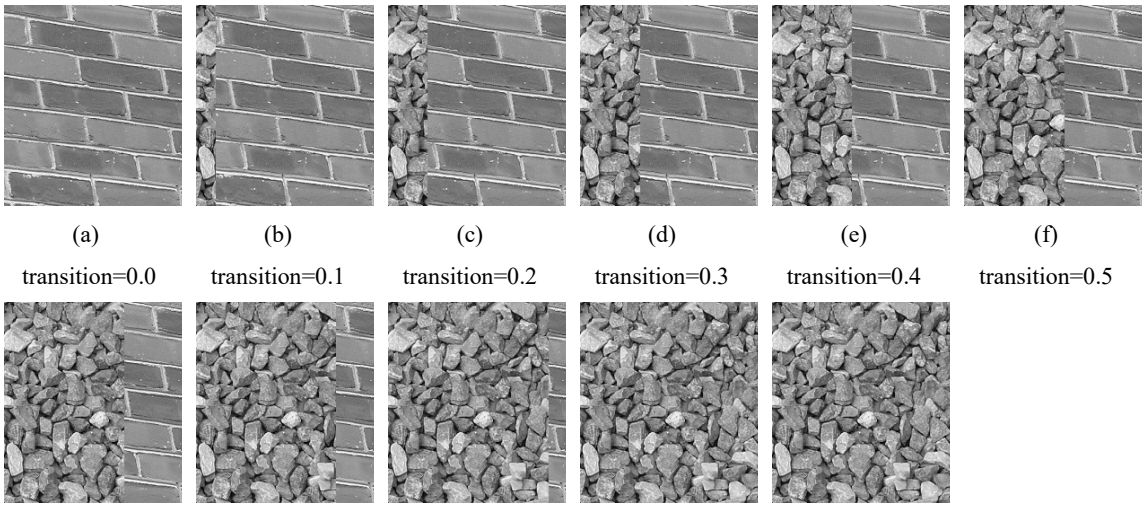
(c)

Figure 17. Typicality value plot for (a) a streaming sample from the pebbles class, (b) a streaming sample from the bricks class, (c) a streaming sample from the new plaid class

The first two samples in the pebbles and bricks class consistently have high typicality values as streaming data is fitted into the model. The typicality value of the third example from the new plaid class has low typicality value at the beginning and high typicality value when a new plaid class is created in the model.

One application of our StreamSoNG model is to detect the environment using drones. Figure 18 mimics a scenario that a drone flies from a brick region to a pebble region. In

Figure 18 (a), the drone is completely in the brick region. It gradually flies over to the pebble region as Figure 18 (b)-(j) show. In the end, the drone is completely in the pebble region as Figure 18 (k) shows.



(a) transition=0.0 (b) transition=0.1 (c) transition=0.2 (d) transition=0.3 (e) transition=0.4 (f) transition=0.5

(g) transition=0.6 (h) transition=0.7 (i) transition=0.8 (j) transition=0.9 (k) transition=1.0

Figure 18. A sequence of transition images from the brick region (class 2) to the pebble region (class 1)

We keep track of the typicalities of the sequence of transition images in Figure 18 while running the StreamSoNG model. The typicalities of the images to the pebble class (class 1) and the brick class (class 2) are shown in Figure 19.

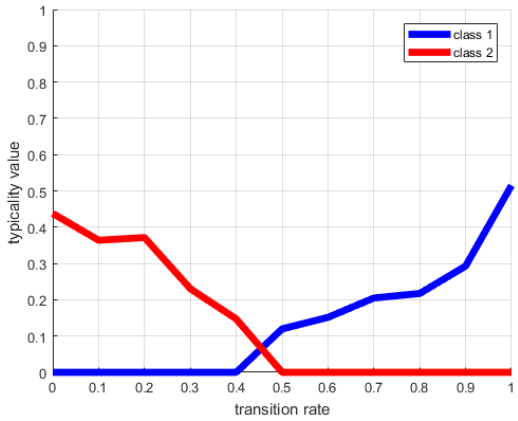


Figure 19. Typicalities of the sequence of transition images to the pebble class (class 1) and the brick class (class 2)

As the environment shifts from the brick region (class 2) to the pebble region (class 1), the typicality value to class 1 is increasing and the typicality value to class 2 is decreasing. Our StreamSoNG reflects the environment transition fact in the typicality plot successfully.

CHAPTER 7. APPLICATION ON ELDERCARE

With the ability to model different patterns and an incremental style of learning, the SPGMM algorithm and the StreamSoNG algorithm are attractive for temporal analysis. As such, we tested the SPGMM with trajectory analysis's effectiveness in locating gradual changes using four case studies of resident data collected by the TigerPlace sensor network system. To aid in annotating the trends, we made use of electronic health records (EHR) populated by the clinical staff.

Table VI shows an overview of the four residents in TigerPlace studied in this work. Resident I was the first resident of TigerPlace who agreed to use sensors in 2005 when the depth sensor was not yet deployed. Resident II was the longest living resident in TigerPlace, who had stayed for more than twelve years; here, we consider the time period during which depth data were collected. Residents III and IV are two recent residents in TigerPlace. Resident IV uses a wheelchair in daily routines; therefore, this resident does not have gait data from the depth sensor. The behavior patterns of each resident and each apartment structure are different, so the number of features extracted from each resident is varied. See subsequent subsections for more details of each resident. We use the first one-third of data of each resident as the initialization set and the remaining two-thirds of data as the stream set for incremental model update and evaluation.

Table VI. Case study residents overview

Case study	Data length	Total days	Features extracted
Resident I	10/10/2005 – 01/28/2007	476	30 (motion, bed)
Resident II	10/03/2011 – 04/07/2013, 06/01/2015 – 09/01/2017	2161	20 (motion, bed, depth)
Resident III	03/25/2015 – 06/01/2019	1530	19 (motion, bed, depth)
Resident IV	07/22/2015 – 06/01/2019	1411	15 (motion, bed)

Note that the temporal analysis is performed sequentially on whatever native dimensional feature space is available from the resident sensor installation (column 4 in Table VI). We also provide a visualization of the results using two popular dimension-reduction algorithms, Principal Component Analysis (PCA) [78] and t-distributed Stochastic Neighbor Embedding (t-SNE) [79], to help visualize the data distributions in the clusters. The PCA algorithm is a linear transformation method, while the t-SNE algorithm is a non-linear embedding technique. Both algorithms require the full dataset to compute the projection, and hence, cannot, and were not, used for our streaming algorithm. But these projections, in a retrospective sense, provide good intuition for what happened in the higher dimensional sensor feature spaces.

7.1. Case study I – Resident I

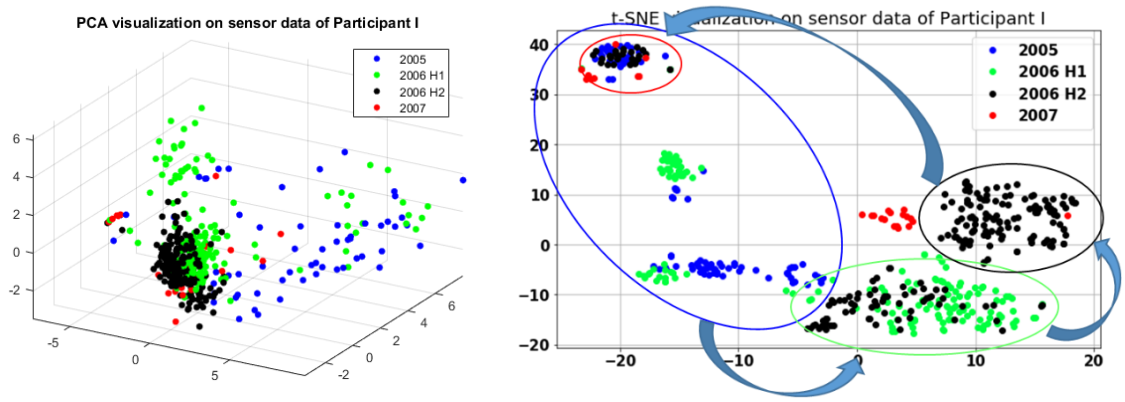
Resident I was the first resident of TigerPlace with sensors shortly after TigerPlace was built. This resident stayed for one and a half years from 10/10/2005 to 01/28/2007. Since Resident I is the first resident we studied, the depth sensor was not deployed at that time, and the features extracted from the motion and bed sensors are a little different from the features we have been using in recent years. However, it is still a good case for us to examine because there are several patterns in this resident's behavior, and we have many

observations (EHR notes) to validate our predictions. Table VII is the feature list used for Resident I.

Table VII. Feature list for Resident I

Feature Type	Measured Features
Motion	Number of nightly/daily bathroom visits, bedtime/wake up time, number of times out of bed in the night, time spent in each room (6 rooms: bathroom, bedroom, living room, closet, kitchen, entryway), aggregated motion firings (8 areas)
Bed	Aggregated restlessness firings (4 levels: low, medium, high, very high), aggregated pulse firings (2 levels: low, normal), aggregated breath firings (3 levels: low, normal, high), the total amount of nap time, total time in bed

The data visualization on Resident I is shown in Figure 20. The data are labeled by year, each with a different color. To better see how the data evolve, the year 2006 is split into two half years where "2006 H1" represents the first half-year of 2006 and "2006 H2" represents the second half-year of 2006. There are a few ellipses and curved arrows that are drawn manually in Figure 20 (b) according to the EHR ground truth retrospectively to help understand how data clusters evolve.

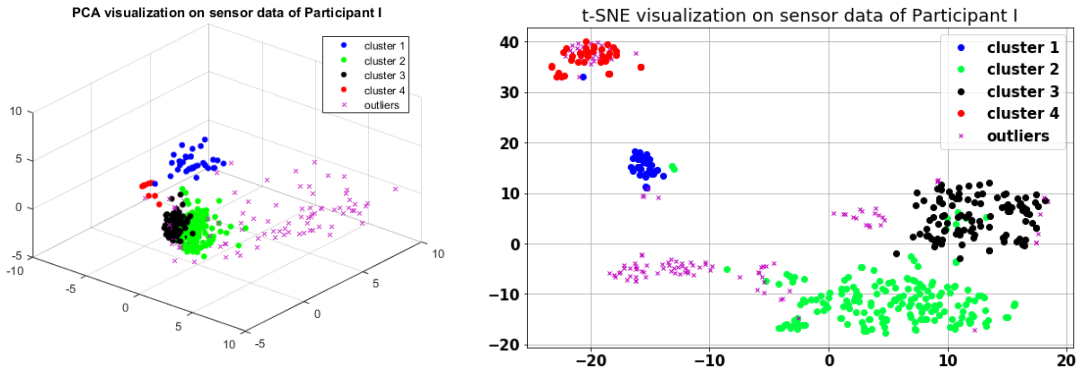


(a) (b)

Figure 20. Retrospective data visualization (projections from the original 20-dimensional data) for Resident I labeled by year using (a) PCA, (b) t-SNE.

During the first three months from October 2005, Resident I had an unstable behavior pattern, as shown in the blue elliptical region in Figure 20 (b). The resident passed out on

12/10/2005 when he was out with his daughter and had surgery on 12/20/2005. The resident had numerous EHR notes regarding his health issues during this period. In March 2006, the resident was back to a normal behavior pattern after the resident recovered from the surgery. In the middle of 2006, the resident's behavior pattern shifts slightly from the green elliptical region to the black elliptical region in Figure 20 (b) due to a decrease in motion behaviors. In November 2006, the behavior pattern shifted back to the top left of the red elliptical region in Figure 20 (b), which was a small partial region in the late 2005 unstable period (the blue elliptical region). On 01/02/2007, the resident had bleeding in the brain and then received hospice care.



(a) (b)

Figure 21. SPGMM clustering results of Resident I using retrospective data visualization (projections from the original 20-dimensional data): (a) PCA and (b) t-SNE.

The SPGMM algorithm is run on the high-dimensional multi-sensor data of Resident I. Figure 21 shows the clustering result of the SPGMM algorithm, projected into 3-space for visualization, that finds four clusters each representing a period of behavior patterns. Many points in the feature space do not directly belong to an existing cluster and are not dense enough to create a new cluster. They are marked as outliers. These are treated as abnormal days. The outlier detection methods have been studied in our previous research

[49],[80]; we focus on detecting the early signs (warnings) of abnormal days on data streams in this work.

The SPGMM algorithm analyzes the trajectories of data streams and tries to detect the trends of each day's streaming data. If SPGMM notices a trend of pattern shifting, it produces a warning message for clinicians to take some corresponding actions to the detected unstable trend. Figure 22 shows the timeline of different degrees of warnings generated by SPGMM and negative EHR notes on Resident I labeled by a clinician. The negative EHR notes may indicate some potential health issues.

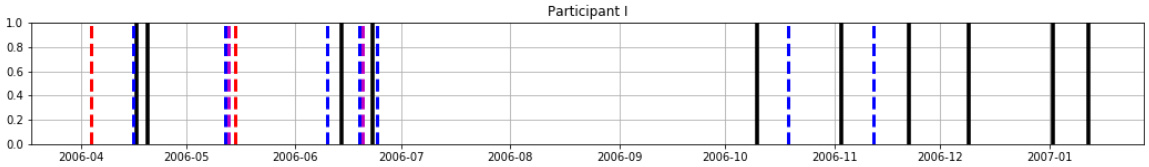


Figure 22. Warnings generated by SPGMM on Resident I. Black represents negative electronic health records (EHR); blue represents a weak warning; magenta, a medium warning; and red, strong warning

Table VIII lists the medium and strong warning dates generated by the SPGMM algorithm. The bolded dates are strong warnings. Related EHR notes are listed along with the warning dates.

Table VIII. Warning dates and related EHR notes of Resident I

Warning date	EHR date	EHR notes
04/04/2006	04/17/2006, 04/20/2006	"Discussed possible medication for depression", "Comp-oth cardiac device"
05/13/2006, **05/15/2006**	06/14/2006	"Begin to take Uroxatral"
06/20/2006	06/23/2006	"Gait unstable, decrease exercises in the last few weeks"

- On 04/04/2006, a strong warning was fired. After approximately two weeks, the resident discussed possible medication for depression with clinicians on 04/17/2006 and

was diagnosed with a "Comp-oth cardiac device" that is a complication due to the cardiac device implant and graft on 04/20/2006.

- On 05/13/2006 and 05/15/2006, there were both a medium warning and a strong warning. After around a month, the resident began to take the drug Uroxatral that can treat urinary problems caused by an enlarged prostate (benign prostatic hyperplasia) on 06/14/2006.

- On 06/20/2006, a medium warning was triggered. After three days, EHR notes record that the resident had an unstable gait and decreased exercises in the last few weeks on 06/23/2006.

Although SPGMM generated only four medium and strong warnings over one year span for resident I, each warning message was related to some of the resident's health issues, which indicates the warnings have some valuable information for clinicians to investigate further. There are some negative EHR notes undetected by the algorithm in the late period of this resident. The events that went undetected might have been due to the unstable measured features since Resident I was the first resident we investigated 14 years ago, and the features were not well tested at that time. The next three subsections introduce our more recent work on three residents in TigerPlace in the past few years.

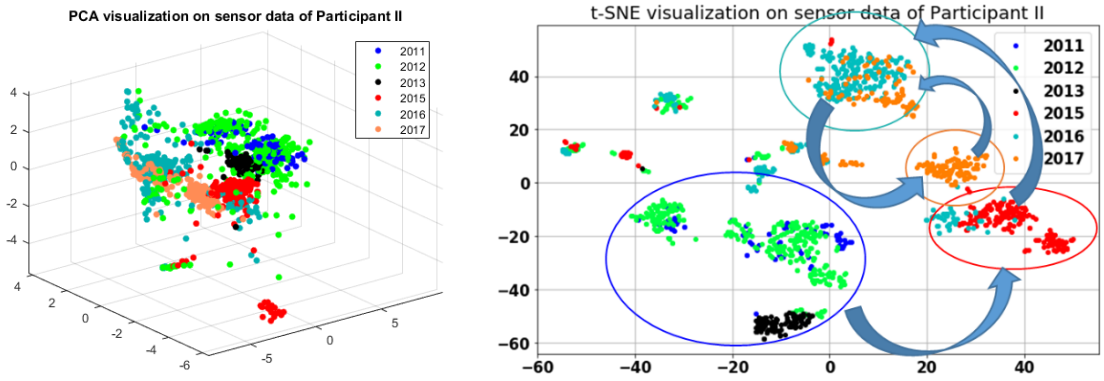
7.2. Case study II – Resident II

Resident II lived at TigerPlace from 2005 to 2017. The depth sensor for Resident II was deployed in 2011, so we use the sensor data starting in 2011. There is a period that this resident's bed sensor recordings were missed due to the equipment update of the bed sensor, so the in-home sensor data periods used for Resident II are 10/03/2011 – 04/07/2013 and 06/01/2015 – 09/01/2017. Table IX shows the 20 features used for analysis.

Table IX. Feature list for Resident II

Feature Type	Measured Features
Motion	Aggregated firings of bathroom, bedroom, closet, front door, kitchen, living room, shower, bed, den, office, laundry room
Bed	Pulse (3 levels), restlessness
Depth	Walking speed, stride time, stride length, stride density, resident's height

The data visualization, with manual labeling for Resident II is shown in Figure 23. The data are labeled by year, each with a different color. A few ellipses and curved arrows are drawn manually in Figure 23 (b) according to the EHR ground truth retrospectively to help understand how data evolves.

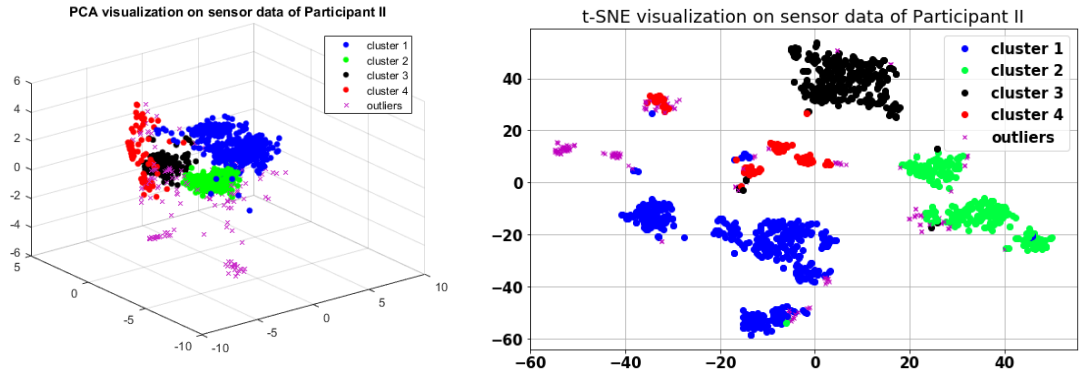


(a) (b)

Figure 23. Retrospective data visualization (projections from the original 20-dimensional data) for Resident II manually labeled by year using (a) PCA, (b) t-SNE.

During the first three years, from 2011 to 2013, Resident II was in a healthy and stable state, as shown in the blue elliptical region in Figure 23 (b). In 2015, the resident's behavior pattern jumped to the red elliptical region because the resident was diagnosed with mild cognitive impairment in the middle of 2015. At the end of 2015, the resident had surgery, and his behavior pattern shifted to the cyan elliptical region. In March 2017, the resident's behavior pattern returned to the orange elliptical region that was "close" to the red elliptical region due to the recovery from the surgery, and a new increasing swallowing problem

emerged in March. The resident had a low back problem in July 2017, causing the resident's pattern to shift back to the cyan elliptical region where his surgery period in 2015 is located. Retrospectively, these changes in physical function make sense. The question is, can they be discovered automatically by our unsupervised clustering and, more importantly, could they have been anticipated by the trajectory analysis added to SPGMM?



(a) (b)

Figure 24. SPGMM clustering results of Resident II using retrospective data visualization (projections from the original 20-dimensional data): (a) PCA, (b) t-SNE.

The SPGMM algorithm is run on the high-dimensional multi-sensor data of Resident II. Figure 24 shows the clustering result of the SPGMM algorithm that finds four clusters, each of which represents a period of behavior pattern. Figure 25 shows the timeline of different degrees of warnings and negative EHR notes on Resident II labeled by a clinician.

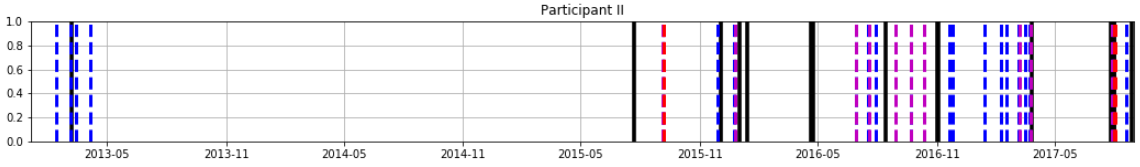


Figure 25. Warnings generated by SPGMM on Resident II. (black: negative EHR notes; blue: weak warning; magenta: medium warning; red: strong warning)

Table X lists the medium and strong warning dates generated by the SPGMM algorithm. The bolded dates are strong warnings. Related EHR notes are also listed along with the warning dates.

Table X. Warning dates and related EHR notes of Resident II

Warning date	EHR date	EHR notes
09/05-06/2015 **09/07/2015**	Middle 2015	"Diagnosed with mild cognitive decline"
12/25/2015	01/16/2016	"Surgery on C4 and C5 – declines physically and mentally"
06/29/2016		None
07/19/2016	08/14/2016	"Complaining of toenails hurting on R foot"
08/28/2016		None
09/22/2016		None
10/11/2016	11/03/2016	"Started on Donepezil HCL/Aricept 10 mg BID (for Dementia)"
03/08/2017		None
03/25/2017	03/27/2017	"Resident reports increasing swallowing problems"
07/29-30/2017, **07/31/2017**, **08/02/2017**	07/29/2017, 07/31/2017	"Begins to have incontinence on a regular basis", "Fall" Many records regarding health issues in the next two weeks

- From 09/05/2015 to 09/07/2015, two medium warnings and one strong warning were generated. These three warnings can be associated with the resident's diagnosis of mild cognitive decline in the middle of 2015.

- On 12/25/2015, a medium warning was triggered. After around 20 days, the resident had a surgery that caused him to decline physically and mentally on 01/16/2016. This medium warning reflected the resident's behavior changes before the surgery.

- On 07/19/2016, a medium warning was fired. This warning can be linked to the EHR notes on 08/14/2016 that the resident complained of toenails hurting on the right foot because the algorithm captured his decrease in motion.

- On 10/11/2016, a medium warning was generated. After around 22 days, the resident started to take drugs for dementia on 11/03/2016.

- On 03/25/2017, a medium warning was triggered. After two days, EHR notes show that the resident reported increasing swallowing problems on 03/27/2017.

- From 07/29/2017 to 08/02/2017, there were two medium warnings and two strong warnings. The resident began to have incontinence on a regular basis on 07/29/2017 and fell on 07/31/2017. There are many EHR notes in these two weeks about his unstable condition. The resident's family reported the resident experienced a sudden occurrence of inability to move his head, complained of pain, being lightheaded and incontinence on 08/28/2017.

Although some of the warnings are not linked to the existing EHR notes, most of the warnings generated by the SPGMM algorithm can be associated with some of this resident's health issues. The warnings reflect that the resident's sensor behaviors shift or change as some health problems happen, and the SPGMM algorithm does an excellent job of capturing the early signs of pattern shifting.

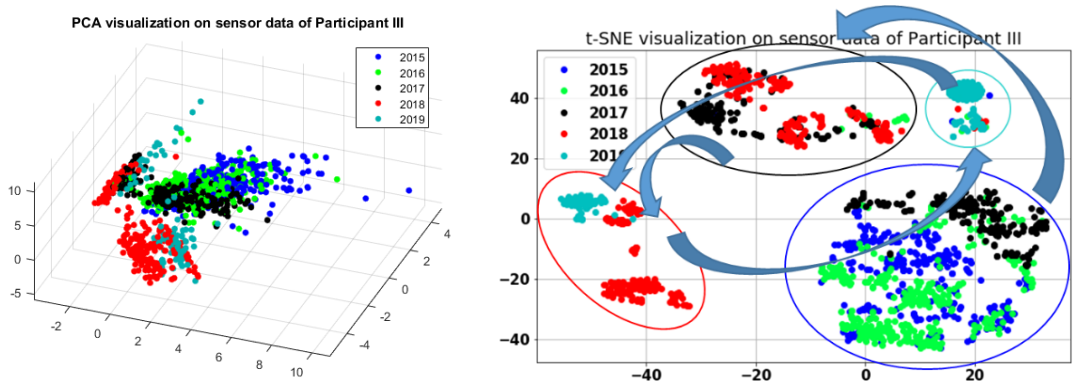
7.3. Case study III – Resident III

Resident III is an active resident who moved to TigerPlace in March 2015. The resident's in-home sensor data was dated from 03/25/2015 to 06/01/2019. Table XI is the feature list for Resident III. This feature list is close to the features extracted from Resident II except for those related to motion because of different motion sensor placement in the apartment.

Table XI. Feature list for Resident III

Feature Type	Measured Features
Motion	Aggregated firings of the bathroom, bedroom, closet, front door, kitchen, living room, shower, bed, patio, laundry room
Bed	Pulse (3 levels), restlessness
Depth	Walking speed, stride time, stride length, stride density, resident's height

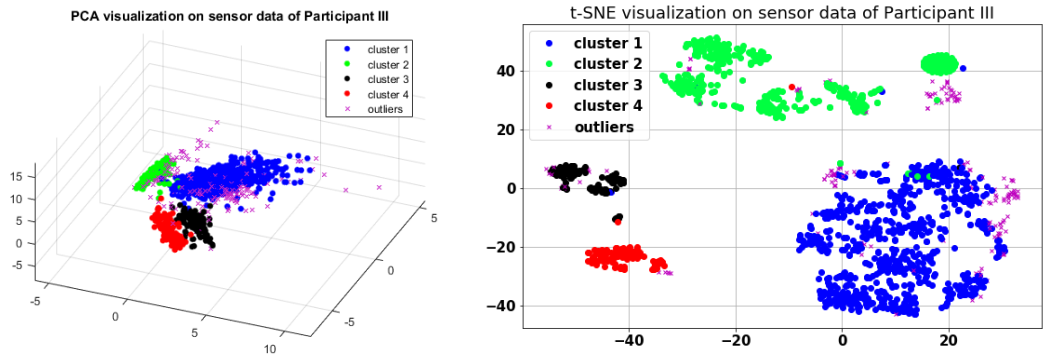
The data visualization on Resident III is shown in Figure 26. The data are labeled by year, each with a different color. There are a few ellipses and curved arrows that are drawn manually in Figure 26 (b) according to the EHR ground truth retrospectively to help understand how data evolves.



(a) (b)

Figure 26. Retrospective data visualization (projections from the original 19-dimensional data) for Resident III manually labeled by year using (a) PCA, (b) t-SNE.

Resident III's status had been stable since the resident moved to TigerPlace until September 2017. The resident had a severe left toe problem in October 2017, and the resident's early days' normal behavior pattern shifts from the blue elliptical region in Figure 26 (b) to the black elliptical region due to his decrease in motion. The second shift from the black elliptical region to the red elliptical region is not actually due to health changes, but rather it is a significant update of the sensor network that causes a measurement shift in sensors. On 01/02/2019, the resident fell to the ground and was admitted to the hospital with sepsis. At the same time, the resident's behavior pattern jumps to the cyan elliptical region until the beginning of April. On 04/03/2019, the resident was back in the apartment, and the resident's behavior pattern returned to normal (the red elliptical region).



(a) (b)

Figure 27. SPGMM clustering results of Resident III using retrospective data visualization (projections from the original 19-dimensional data): (a) PCA, (b) t-SNE.

The SPGMM algorithm is run on the high-dimensional multi-sensor data of Resident III. Figure 27 shows the clustering result of the SPGMM algorithm that finds four clusters; each of them represents a period of a particular pattern. Figure 28 shows the timeline of different degrees of warnings and negative EHR notes on Resident III labeled by a clinician.

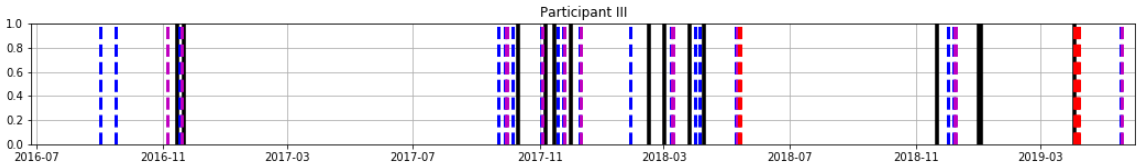


Figure 28. SPGMM warnings on Resident III. Black represents negative EHR notes; blue represents a weak warning; magenta represents a medium warning, and red shows a strong warning)

Table XII lists the medium and strong warning dates generated by the SPGMM algorithm. The bolded dates are strong warnings. Related EHR notes are also listed along with the warning dates.

Table XII. Warning dates and related EHR notes of Resident III

Warning date	EHR date	EHR notes
11/05/2016	11/15/2016	"Fell in the bathroom. BP was low. VS: T-97.6, HR 88, BP 100/52, O2 96%"
11/19/2016	11/21/2016	"In hospital, has Staph Infection in his neck"

09/29-30/2017	10/11/2017	“Spot on left toe problem”
11/03/2017	11/07/2017	“Blood and discharge from big toe undercast”
11/25/2017	12/01/2017	“Reports shortness of breath with his lack of activity from his foot has some problems with falling asleep”
12/11/2017		None
03/09-10/2018	03/26/2018, 04/09/2018	“Notice heel swelling from the ace wrap”, “Small skin tear on the right elbow”
05/11/2018 **05/13-14/2018**		None
12/08-09/2018	12/31/2018, 01/02/2019	“Has infected diabetic foot ulcer, c/o lower extremity weakness”, “Admitted to hospital with sepsis”
04/03-04/2019 04/06-07/2019	04/03/2019	“Back to apartment from the hospital”
05/19/2019		None

- On 11/05/2016, a medium warning was generated. After around ten days, the resident fell in the bathroom, and the resident's blood pressure was low on 11/15/2016.

- On 11/19/2016, a medium warning was triggered. After two days, the resident was admitted to the hospital and had Staph Infection in his neck on 11/21/2016.

- On 09/29-30/2017, two medium warnings were fired. After around 12 days, the resident was spotted with a left toe problem on 10/11/2017. These two warnings reflect the resident’s decrease in motion due to his toe problem.

- On 11/25/2017, a medium warning was generated. After around a week, the resident reported shortness of breath with his lack of activity from his foot and had some problems with falling asleep on 12/01/2017. The algorithm detected this health issue because of the unstable bed sensor.

- On 03/09-10/2018, two medium warnings were generated. These two warnings can be linked to the resident's two EHR notes: the resident was noticed heel swelling from the

ace wrap on 03/26/2018, and the resident had small skin tear on the right elbow on 04/09/2018.

- On 12/08-09/2018, there were two medium warnings. These can be associated with two EHR notes: the resident had an infected diabetic foot ulcer, complained of lower extremity weakness on 12/31/2018, and was admitted to the hospital with sepsis on 01/02/2019. The resident was then in the hospital until 04/03/2019. From 04/03/2019 to 04/07/2019, there were four strong warnings because the resident was back at TigerPlace. These four warnings reflected the in-home sensor patterns shifting from a period with few sensor firings back to regular behavior patterns. It is important to note that the warnings are not always due to the adverse health changes; they indicate the behavior changes from one pattern to another.

Looking at the timeline of this resident in Figure 28, we found most of the black lines (negative EHR notes) are very close to the dotted lines (different degrees of warnings). That indicates that most of the warnings generated by the SPGMM algorithm have some associations with the negative EHR notes that reflect health problems.

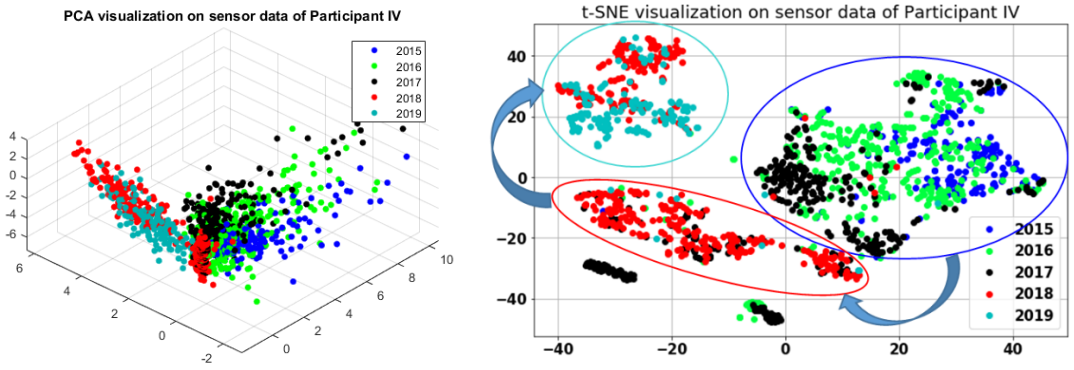
7.4. Case study IV – Resident IV

Resident IV is an active resident at TigerPlace. The resident has problems with walking and uses a wheelchair daily for mobility; therefore, the depth sensor data are not captured. The resident's motion and bed sensor data period is from 07/22/2015 to 06/01/2019. Table XIII displays the feature list for Resident IV.

Table XIII. Feature list for Resident IV

Feature Type	Measured Features
Motion	Aggregated firings of two bathrooms, two bedrooms, closet, front door, kitchen, living room, shower, patio, laundry room
Bed	Pulse (3 levels), restlessness

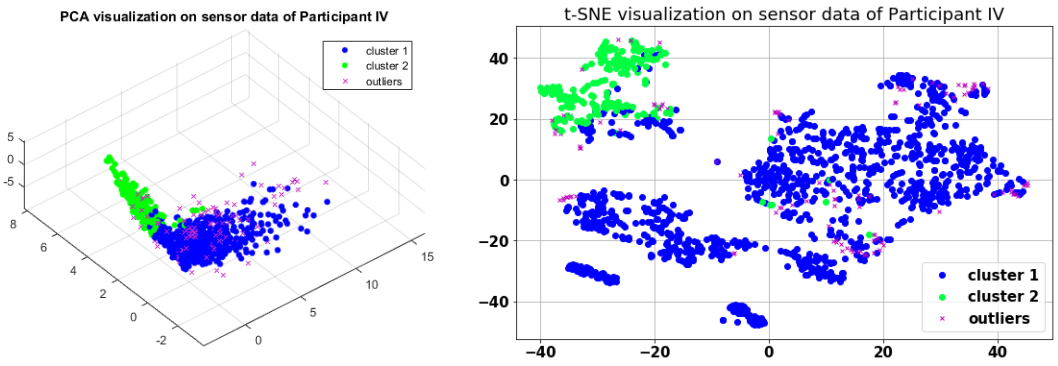
The data visualization on Resident IV is shown in Figure 29. The data are labeled by year, each with a different color. There are a few ellipses and curved arrows that are drawn manually in Figure 29 (b) according to the EHR ground truth retrospectively to help understand how data evolves.



(a) (b)

Figure 29. Retrospective data visualization (projections from the original 15-dimensional data) for Resident IV manually labeled by year using (a) PCA and (b) t-SNE.

Resident IV's behavior pattern had been stable since he moved into TigerPlace in 2015, as shown in the blue elliptical region in Figure 29 (b). In October 2017, the resident had a back-pain problem then his behavior pattern shifted to the red elliptical region. Just like Resident III, there was also an update of the sensor network that causes a measurement shift in sensors for Resident IV in the middle of 2018, which renders the pattern shift to the cyan elliptical region in Figure 29 (b).



(a) (b)

Figure 30. SPGMM clustering results of Resident IV using retrospective data visualization ((projections from the original 15-dimensional data)): (a) PCA, (b) t-SNE.

The SPGMM algorithm is run on the high-dimensional multi-sensor data of the Resident IV. Figure 30 shows the clustering result of the SPGMM algorithm that finds two clusters; each represents a period of a specific pattern. Resident IV has a stable health status, so he does not have as many behavior patterns or clusters as other residents. Figure 31 shows the timeline of different degrees of warnings and negative EHR notes on Resident IV labeled by a clinician.

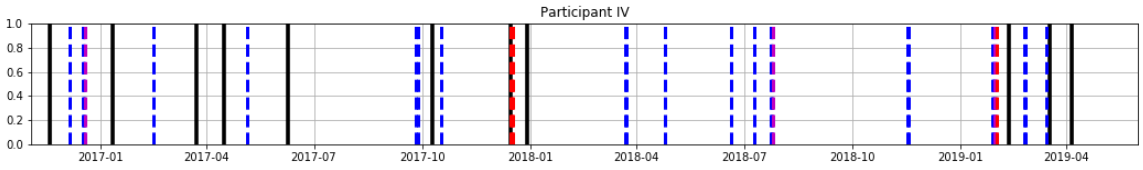


Figure 31. Warnings generated by SPGMM on Resident IV (black shows negative EHR notes; blue shows weak warning; magenta represents a medium warning, and red is a strong warning).

Table XIV lists the medium and strong warning dates generated by the SPGMM algorithm. The bolded dates are strong warnings. Related EHR notes are also listed along with the warning dates.

Table XIV. Warning dates and related EHR notes of Resident IV

Warning date	EHR date	EHR notes
12/19/2016	01/11/2017	“Fell in the bathroom”

12/15-17/2017	12/15/2017 & 12/29/2017	"Fell, found sitting on the floor"
07/26/2018		None
01/30-31/2019 **02/01/2019**	02/11/2019	Five vitals (BP-Systolic, O2 sats, Pulse, respiration, temperature)

- On 12/19/2016, a medium warning was generated. After 22 days, the resident had fallen in the bathroom on 01/11/2017.

- On 12/15-17/2017, three strong warnings were fired. These warnings can be associated with the fall incidents that happened on 12/15/2017 and 12/29/2017.

- From 01/30/2019 to 02/01/2019, there were two medium warnings and one strong warning. After around ten days, five sets of vitals reflected some abnormal clinical measures that were marked by the clinicians.

While this resident moved into TigerPlace in the same year (2015) as the Resident III, the warnings generated for this resident by the SPGMM algorithm are fewer than those of the Resident III most likely because this resident's health state is more stable. This resident has problems walking and uses a wheelchair in daily routines, so we observed many falls in his EHR notes. The algorithm only captured a few warnings related to the falls, perhaps due to the lack of gait information from a depth sensor.

7.5. Discussion

From subsection 7.1 to 7.4, we found many warnings produced by our trajectory analysis coupled with SPGMM that can be linked to the EHR notes. Hence, the warnings did predict related health events. We now provide numerical evaluations on the warnings for the above four residents.

If a warning date is followed by a negative EHR note within a window size of two weeks, this warning date will be validated as a suitable 2-weeks-validated warning. Likewise, if the window size is a month, the warning date will be validated as a suitable month-validated warning.

Table XV shows the evaluation of the medium and strong warnings using precision, recall, and F1 scores. The precision score measures how many warnings are related to negative EHR notes. The recall score measures how many negative EHR notes are detected by the algorithm. The F1 score is the harmonic mean of the precision score and recall score.

Table XV. Evaluation of SPGMM warnings

	Warnings	Negative EHR	2-Weeks validated			Month validated		
			Precision	Recall	F1	Precision	Recall	F1
Resident I	4	10	0.50	0.20	0.286	0.75	0.40	0.522
Resident II	15	17	0.333	0.294	0.312	0.60	0.588	0.594
Resident III	19	14	0.368	0.50	0.424	0.579	0.786	0.667
Resident IV	8	11	0.75	0.273	0.4	0.875	0.364	0.514

As we can see in the above table, the month-validated F1 scores are higher than the 2-week-validated F1 scores because the month-validated score depicts the relationship between a warning date and a more extended period (month vs. two weeks).

We compare our SPGMM algorithm with the 1-dimensional (1-D) algorithm that we are currently using in TigerPlace, a streaming clustering algorithm, DenStream [53], and a popular outlier detection algorithm, Isolation Forest (iForest) [81]. Table XVI shows the F1 score of different algorithms on the four residents' sensor data in TigerPlace. Note that Resident I did not have the 1-D algorithm alerts since the algorithm was not deployed at that time.

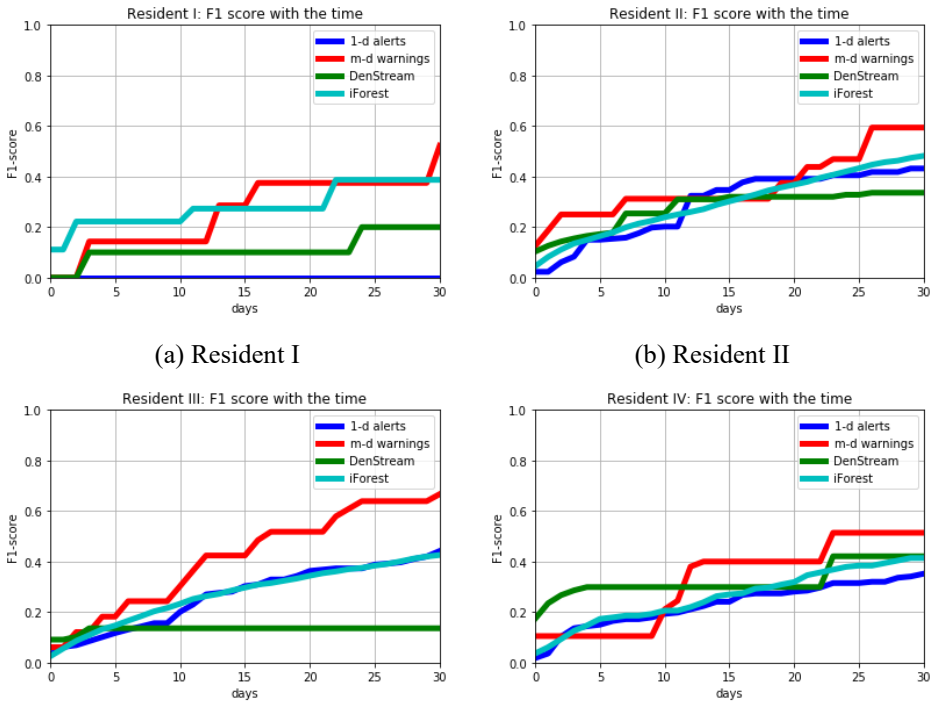
Table XVI. F1 scores of different algorithms

	2-Weeks validated				Month validated			
	1-D	DenStream	iForest	SPGMM	1-D	DenStream	iForest	SPGMM
Resident I	NA	0.1	0.273	**0.286**	NA	0.2	0.387	**0.522**
Resident II	**0.347**	0.31	0.288	0.312	0.432	0.336	0.482	**0.594**
Resident III	0.281	0.136	0.286	**0.424**	0.442	0.136	0.426	**0.667**
Resident IV	0.241	0.352	0.263	**0.4**	0.352	0.421	0.414	**0.514**
Average	0.290	0.225	0.278	**0.356**	0.409	0.273	0.427	**0.574**

As we can see in Table XVI, our SPGMM algorithm has the highest month-validated F1 score on all four residents and the highest 2-weeks-validated F1 score on Resident I, II, and IV. The main reason that the 1-D algorithm and the iForest algorithm do not work well is that they are not streaming algorithms that can learn new information in data streams. Once there a new pattern emerges in data streams, both the 1-D algorithm and the iForest algorithm will treat it as an outlier. DenStream, as a density-based streaming clustering algorithm, can create new clusters in the data stream, but it cannot separate close clusters and it also ignores the trajectory relationship in data streams. Our SPGMM can learn new patterns in data streams and adjust its structure to adapt to the new changes in data streams.

Besides using 14 days (2 weeks) and 30 days (1 month) to validate warnings, we compute F1 scores for each different day's validated warnings, as shown in Figure 32. The X-axis is the number of days. We computed a F1 score for each number of days x. We consider a warning is a good warning if it is within x days earlier before a negative EHR note.

If a warning date is followed by a negative EHR note within a window size of two weeks, this warning date will be validated as a suitable 2-weeks-validated warning.



(a) Resident I (b) Resident II

(c) Resident III (d) Resident IV

Figure 32. F1 score for each days' validated warnings generated by current 1-D alerts (blue), SPGMM (red), DenStream (green), and iForest (cyan)

As we can see, our multi-dimensional warnings have a higher F1 score than the other three algorithms on the longer period validated F1 score, especially for Resident III. This means our algorithm can provide an earlier indication of health changes. One of the drawbacks of our current 1-D algorithm is that it produces many alerts in one day because it examines one single feature at a time and produces alerts for different types of sensor features. The use of the multi-dimensional warning algorithm significantly reduces the number of alerts because it considers multiple sensor features at the same time and generates warnings afterwards.

The month-validated F1 score of Resident I and Resident IV are low compared with the scores of Resident II and Resident III, perhaps because both Resident I and Resident

IV did not have depth sensor data. In addition, Resident I used old features that were investigated 14 years ago, and Resident IV is quite a healthy and stable resident who did not have many behavior patterns. The scores of Resident II and Resident III are relatively high, and it has been shown in the above subsections that most of their warnings generated by SPGMM can be linked to the negative EHR notes labeled by a clinician. Furthermore, different degrees of warnings show an order of priority for clinicians to deal with these warnings. This chapter's result provides support to show that the warnings generated by SPGMM are associated with impending health problems. These warnings provide sufficient and flexible information for clinicians to recognize early signs of health changes.

7.6. Automatic ground truth via CLAMP

SPGMM generates three levels of warnings (weak, medium, and strong), and we used the negative EHR notes labeled by a clinician to evaluate the warnings. The negative EHR notes may indicate some potential health issues. However, the process of a clinician manually evaluating warnings may take a huge amount of time and effort. Furthermore, a clinician can sometimes be biased, resulting in a negative EHR note being classified as positive by another clinician because different clinicians have different standards of evaluating negative EHR notes. Thus, we developed a clinical natural language processing toolkit to automate this labeling process without human bias. In this section, I will introduce the clinical language annotation, modeling, and processing (CLAMP) model and discuss how it can be used as an automatic ground truth toolkit to classify EHR notes.

7.6.1. Introduction to Clinical Language Annotation, Modeling, and Processing (CLAMP)

CLAMP [82] is a newly developed user-friendly clinical natural language processing (NLP) toolkit that not only provides state-of-the-art NLP components, but also a graphic user interface that can help users quickly build customized NLP pipelines for their individual applications.

CLAMP follows a pipeline-based architecture that decomposes an NLP system into multiple components. Most CLAMP components have been validated in multiple clinical NLP challenges, such as the informatics for Integrating Biology and the Bedside (i2b2), an NIH-funded National Center for Biomedical Computing (NCBC) based at Partners HealthCare System in Boston, MA [83] and through the computational semantics and semantic evaluation (SemEVAL) (encoding to evaluate unique identifiers task) [84]. Various technologies, including machine learning-based methods and rule-based methods, were used when developing these components. A list of CLAMP's available components is:

- Sentence boundary detection
- Tokenizer
- Part-of-speech tagger
- Section header identification
- Abbreviation reorganization and disambiguation
- Named entity recognizer
- Assertion and negation
- UMLS encoder
- Rules engine

In addition to the above NLP components, CLAMP has 338 clinical notes with entity annotations derived from medical transcription sample reports (see MTSamples at www.mtsamples.com), a collection of different types of transcribed clinical note examples made for clinical documentation education. This is a test corpus released with CLAMP.

Figure 33 shows a screenshot of the main interface of CLAMP for building an NLP pipeline. Built-in NLP components are listed in the top-left palette, and the user-defined MLP pipelines are displayed in the left-bottom palette. The details of each pipeline are displayed in the center area after users select and click the component they want in the listed areas of interest.

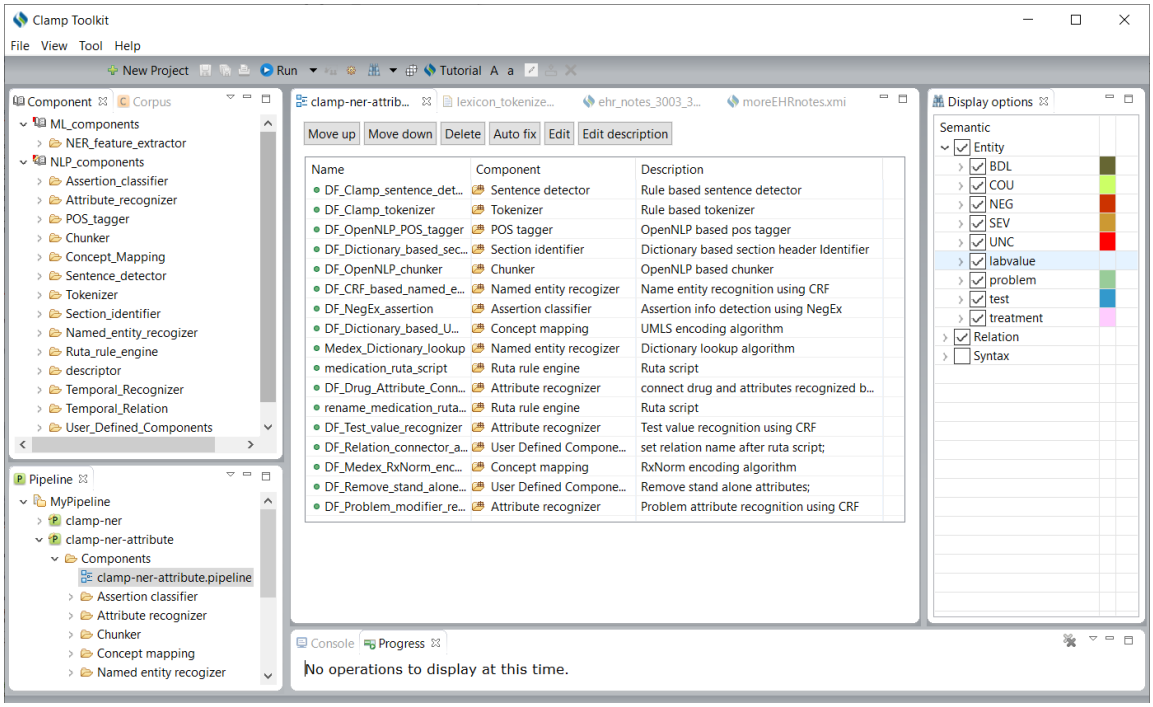


Figure 33. The user interface of CLAMP

We treat this automatic labeling task of negative EHR notes as a named entity recognition (NER) task. We mainly focus on the "problem" entity that indicates potential health issues. If one EHR note has a "problem" entity without a negation term linked to it, this note will be labeled as a negative EHR note. For the NER task, three annotated corpora

are included to pretrain the model following the guidelines in the i2b2 challenges [83]: (1) the discharge summaries used in the 2010 i2b2 challenge (i2b2 with, 871 annotated notes), (2) a new corpus of outpatient clinic visit notes from the University of Texas Health Science Center at Houston (with 1,351 UTNotes), and (3) a new corpus of mock clinical documents from MTSamples (with 338 notes).

The machine learning classification model used in CLAMP is Conditional Random Field (CRF) [85]. We could further retrain the CRF model with more annotated EHR notes labeled by our clinicians, but we find the released pretrained CRF model by CLAMP has already done a decent job of detecting potential health issues (see below results). One issue with the pretrained CRF model is that it does not recognize the residents falling problem because the model treats the falling problem as an emergent and unexpected problem, not a health issue. However, CLAMP provides us with an option to post-process the results by checking the words in EHR notes with a predefined dictionary list. Therefore, we added a rule that flags the words “fall”, “fell”, “falling” as a “problem” entity so that CLAMP can detect the resident falling problem.

To facilitate building machine learning-based NER modules on local data, CLAMP provides interfaces for corpus annotation and model training. Figure 34 shows the annotating entities and related results on the EHR notes.

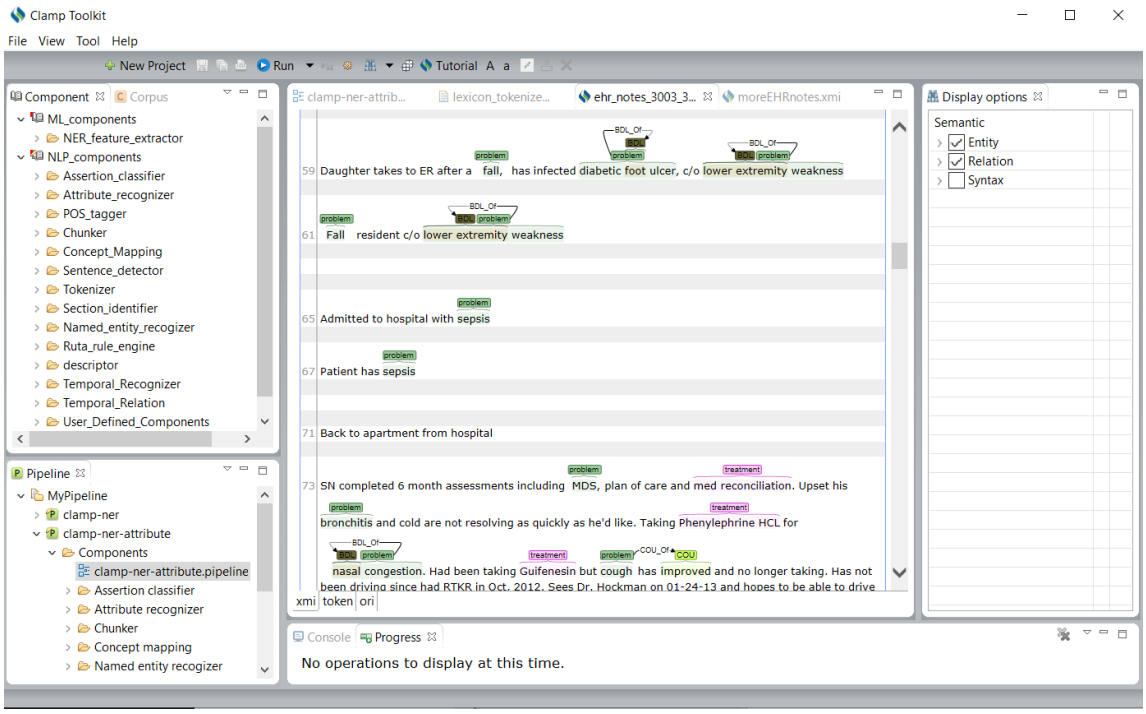


Figure 34. The interface in CLAMP for annotating entities and relationships

For example, the note "*Daughter takes* (the resident) *to ER after a fall, has infected diabetic foot ulcer, c/o lower extremity weakness*" in Figure 34, "fall," "diabetic foot ulcer," and "lower extremity weakness" are each labeled as a "problem" by CLAMP so this note is labeled as a negative EHR note.

7.6.2. Results

In this section, I used CLAMP to evaluate one of the TigerPlace residents' notes (Resident III in Section 7.3). This resident had 27 notes in total and two independent clinicians label the 27 EHR notes. We compared the performance of CLAMP with two clinicians' ratings. Figure 35 shows the comparison between two clinicians' ratings and the CLAMP labeling.

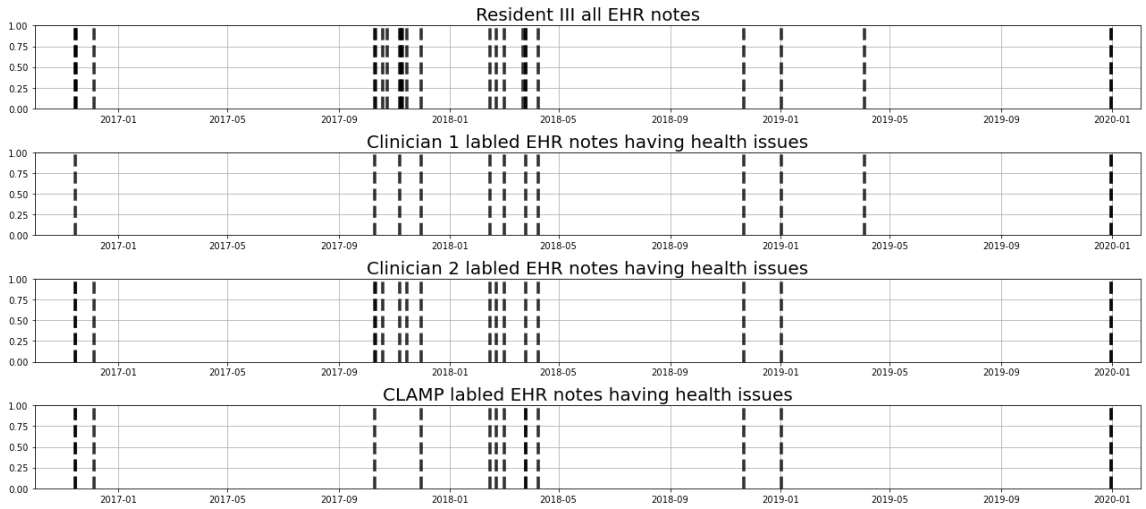


Figure 35. Comparison between two clinicians' ratings and CLAMP labeling

Each section of notations in Figure 35 represents a timeline over a period of four years from November 2016 to January 2020. The top section reveals all the EHR notes for Resident III. The second section shows the health issue EHR notes labeled by Clinician 1. The third section accounts for the health issue EHR notes labeled by Clinician 2, and the bottom section reveals the health issue EHR notes labeled by CLAMP.

As we can see in Figure 35, not even two clinicians could agree 100% on all EHR health issues, which confirms our argument that humans sometimes have different interpretations of the same evidence. The negative EHR notes labeled by CLAMP correlate with many of the negative EHR notes labeled by the two clinicians, showing an excellent performance of CLAMP for detecting potential health issues from EHR notes.

We conducted a numerical comparison between the three raters (two clinicians and CLAMP). Cohen's Kappa coefficient (K) is a statistic that is used to measure interrater reliability for qualitative (categorical) items [86]. It is generally thought to be a more robust measure than simple percent agreement calculation, as K considers the possibility of the agreement occurring by chance.

Cohen's kappa coefficient measures the agreement between two raters who each classify *N* items into *C* mutually exclusive categories. The definition of *K* is

$$K = \frac{p_0 - p_e}{1 - p_e} = 1 - \frac{1 - p_0}{1 - p_e} \tag{29}$$

where p_0 is the relative observed agreement among raters, and p_e is the hypothetical probability of chance agreement, using the observed data to calculate the probabilities of each observer randomly seeing each category by equation (30).

$$p_e = \frac{1}{N^2} \sum_k n_{k1} n_{k2} \tag{30}$$

where n_{ki} is the number of times rater *i* predicted category *k*.

In our case of binary classification (health-related labeling or not health-related labeling), Cohen's kappa coefficient can be simplified using equation (31).

$$K = \frac{2 \times (TP \times TN - FN \times FP)}{(TP + FP) \times (FP + TN) + (TP + FN) \times (FN + TN)} \tag{31}$$

where *TP* are the true positives, *FP* are the false positives, *TN* are the true negatives, and *FN* are the false negatives.

Table XVII shows Cohen's kappa score of three raters (two clinicians and CLAMP) on Residents II, III, and IV.

Table XVII. Cohen's kappa score of three raters (two clinicians and CLAMP) on Resident II, III, IV

	Clinician 1 – Clinician 2	CLAMP – Clinician 1	CLAMP – Clinician 2
Resident II	0.57	0.51	0.63
Resident III	0.49	0.56	0.62
Resident IV	1.00	1.00	1.00
Average	0.69	0.69	0.77

The interpretation of Cohen's kappa coefficient is shown in Table XVIII.

Table XVIII. The interpretation of the Cohen's kappa coefficient

Cohen's Kappa statistic (K)	Strength of agreement
< 0.00	Poor
0.00-0.20	Slight
0.20-0.40	Fair
0.40-0.60	Moderate
0.60-0.80	Substantial
0.80-1.00	Almost perfect

As shown in Table XVII and Table XVIII, the average agreement between the two clinicians ($K = 0.69$) is substantial. This agreement makes sense because both clinicians have a strong related background in nursing and healthcare. The average agreements between CLAMP and each of the clinicians are also substantial ($K = 0.69\ and\ K = 0.77$), which makes CLAMP a useful clinical NLP tool to detect potential health issues from EHR notes. Using CLAMP not only helps clinicians save substantial time and effort in labeling negative EHR notes, but it also eliminates human bias during the labeling process.

CHAPTER 8. ALGORITHM EXPLAINABILITY

Machine learning algorithms are able to provide accurate predictions, but lack the ability to explain to users how those predictions are made. The proposed algorithm in this work, the sequential possibilistic Gaussian mixture model (SPGMM), works great on predicting older people's potential health issues, but it cannot offer an explanation to either the older patients or the clinicians on what factors cause the potential health issues. Explainable AI (XAI) tries to provide explanations for machine decisions and predictions to justify their reliability [87]. This chapter presents four methods to explain the output of the SPGMM algorithm. In section 8.1, we used the linguistic summary method on each sensor reading to interpret the warnings generated by SPGMM. In section 8.2, different annotation types from the EHR notes are grouped together in each cluster to visualize the annotation distributions of each cluster. Section 8.3 presents a popular machine learning explainable approach called SHAP to explain the output of SPGMM. In section 8.4, the functional health (FH) score and SPGMM-generated clusters are analyzed together to discuss their potential relationships.

8.1. Linguistic summarization for warnings

In [88], a linguistic summarization algorithm was proposed to translate sensor data to natural language using fuzzy logic. This translation method can be applied to SPGMM to interpret the warnings. If there are one or more warnings generated for a resident, the sensor data leading to the current day is summarized. All health features are examined to check if there is an unusual change in any one of the sensors; then a clinician is contacted and asked to consider evaluating the unusual sensor readings before making any judgments.

Figure 36 shows an example in which three medium warning dates generated by SPGMM are marked and a linguistic summarization is conducted on the three marked warning dates. The three medium warning dates were selected to analyze because those are followed by negative EHR notes.

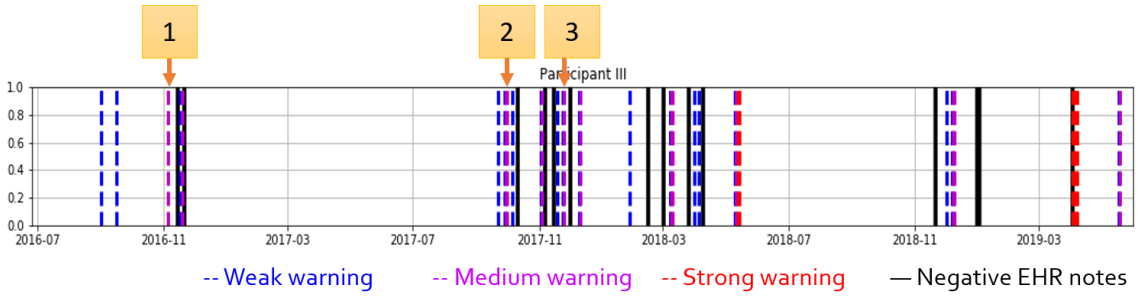


Figure 36. The warnings and the negative EHR notes for resident III with three medium warning dates are marked (1: 11/5/2016; 2: 9/30/2017; 3: 12/25/2017)

On the first warning date (11/5/2016) in Figure 36, the linguistic summary reads as follows: "*In the past two weeks, there were many days with high walking speed & high stride length, low respiration rate & high walking speed, low respiration rate & high stride length. Day & night-time overall activity, day-time bed restlessness, night-time pulse rate (from 63.12 to 80.83 beats/min), overall walking speed (from 52.1 to 54 cm/sec) and overall stride length (from 68.3 to 70.3 cm) have been increasing for the past 10 days. Night-time spent in bathroom have been decreasing for the past 13 days.*". Ten days later (11/15/2016), we have a negative EHR note recorded that states: "*Fell in the bathroom. BP was low. VS: T-97.6, HR 88, BP 100/52, O2 96%.*" Before this resident's falling in the bathroom, the resident's walking speed and stride length were unusually high according to the linguistic summary, which could be an early signal of falling.

On the second warning date (9/30/2017) in Figure 36, the linguistic summary is "*Night-time bed restlessness have been increasing for the past 11 days. Day-time pulse rate (from 92.73 to 60.25 beats/min) have been decreasing for the past 11 days. Day-time*

respiration rate (from 11.73 to 14.79 breaths/min), overall stride time (from 1.3 to 1.4 sec) have been increasing for the past 11 days. Day-time bed restlessness, day-time time in bed, day-time pulse rate (from 92.73 to 62.42 beats/min) have been decreasing for the past 12 days." Eleven days later (10/11/2017), we have a negative EHR note in record that shows "*Daughter noticed spot on resident's left toe. Big toe was inflamed with no signs of infection, warmth, or drainage. Callous left big toe has turned black on the surface. Resident was encouraged to wear loose shoes to decrease pressure.*" The multiple unusual patterns of bed sensors could be a reason for a clinician to check because the resident might have a hard time falling asleep due to the left toe problem.

On the last warning date (12/25/2017) in Figure 36, the linguistic summary states: "*Day-time overall activity, day & night-time bed restlessness have been increasing for the past 11 days. Night-time overall activity, night-time pulse rate (from 71.67 to 54.56 beats/min), overall walking speed (from 48.7 to 47 cm/sec), and overall stride length (from 66.6 to 64.3 cm) have been decreasing for the past 9 days.*" Six days later (12/1/2017), we have a negative EHR note in record that shows "*SF12 Assessment performed. Reports shortness of breath with his lack of activity* (caused by) *his foot;* (he) *has plenty of interest in doing things, has some problems with falling asleep.*" The overall increasing restlessness might explain the resident's falling asleep problem. The restlessness value tends to increase if the resident tosses and turns on bed and cannot fall asleep.

While I only showed a snapshot of three examples, the linguistic summaries are available for all warning dates and can be used as explanations. Each feature on the warning date is examined to provide support for clinicians to understand the residents' behavior changes.

8.2. Annotation distribution of clusters

The output of SPGMM not only includes three levels of warnings but also a cluster index, which indicates which cluster (pattern) each day belongs to. However, we do not know the differences among clusters. In this section, I used the annotation types from the EHR notes to analyze each cluster's patterns.

There are six annotation types in the EHR notes:

- **sosy** (Sign of Symptom): pain, discomfort, sore, constipation, etc.
- **medication**: Combivent Aerosol 18-103 MCG/ACT, etc.
- **medd** (Medical Device): tape, gauze, inhaler, etc.
- **phsu** (Pharmacologic Substance): Symbicort, Spiriva, Lasix, etc.
- **topp** (Therapeutic or Preventive Procedure): foam dressing, wound care, etc.
- **parf** (Pathologic Function): infection, bleeding, side effects, etc.

Cluster #	Number of days	Having annotation days
Cluster #1	732	25 (3.4%)
Cluster #2	415	58 (14%)
Cluster #3	102	24 (23.5%)
Cluster #4	106	4 (3.8%)

Figure 37. Annotation type statistics of each cluster for Resident III

Figure 37 provides a resident example (Resident III) to show how annotation types help caregivers and researchers to understand cluster patterns. Among the 1,530 days that we devoted to this study, 732 days were in cluster 1 (25 days with annotation notes), 415 days are in cluster 2 (58 days with annotation notes), 102 days are in cluster 3 (24 days with annotation notes), and 106 days in cluster 4 (4 days with annotation notes). The distribution of annotation types in each cluster is shown in Figure 38.

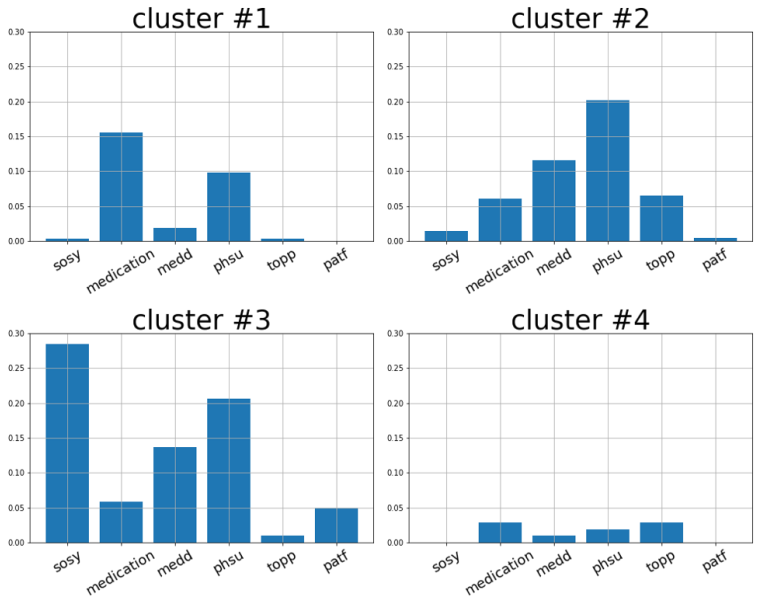


Figure 38. The distribution of annotation types in each cluster for Resident III

As shown in Figure 38, the resident took many medications and pharmacologic substances (phsu) during the cluster 1 period. As the resident shifted to cluster 2, his pharmacologic substances (phsu) increased, and he started to use some medical devices (medd). In cluster 3, the resident had several symptomatic signs (sosy, including pain, discomfort, soreness, constipation) and pathologic functions (patf, including infection, bleeding, and side effects). Cluster 3 was clearly the period in which the resident had some health issues. In Cluster 4, the resident went back to normal and his symptomatic signs (sosy) and pathologic function (patf) went away. After observing the distribution of annotation types in each cluster, the overall symptoms and results were easier to understand based on the activities during each cluster period.

8.3. SHAP

A sophisticated machine learning algorithm can usually produce accurate predictions, but its “black box” nature does not help adoption at all. Scott Lundberg (owner of the

lundberg-shap repository on GitHub (github.com/slundberg/shap) defines his model, SHAP, [89] as:

> "SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions." – Scott Lundberg, 2021

In order to use SHAP to explain the output of SPGMM, I used the cluster index generated by SPGMM as a target label and treated it as a supervised learning classification problem. The benefits of SHAP are its global interpretability and its local interpretability. The global interpretability helps us answer the question: "What is this cluster?" The collective SHAP values can show how much each predictor contributes, either positively or negatively, to the target variable (the cluster index in this case). This is like the variable importance plot, but it is able to show the positive or negative relationship for each variable with the target (the cluster index). The local interpretability helps us determine "Why a particular point belongs to this cluster?" Each observation gets its own set of SHAP values. This greatly increases its transparency. We can explain why a data point belongs to its prediction (the cluster index) and the contributions of the features. Traditional variable importance algorithms only show the results across the entire population but not on each individual case. The local interpretability enables us to pinpoint the contrast and its effects on the factors.

I used the same case study (Resident III) in the above section to provide an example for global interpretability.

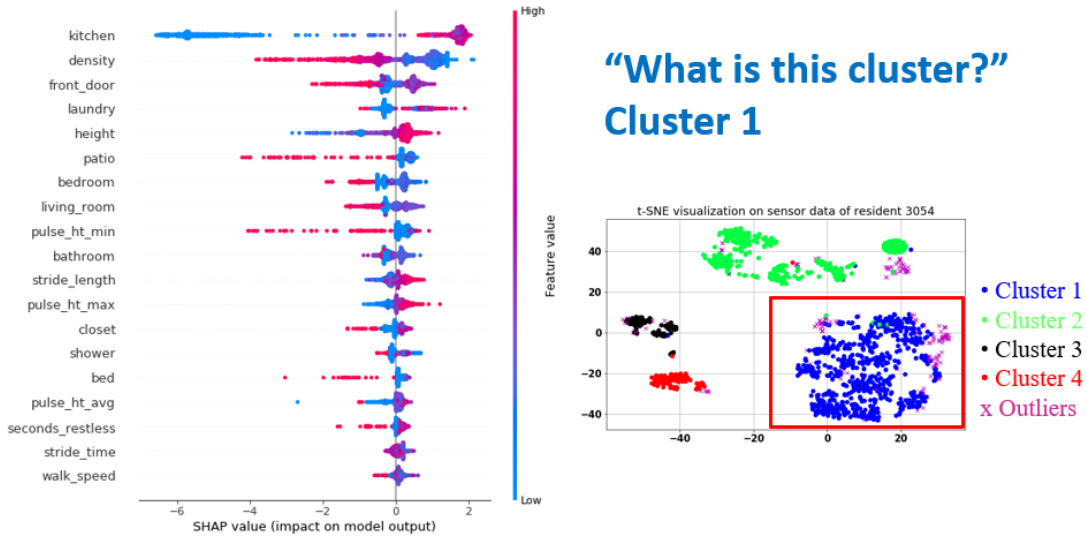
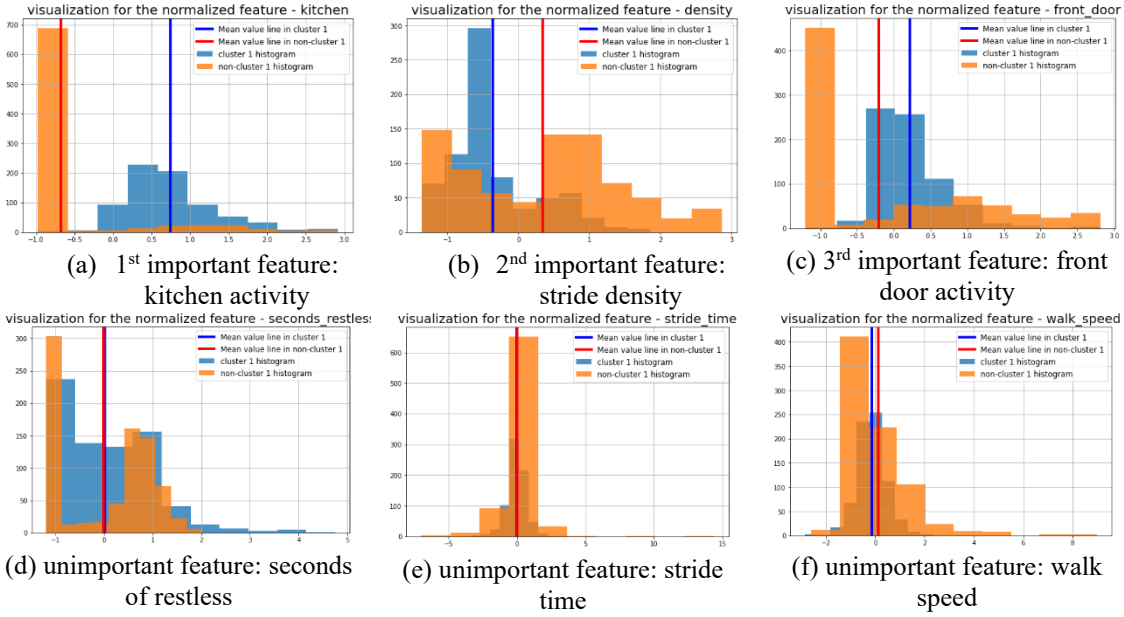


Figure 39. SHAP explanation for cluster 1 of Resident III

Figure 39 shows the SHAP value (impact on model output) for cluster 1 of Resident III. The y-axis indicates the variable name, in the order of importance from top to bottom. The x-axis is the SHAP value that indicates how much each feature contributes to cluster 1. Each point represents a row from the original dataset. The gradient color change from blue to red represents the value change from low to high. The higher the feature is on the SHAP table, the more contributions the feature makes to cluster 1. For example, the kitchen activity value is high (colored in red), so it is very likely to positively contribute to cluster 1. The stride density is low (colored in blue), and so, it is likely to positively contribute to cluster 1. The features whose SHAP values are around zero have little contributions to cluster 1 so those features are not important and are usually listed on the bottom of the SHAP table. Hence, the cluster 1 SHAP summary is: high kitchen activity, low stride density activity, high front door activity, high laundry activity, high stride height, and so on (the remaining features have little positively/negatively contributions to cluster 1).



(a) 1st important feature: kitchen activity

(b) 2nd important feature: stride density

(c) 3rd important feature: front door activity

(d) unimportant feature: seconds of restless

(e) unimportant feature: stride time

(f) unimportant feature: walk speed

Figure 40. Feature distribution of (a)-(c) important features, and (e)-(f) unimportant features, in cluster 1 vs. non-cluster 1, where orange histogram is cluster 1, blue histogram is non-cluster 1, blue vertical line is the mean feature value of cluster 1, and red vertical line is the mean feature value of non-cluster 1

In order to make sure the SHAP summary makes sense, Figure 40 shows the histogram distribution plots for the top three important features and the top three unimportant features where cluster 1 and non-cluster 1 are plotted in different colors. The blue lines (the mean feature value of cluster 1) in Figure 40 are compared with the red lines (the mean feature value of non-cluster 1) to validate the SHAP summary. For example, the high kitchen activity is the most important SHAP summary for cluster 1. In Figure 40 (a), the blue line (the mean value of kitchen activity value in cluster 1) is much larger than the red line (the mean value of kitchen activity value in non-cluster 1), which confirms that cluster 1 has a higher kitchen activity than other clusters. However, in Figure 40 (d)-(f), the mean feature values of cluster 1 and non-cluster 1 are very close and are difficult to discriminate, which makes them unimportant features for cluster 1. Hence, the SHAP summary does a good job of reflecting the true distributions of important features in a cluster.

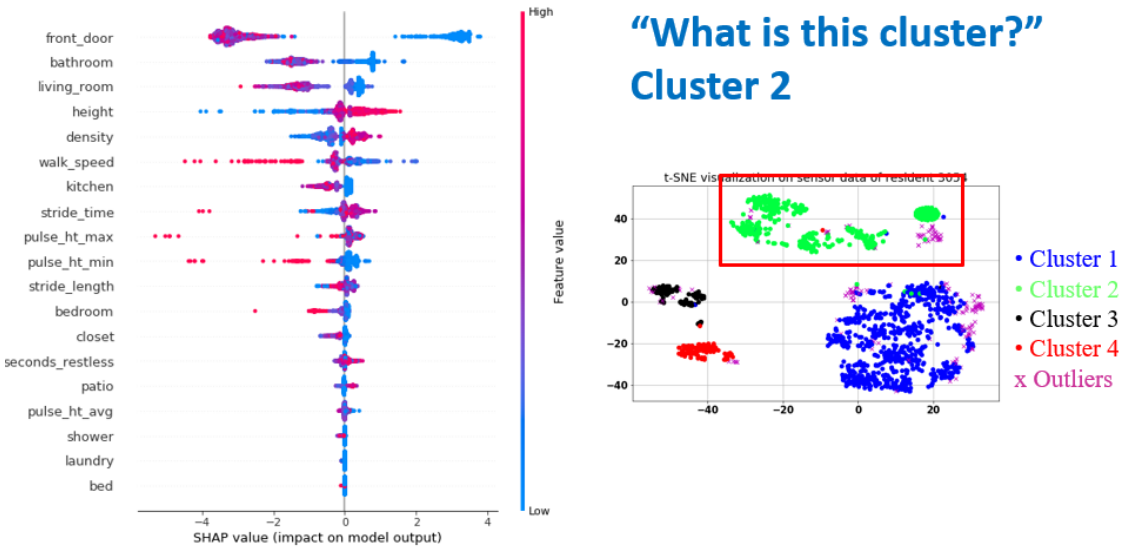


Figure 41. SHAP explanation for cluster 2 of Resident III

Figure 41 shows the SHAP value (impact on model output) for cluster 2 of Resident III. The cluster 2 SHAP summary is: low front door activity, low bathroom activity, low living room activity, high stride height, high stride density, low walk speed, and so on (the remaining features have little positively/negatively contributions to cluster 2).

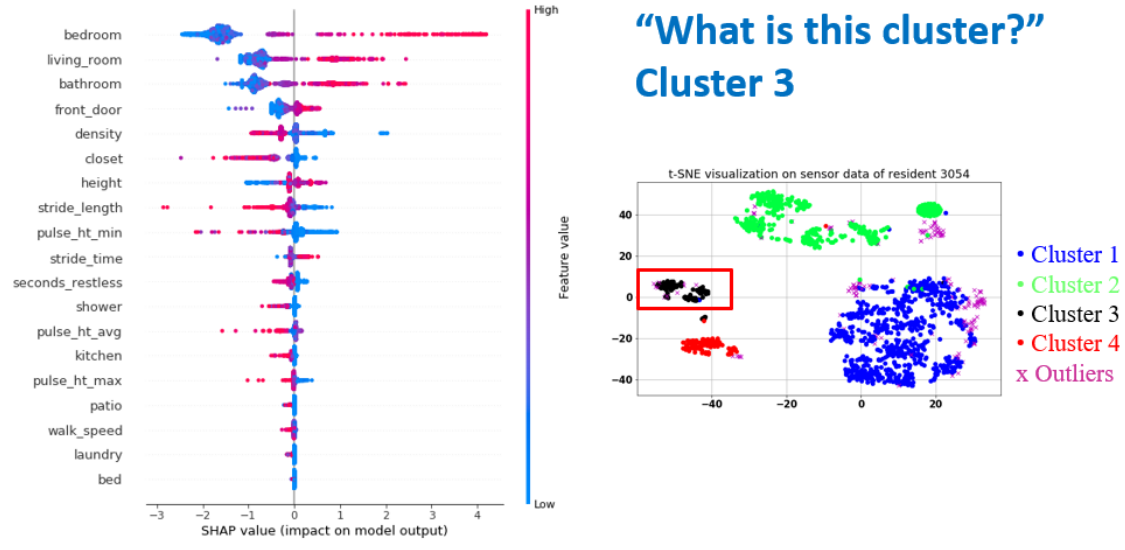


Figure 42. SHAP explanation for cluster 3 of Resident III

Figure 42 shows the SHAP value (impact on model output) for cluster 3 of Resident III. The cluster 3 SHAP summary is: high bedroom activity, high living room activity, high bathroom activity, high front door activity, low stride density, and so on (the remaining features have little positively/negatively contributions to cluster 3).

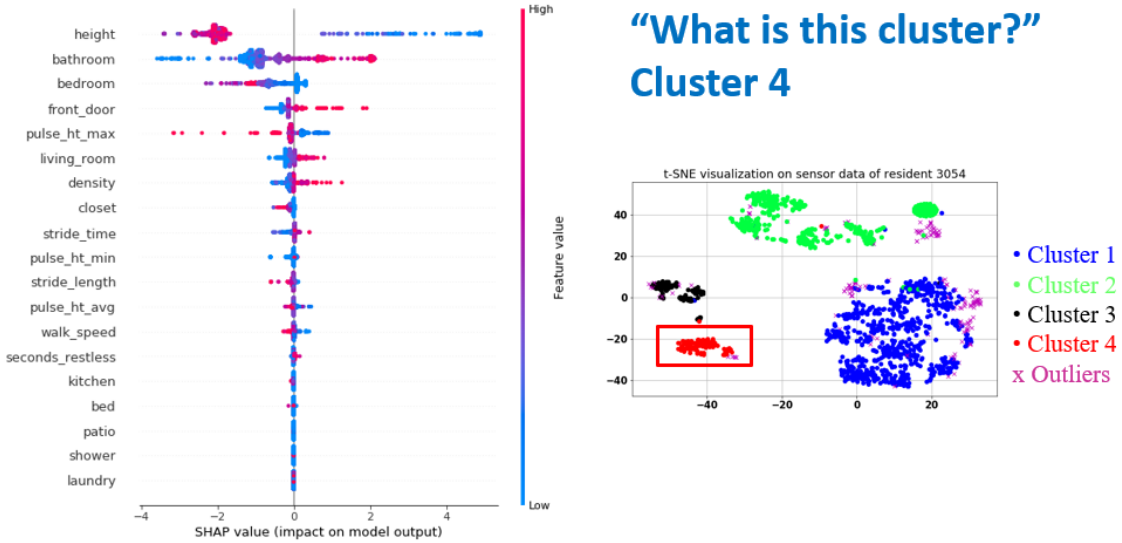


Figure 43. SHAP explanation for cluster 4 of Resident III

Figure 43 shows the SHAP value (impact on model output) for cluster 4 of Resident III. The cluster 4 SHAP summary is: low stride height, high bathroom activity, low bedroom activity, high front door activity, low pulse rate, high living room activity, and so on (the remaining features have little positively/negatively contributions to cluster 4).

Besides global interpretability, SHAP also provides local interpretability to explain why a particular day belongs to a cluster. Figure 44 shows the SHAP explanation of which cluster the day 03/26/2015 belongs to. Cluster 1 has the highest score (8.14) so the day 03/26/2015 belongs to cluster 1. Besides, the SHAP explanation shows that the kitchen activity, the stride density, and the laundry activity have a positive impact when identifying

cluster 1, and the front door activity and shower activity have a slightly negative impact when identifying cluster 1.

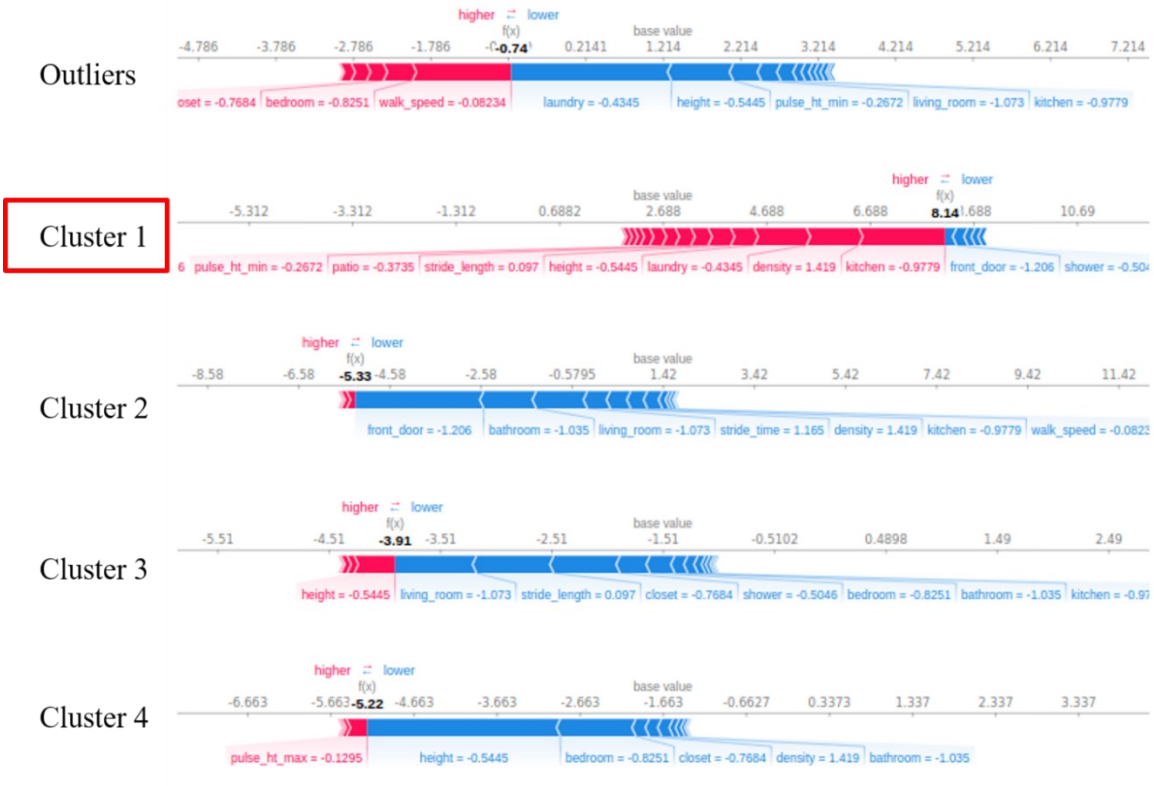


Figure 44. SHAP explanation of which cluster the day, 03/26/2015, belongs to

8.4. Functional health score

Higher levels of functional health in older adults lead to a higher quality of life and improve the ability to age in place. Tracking functional health objectively could help clinicians make decisions for interventions in cases of health deterioration. In this section, I compared the SPGMM clustering results with the functional health (FH) score [11] to see their connection. FH used geriatric assessment data collected from older adults to develop and validate a functional health prediction model based on risks associated will falls, hospitalizations, emergency visits, and death. It used mixed effect logistic regression to construct the model. The geriatric assessment included information from Activities of

Daily Living (ADL), Instrumental Activities of Daily Living (IADL), Mini-Mental State Examination (MMSE), Geriatric Depression Scale (GDS), and Short Form 12 (SF12).

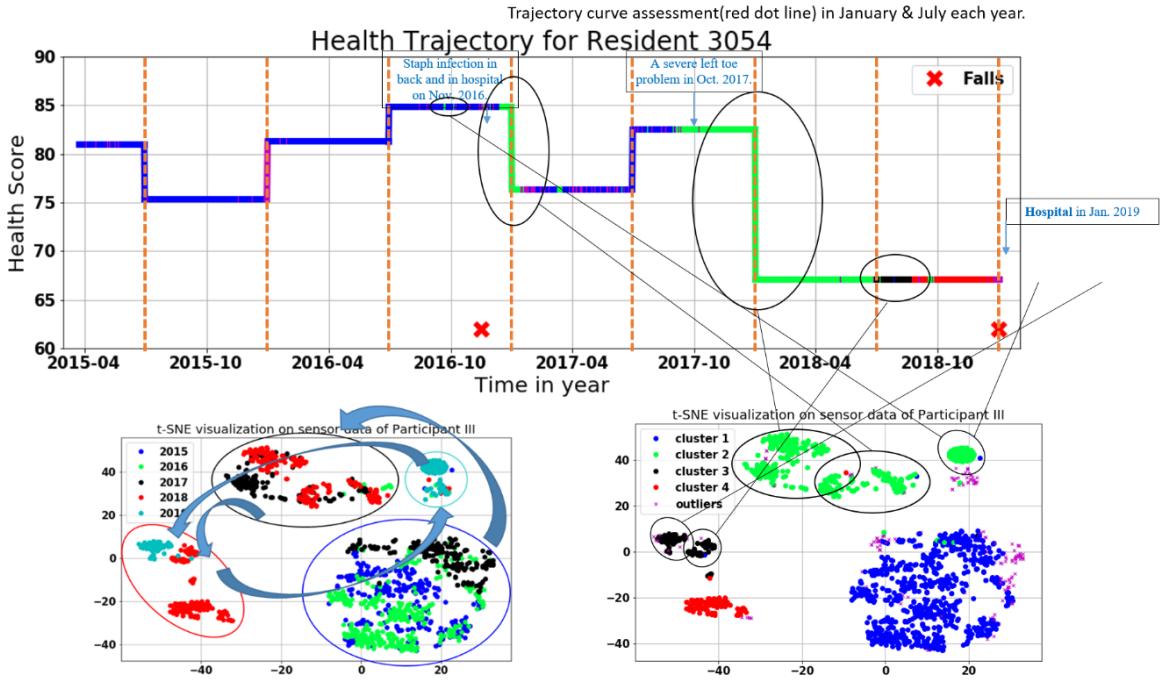


Figure 45. Functional health trajectory with SPGMM clustering result

Figure 45 shows the functional health trajectory with SPGMM clustering results for Resident III. As we can see at the top of Figure 45, the resident had different functional health scores in his different life stages. The colors in the functional health score and the clustering result of SPGMM (the bottom right figure in Figure 45) are matched so that we can compare health scores and clusters. The two significant drops (2017-01 and 2018-01) all happened in the green cluster period; therefore, clinicians should pay close attention to the resident if the resident's pattern falls into the green cluster.

The functional health trajectory with warnings generated by SPGMM is shown in Figure 46.

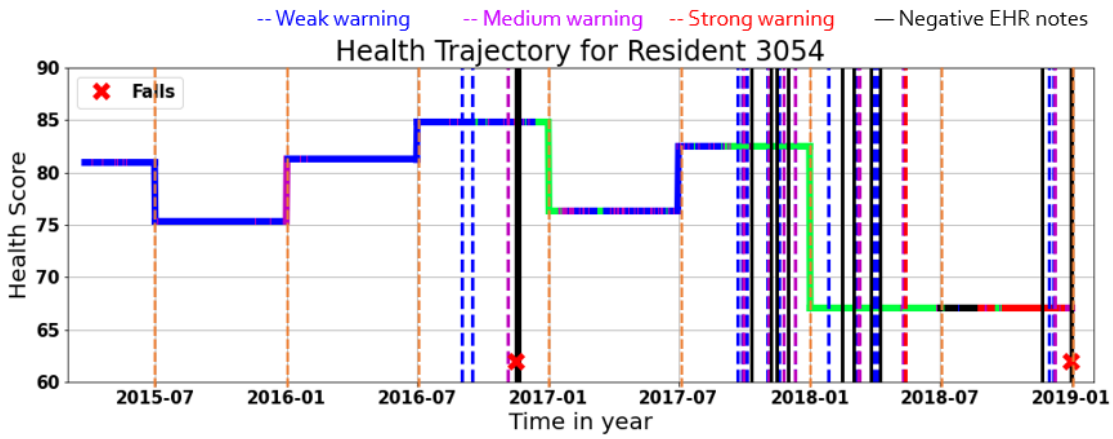


Figure 46. Functional health score with warnings generated by SPGMM

As we can see in Figure 46, many warnings are generated before the negative EHR notes are completed, and also before the functional health score drops. However, the geriatric assessment is performed every six months (January and July each year). The real health drop could happen between two assessments when the negative EHR notes are documented. We might need to shorten the assessment gap in order to get a more accurate estimation of residents' real health drop time. The functional health score provides a numeric perspective to understanding the clusters generated by SPGMM.

CHAPTER 9. CONCLUSIONS AND FUTURE WORK

9.1. Conclusions

With the ability to learn data distribution patterns from streaming data, both SPGMM and StreamSoNG are useful approaches to extract standard patterns from massive streaming data. The SPGMM algorithm with trajectory analysis is suitable to detect very early signs of behavior changes in a streaming fashion, and is a powerful and promising tool to help elders maintain an independent and healthy lifestyle. Unlike our previous one-dimensional algorithm, SPGMM runs on high-dimensional data that is a collection of motion, bed, and depth in-home sensor data over potentially many years. The algorithm first utilizes a period of time with different types of sensors that are tailored to each older adult to initialize the model, then incrementally updates the model and analyzes the trajectory of data streams to detect the early signs of potential health changes. This is our first time to incorporate three types of sensors (motion, bed, and depth) together into the new, multi-dimensional early health change detection algorithm. SPGMM can learn the regular resident pattern and then track the trends in data streams while our previous one-dimensional algorithm only examines a small window size of sensor data in one dimension that "forgets" patterns beyond that window.

SPGMM and StreamSoNG approaches are first validated on synthetic and real-world data. Then four case studies of residents in living in an aging in place facility, TigerPlace, are explored using the SPGMM with trajectory analysis. While the analysis takes place in high dimensional feature space, both the EHR derived ground truth, and the evolving clusters are visualized with the help of two dimension-reduction algorithms, PCA and t-SNE in a retrospective fashion. Using these retrospective case studies, we have shown that

the trajectories in data streams track early signs of pattern changes. The SPGMM with trajectory analysis is tested on sensor data from four residents and it generated early warnings, in some cases, earlier warnings before the patterns in data streams signaled an alert condition using the 1-D algorithm. The generated warnings have been evaluated to be relevant to residents' health events when combining the EHR notes and nursing assessments. On the other side, StreamSoNG provides a different possibility of learning data distributions from streaming data. SPGMM can detect clusters in Gaussian shape while StreamSoNG can discover any shape of clusters, so StreamSoNG has more flexibilities to learn data distributions from streaming data.

Both SPGMM and StreamSoNG cannot provide explainability to support the prediction they made. Four algorithm explainability approaches are discussed to open the "black box" of machine learning models and try to interpret the prediction of how the decision was reached so that the algorithm can not only make accurate predictions but also earn people's trust.

9.2. Future directions

Although many warnings generated by SPGMM reflect the residents' behavior pattern changes, they are not always due to adverse health changes and they indicate the behavior changes from one pattern to another. For example, the resident's motion sensor data can be influenced by a visitor so his/her motion activity increases due to the addition of a visitor's motion activity. Visitor detection at home is a future research topic and we would like to separate the resident's motion activity and visitor's motion activity. Another future direction is to study the behavior change of one specific disease such as pneumonia and urinary tract infection (UTI) to investigate the common behavior of one specific disease.

The SPGMM algorithm works well on data streams, but it needs some "training data" to initialize the model. In the future, we will investigate methods to reduce the amount of initialization training data but keep the same accuracy. Transfer learning that moves model parameters among residents is a promising approach [90]. Furthermore, the current model used for the CLAMP toolkit is the conditional random field (CRF) model, we can further improve the CLAMP performance using some recent state-of-the-art NLP models like clinicalBERT [91].

9.3. Publications

[1] **W. Wu**, J. M. Keller, M. Skubic, M. Popescu, and K. R. Lane, "Early Detection of Health Changes in the Elderly Using In-Home Multi-Sensor Data Streams," ACM Transactions on Computing for Healthcare, vol. 2, no. 3, pp. 1–23, 2021.

[2] **W. Wu**, J. M. Keller, J. Dale, and J. C. Bezdek, "StreamSoNG: A Soft Streaming Classification Approach," IEEE Transactions on Emerging Topics in Computational Intelligence, pp. 1–10, 2021.

[3] **W. Wu**, and J. M. Keller, "Sequential Possibilistic Local Information One-Means Clustering for Image Segmentation," In 2020 IEEE World Congress on Computational Intelligence. **(Best Paper Award Finalist)**

[4] B. Ruprecht, **W. Wu**, M. A. Islam, D. Anderson, J. Keller, G. Scott, C. Davis, F. Petry, P. Elmore, K. Nock, and E. Gilmour, "Possibilistic clustering enabled Neuro Fuzzy Logic," In 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).

[5] **W. Wu**, J. M. Keller, M. Popescu, and M. Skubic, "Data Stream Trajectory Analysis Using Sequential Possibilistic Gaussian Mixture Model," In 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).

[6] **W. Wu**, J. M. Keller, and T. A. Runkler, "Sequential Possibilistic One-Means Clustering with Dynamic Eta," In 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).

REFERENCES

[1] J. S. Passel and D. D'Vera Cohn, *US population projections, 2005-2050*. Pew Research Center Washington, DC, 2008.

[2] J. S. Goodwin, B. Howrey, D. D. Zhang, and Y.-F. Kuo, "Risk of continued institutionalization after hospitalization in older adults," *Journals Gerontol. Ser. A Biomed. Sci. Med. Sci.*, vol. 66, no. 12, pp. 1321–1327, 2011.

[3] M. J. Rantz *et al.*, "Improving nurse care coordination with technology," *CIN Comput. Informatics, Nurs.*, vol. 28, no. 6, pp. 325–332, 2010.

[4] K. Black, "Health and aging-in-place: Implications for community practice," *J. Community Pract.*, vol. 16, no. 1, pp. 79–95, 2008.

[5] P. N. Dawadi, D. J. Cook, and M. Schmitter-Edgecombe, "Automated cognitive health assessment using smart home monitoring of complex tasks," *IEEE Trans. Syst. man, Cybern. Syst.*, vol. 43, no. 6, pp. 1302–1313, 2013.

[6] S. Teipel *et al.*, "Use of nonintrusive sensor-based information and communication technology for real-world evidence for clinical trials in dementia," *Alzheimer's Dement.*, vol. 14, no. 9, pp. 1216–1231, 2018.

[7] M. Rantz *et al.*, "Randomized trial of intelligent sensor system for early illness alerts in senior housing," *J. Am. Med. Dir. Assoc.*, vol. 18, no. 10, pp. 860–870, 2017.

[8] M. Rantz *et al.*, "Enhanced registered nurse care coordination with sensor technology: Impact on length of stay and cost in aging in place housing," *Nurs. Outlook*, vol. 63, no. 6, pp. 650–655, 2015.

[9] M. Skubic, R. D. Guevara, and M. Rantz, "Automated health alerts using in-home sensor data for embedded health assessment," *IEEE J. Transl. Eng. Heal. Med.*, vol. 3, pp. 1–11, 2015.

[10] M. J. Rantz *et al.*, "A technology and nursing collaboration to help older adults age in place," *Nurs. Outlook*, vol. 53, no. 1, pp. 40–45, 2005.

[11] A. K. Mishra *et al.*, "Tracking personalized functional health in older adults using geriatric assessments," *BMC Med. Inform. Decis. Mak.*, vol. 20, no. 1, pp. 1–10, 2020.

[12] S. Wang, "'Change detection for eldercare using passive sensing," *Univ. Missouri-Columbia*, 2011.

[13] I. J. Sledge, J. M. Keller, T. C. Havens, G. L. Alexander, and M. Skubic, "Temporal Activity Analysis.," in *AAAI Fall Symposium: AI in Eldercare: New Solutions to Old Problems*, 2008, pp. 100–108.

[14] S. Aminikhanghahi, T. Wang, and D. J. Cook, "Real-time change point detection

with application to smart home time series data," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 5, pp. 1010–1023, 2018.

[15] E. E. Stone and M. Skubic, "Testing real-time in-home fall alerts with embedded depth video hyperlink," in *International Conference on Smart Homes and Health Telematics*, 2014, pp. 41–48.

[16] E. E. Stone and M. Skubic, "Fall detection in homes of older adults using the Microsoft Kinect," *IEEE J. Biomed. Heal. informatics*, vol. 19, no. 1, pp. 290–301, 2014.

[17] W. Wu, J. M. Keller, M. Skubic, and M. Popescu, "Data Stream Trajectory Analysis Using Sequential Possibilistic Gaussian Mixture Model," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2019, pp. 1–7.

[18] W. Wu, J. M. Keller, J. Dale, and J. C. Bezdek, "StreamSoNG: A Soft Streaming Classification Approach," *arXiv Prepr. arXiv2010.00635*, 2020.

[19] M. Skubic, G. Alexander, M. Popescu, M. Rantz, and J. Keller, "A smart home application to eldercare: Current status and lessons learned," *Technol. Heal. Care*, vol. 17, no. 3, pp. 183–201, 2009.

[20] M. Popescu and A. Mahnot, "Early illness recognition using in-home monitoring sensors and multiple instance learning," *Methods Inf. Med.*, vol. 51, no. 04, pp. 359–367, 2012.

[21] D. Heise and M. Skubic, "Monitoring pulse and respiration with a non-invasive hydraulic bed sensor," in *2010 annual international conference of the IEEE engineering in medicine and biology*, 2010, pp. 2119–2123.

[22] L. Rosales, B. Y. Su, M. Skubic, and K. C. Ho, "Heart rate monitoring using hydraulic bed sensor ballistocardiogram 1," *J. Ambient Intell. Smart Environ.*, vol. 9, no. 2, pp. 193–207, 2017.

[23] E. E. Stone and M. Skubic, "Unobtrusive, continuous, in-home gait measurement using the Microsoft Kinect," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 10, pp. 2925–2932, 2013.

[24] B. G. Celler *et al.*, "Remote monitoring of health status of the elderly at home. A multidisciplinary project on aging at the University of New South Wales," *Int. J. Biomed. Comput.*, vol. 40, no. 2, pp. 147–155, 1995.

[25] A. P. Glascock and D. M. Kutzik, "Behavioral telemedicine: A new approach to the continuous nonintrusive monitoring of activities of daily living," *Telemed. J.*, vol. 6, no. 1, pp. 33–44, 2000.

[26] D. L. Learned, "Independent LifeStyle Assistant^{TM}(ILSA)," 2004.

[27] N. M. Barnes, N. H. Edwards, D. A. D. Rose, and P. Garner, "Lifestyle monitoring-technology for supported independence," *Comput. Control Eng. J.*, vol. 9, no. 4, pp. 169–174, 1998.

[28] B. A. Majeed and S. J. Brown, “Developing a well-being monitoring system—Modeling and data analysis techniques,” *Appl. Soft Comput.*, vol. 6, no. 4, pp. 384–393, 2006.

[29] M. P. Rajasekaran, S. Radhakrishnan, and P. Subbaraj, “Elderly patient monitoring system using a wireless sensor network,” *Telemed. e-Health*, vol. 15, no. 1, pp. 73–79, 2009.

[30] I. Korhonen, J. Pärkkä, and M. van Gils, “Health monitoring in the home of the future. Infrastructure and usage models for wearable sensors,” *IEEE Eng Med Biol*.

[31] A. A. Aramendi, A. Weakley, A. A. Goenaga, M. Schmitter-Edgecombe, and D. J. Cook, “Automatic assessment of functional health decline in older adults based on smart home data,” *J. Biomed. Inform.*, vol. 81, pp. 119–130, 2018.

[32] S. S. Intille *et al.*, “Using a live-in laboratory for ubiquitous computing research,” in *International Conference on Pervasive Computing*, 2006, pp. 349–365.

[33] C. D. Kidd *et al.*, “The aware home: A living laboratory for ubiquitous computing research,” in *International Workshop on Cooperative Buildings*, 1999, pp. 191–198.

[34] J. C. Bezdek, R. Ehrlich, and W. Full, “FCM: The fuzzy c-means clustering algorithm,” *Comput. Geosci.*, vol. 10, no. 2–3, pp. 191–203, 1984.

[35] R. Krishnapuram and J. M. Keller, “A possibilistic approach to clustering,” *IEEE Trans. fuzzy Syst.*, vol. 1, no. 2, pp. 98–110, 1993.

[36] T. A. Runkler and J. M. Keller, “Sequential possibilistic one-means clustering,” in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017, pp. 1–6.

[37] T. Runkler and J. Keller, “Sequential Possibilistic One–Means Clustering With Variable Eta,” in *Proceedings, 27th Workshop on Computational Intelligence*, 2017, pp. 103–116.

[38] W. Wu, J. M. Keller, and T. A. Runkler, “Sequential possibilistic one-means clustering with dynamic eta,” in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2018, pp. 1–8.

[39] M.-S. Yang and C.-Y. Lai, “A robust automatic merging possibilistic clustering method,” *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 1, pp. 26–41, 2010.

[40] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, “A possibilistic fuzzy c-means clustering algorithm,” *IEEE Trans. fuzzy Syst.*, vol. 13, no. 4, pp. 517–530, 2005.

[41] C. C. Aggarwal, *Data streams: models and algorithms*, vol. 31. Springer Science & Business Media, 2007.

[42] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, “On demand classification of data streams,” in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 503–508.

[43] P. Domingos and G. Hulten, “Mining high-speed data streams,” in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000, pp. 71–80.

[44] F. Ferrer-Troyano, J. S. Aguilar-Ruiz, and J. C. Riquelme, “Discovering decision rules from numerical data streams,” in *Proceedings of the 2004 ACM symposium on Applied computing*, 2004, pp. 649–653.

[45] G. Hulten, L. Spencer, and P. Domingos, “Mining time-changing data streams,” in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 97–106.

[46] Y.-N. Law and C. Zaniolo, “An adaptive nearest neighbor classification algorithm for data streams,” in *European Conference on Principles of Data Mining and Knowledge Discovery*, 2005, pp. 108–120.

[47] M. Carnein and H. Trautmann, “Optimizing data stream representation: An extensive survey on stream clustering algorithms,” *Bus. Inf. Syst. Eng.*, vol. 61, no. 3, pp. 277–297, 2019.

[48] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. de Carvalho, and J. Gama, “Data stream clustering: A survey,” *ACM Comput. Surv.*, vol. 46, no. 1, pp. 1–31, 2013.

[49] O. A. Ibrahim, J. Shao, J. M. Keller, and M. Popescu, “A temporal analysis system for early detection of health changes,” in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2016, pp. 186–193.

[50] O. A. Ibrahim, Y. Du, and J. Keller, “Robust on-line streaming clustering,” in *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2018, pp. 467–478.

[51] M. Moshtaghi, C. Leckie, and J. C. Bezdek, “Online clustering of multivariate time-series,” in *Proceedings of the 2016 SIAM international conference on data mining*, 2016, pp. 360–368.

[52] N. Mozafari, S. Hashemi, and A. Hamzeh, “A statistical approach for clustering in streaming data.,” *Artif. Intell. Res.*, vol. 3, no. 1, pp. 38–45, 2014.

[53] F. Cao, M. Estert, W. Qian, and A. Zhou, “Density-based clustering over an evolving data stream with noise,” in *Proceedings of the 2006 SIAM international conference on data mining*, 2006, pp. 328–339.

[54] C. C. Aggarwal, S. Y. Philip, J. Han, and J. Wang, “A framework for clustering evolving data streams,” in *Proceedings 2003 VLDB conference*, 2003, pp. 81–92.

[55] S. Guha, N. Mishra, R. Motwani, and L. o’Callaghan, “Clustering data streams,” in *Proceedings 41st Annual Symposium on Foundations of Computer Science*, 2000, pp. 359–366.

[56] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: an efficient data clustering method for very large databases,” *ACM sigmod Rec.*, vol. 25, no. 2, pp. 103–114,

1996.

[57] M. R. Ackermann, M. Märtens, C. Raupach, K. Swierkot, C. Lammersen, and C. Sohler, “StreamKM++ A clustering algorithm for data streams,” *J. Exp. Algorithmics*, vol. 17, pp. 1–2, 2012.

[58] S. Blažič and I. Škrjanc, “Incremental Fuzzy C-Regression Clustering From Streaming Data for Local-Model-Network Identification,” *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 4, pp. 758–767, 2019.

[59] P. P. Angelov and X. Zhou, “Evolving fuzzy-rule-based classifiers from data streams,” *Ieee Trans. fuzzy Syst.*, vol. 16, no. 6, pp. 1462–1475, 2008.

[60] O. A. Ibrahim, J. M. Keller, and J. C. Bezdek, “Evaluating evolving structure in streaming data with modified Dunn’s indices,” *IEEE Trans. Emerg. Top. Comput. Intell.*, 2019.

[61] O. A. Ibrahim, Y. Wang, and J. M. Keller, “Analysis of incremental cluster validity for big data applications,” *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, vol. 26, no. Suppl. 2, pp. 47–62, 2018.

[62] O. A. Ibrahim, J. M. Keller, and M. Popescu, “A New Incremental Cluster Validity Index for Streaming Clustering Analysis,” in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2019, pp. 1–8.

[63] O. A. Ibrahim, J. M. Keller, and J. C. Bezdek, “Analysis of streaming clustering using an incremental validity index,” in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2018, pp. 1–8.

[64] M. Moshtaghi, J. C. Bezdek, S. M. Erfani, C. Leckie, and J. Bailey, “Online cluster validity indices for performance monitoring of streaming data clustering,” *Int. J. Intell. Syst.*, vol. 34, no. 4, pp. 541–563, 2019.

[65] A. Galusha, J. Dale, J. M. Keller, and A. Zare, “Deep convolutional neural network target classification for underwater synthetic aperture sonar imagery,” in *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIV*, 2019, vol. 11012, p. 1101205.

[66] T. Martinetz and K. Schulten, “A" neural-gas" network learns topologies,” 1991.

[67] H. Frigui and P. Gader, “Detection and discrimination of land mines in ground-penetrating radar based on edge histogram descriptors and a possibilistic $ k $-nearest neighbor classifier,” *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 1, pp. 185–199, 2008.

[68] C. M. Biship, “Pattern recognition and machine learning (information science and statistics).” Springer, 2007.

[69] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, 1999, vol. 2, pp. 246–252.

[70] T. Kohonen, “The self-organizing map,” *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[71] J. M. Keller, M. R. Gray, and J. A. Givens, “A fuzzy k-nearest neighbor algorithm,” *IEEE Trans. Syst. Man. Cybern.*, no. 4, pp. 580–585, 1985.

[72] “LG weather dataset,” [Online]. Available: https://lcav.epfl.ch/page-145180-en.htm.

[73] N. S. Altman, “An introduction to kernel and nearest-neighbor nonparametric regression,” *Am. Stat.*, vol. 46, no. 3, pp. 175–185, 1992.

[74] H. M. Gomes *et al.*, “Adaptive random forests for evolving data stream classification,” *Mach. Learn.*, vol. 106, no. 9, pp. 1469–1495, 2017.

[75] P. Kosina and J. Gama, “Very fast decision rules for classification in data streams,” *Data Min. Knowl. Discov.*, vol. 29, no. 1, pp. 168–202, 2015.

[76] Y. Xu, H. Ji, and C. Fermüller, “Viewpoint invariant texture description using fractal analysis,” *Int. J. Comput. Vis.*, vol. 83, no. 1, pp. 85–100, 2009.

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

[78] I. T. Jolliffe and J. Cadima, “Principal component analysis: a review and recent developments,” *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 374, no. 2065, p. 20150202, 2016.

[79] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[80] Z. Hajihashemi and M. Popescu, “An Early illness recognition framework using a temporal smith waterman algorithm and NLP,” in *AMIA Annual Symposium Proceedings*, 2013, vol. 2013, p. 548.

[81] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation-based anomaly detection,” *ACM Trans. Knowl. Discov. from Data*, vol. 6, no. 1, pp. 1–39, 2012.

[82] E. Soysal *et al.*, “CLAMP–a toolkit for efficiently building customized clinical natural language processing pipelines,” *J. Am. Med. Informatics Assoc.*, vol. 25, no. 3, pp. 331–336, 2018.

[83] Ö. Uzuner, I. Solti, and E. Cadag, “Extracting medication information from clinical text,” *J. Am. Med. Informatics Assoc.*, vol. 17, no. 5, pp. 514–518, 2010.

[84] Y. Zhang *et al.*, “UTH_CCB: a report for semeval 2014–task 7 analysis of clinical text,” in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 802–806.

[85] H. M. Wallach, “Conditional random fields: An introduction,” *Tech. Reports*, p. 22, 2004.

[86] M. L. McHugh, “Interrater reliability: the kappa statistic,” *Biochem. medica*, vol. 22, no. 3, pp. 276–282, 2012.

[87] E. Tjoa and C. Guan, “A survey on explainable artificial intelligence (xai): Toward medical xai,” *IEEE Trans. Neural Networks Learn. Syst.*, 2020.

[88] A. Jain, M. Popescu, J. Keller, M. Rantz, and B. Markway, “Linguistic summarization of in-home sensor data,” *J. Biomed. Inform.*, vol. 96, p. 103240, 2019.

[89] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Proceedings of the 31st international conference on neural information processing systems*, 2017, pp. 4768–4777.

[90] R. Gargees, J. Keller, and M. Popescu, “Early illness recognition in older adults using transfer learning,” in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2017, pp. 1012–1016.

[91] K. Huang, J. Altosaar, and R. Ranganath, “Clinicalbert: Modeling clinical notes and predicting hospital readmission,” *arXiv Prepr. arXiv1904.05342*, 2019.

VITA

Wenlong Wu was born in Qingdao, Shandong, China. He received his Bachelor of Science degree in Electrical Engineering and Automation at East China University of Science and Technology in 2016. He received his Master of Science degree in 2018 and Doctor of Philosophy degree in 2022 from the Department of Electrical Engineering and Computer Science at University of Missouri at Columbia. His research interests include computational intelligence, machine learning, and streaming clustering. He was the recipient of the "Outstanding Master Student" award at the Missouri Honor Awards, 2018. He received the "Best Paper Award Finalist" at WCCI 2020 by IEEE CIS.