Project 3:

Health Monitoring for The Elderly

A. Benichou, W. Gharbi, G. Shindel, N. Turney, & T. Vu

CS420: Artificial Intelligence

Lafayette College

October 25, 2017

— Submitted to

Prof. Chun Wai Liew

A. Introduction

American retail pharmacy and healthcare companies, such as Walgreens and CVS Health, have been developing a new technology that can record vital signs from the human body. These companies aim to increase efficiency and quality of healthcare while decreasing the costs. The advance in wearable and wireless sensor technology have made it possible to monitor multiple vital signs of a patient anytime, anywhere [1]. These technologies are an essential aspect of daily monitoring and disease prevention. This watch-like accessory can be strapped on the wrist to collect vitals such as body temperature, heart rate, blood pressure, sugar level, etc. The goal is to deploy this technology to older patients and to utilize these measurements to monitor the health of these patients who are at risk of developing a wide range of ailments as they age. Our team has been contracted by Walgreens to develop a software system that works with this new technology to monitor the health of the clients in order to take preemptive measures in dealing with ailments before they become major health issues.

This technology is not meant to handle sudden health emergencies, such as heart attacks and strokes, but instead is meant to be a helpful mechanism to aid physicians in having a clear insight into their patients' health which better informs the patients and their families. Specifically, physicians would be able to utilize the system for early knowledge discovery using pattern detection which enables them to make accurate and real-time data-driven decisions. This technology should utilize the gathered data about the patient's vitals to answer a series of questions:

- When is the best time to administer medication?
    - Is it more effective to administer the medication in the morning or evening?
- How efficient is the administered medication?
    - How long does the effect of medication persist?
    - Is it more effective to prescribe multiple small doses or one large dose?

- Are medications adversely affecting each other?
- Are there any long-term changes in the patient's health?
    - Is there a trend being detected that signals the early signs of an ailment?
- How is the patient's ability to ingest medication?
    - Do they often forget to take their medication?

Our team has developed a system that collects data from a patient's vitals to use in order to answer these guiding questions and provide the physician with further insight regarding their patients' health.

B. Parameters

I. Input

This technology will be given to a patient by their doctor. Prior to being administered, the watch will be initialized by the doctor. This means that the doctor will input the patient's basic information (weight, height, age), the patients' demographic data (race, ethnicity, religion), the patient's previous medical records (Have they already been diagnosed with a disease?) and the patient's family medical history (Does this disease run in their family?), either manually or via existing documents. This information will enhance the personalization of the system for the patient. Additionally, if the doctor suspects that the patient may have a particular disease or wants to monitor a specific issue, the doctor can input this into the system which tells the system to look out for certain trends. All watches will collect the same vital signs which includes body temperature, heart rate, blood pressure, and sugar levels. These vitals will be collected at the same rate for all patients and will be sent to a cloud platform for further processing.

II. Knowledge

After the doctor has loaded the patient's information into the watch, the system is informed with the patient's personal and medical data. Using the patient's personalized data, the system will be able to enhance its feature detection and extraction techniques. Additionally, the system contains a predefined database which includes common diseases among patients that are at least 50 years old and which vital signs (or combination of vital signs) can be examined to determine if a patient has a particular disease. This set of diseases and vital sign indicators can be assembled by a group of various physicians.

Our team identified the differences of knowledge representation between data mining tasks such as prediction, anomaly detection and diagnosis which are said to be the predominant data mining tasks [3]. Each of these tasks has a different position in relation to the many aspects of wearable sensing. One dimension involves the setting in which monitoring occurs: prediction and anomaly detection occur in a home or a remote location, such as with the patient on their watch-like technology or within a remote cloud platform, while diagnosis occurs in a clinical location where a licensed physician can utilize tools to make informed decisions. Our system is not permitted to directly send the patient a notification regarding their health, for legal reasons the system must send a notification to the doctor who will inform the patient themselves.

III. Output

Overall, the system will output an analysis of the data based on detected trends and patterns. This report will include any detected trends or anomalies within their vitals and information regarding the efficacy of the medication that a patient is taking. Specifically, this report includes the best times to administer medication, the efficacy of medications, the ability of the patient to ingest medications and any long-term changes in their vitals. This report is sent directly to the doctor for them to utilize in their next appointment with the patient. Using this report, the doctor will be able to complete further tests for possible diseases, adjust the prescription of a patient's medication or dismiss leads that the system might have on a patient but that the doctor thinks is incorrect. After the appointment with the patient, the doctor will provide the system feedback based on the report that the system can utilize to enhance their techniques thus producing frequent learning opportunities.

IV. Assumptions

Assumptions reduce the complexity of the problem by reducing use cases, creating a more homogenous problem state. Assumptions about the processes and baseline capabilities of the system allow for the description of the best possible problem state. The assumptions that define our system and problem are as follows:

- The doctor initializes the watch for the patient with their personal and medical history.
- The watch is charged using body heat thus requires no sort of batteries or charging.
- The watch/system is continuously updated by the doctor with medical news regarding the patient, such as recently diagnosed diseased or prescribed medications.
- The patients are at least 50 years of age and have a known medical record.
- All patients have appointments with their doctors at least once a month.
- Data storage and processing are done within safe and trusted environments thus no data breach on the patient's medical records or vital data can occur.

V. Constraints

Constraints, imposed by the physical limitations of the problem, as well as by the software, must be considered to appropriately design a solution. The constraints on our system are as follows:

- The watch exclusively sends vital sign data to the cloud platform, where it is processed by algorithms.
- The collected data and inferred results cannot be directly displayed to or given to the patient. Only the doctor has access to the results.

VI. Use cases

In the Introduction section, we defined our problem as a series of questions that our client (the health provider/doctor) would like to be answered. These questions serve as mere insight that helps doctors make decisions about their patient's medication prescriptions, treatments, and diets. Our software system is in no way a replacement of a doctor's advice and should never be providing expert medical advice, it, however, should be able to put meaningful data into focus and provide the doctor with detailed statistics and information about the patient's vitals, their activities and the results of monitoring their intake of medication. To this end, we chose to employ multiple data-mining techniques (sensor-mining to be specific) including prediction, classification, segmentation, and summarization. We then learn from and analyze the curated data through General Dynamic Linear Models (GDLMs) in order to answer questions such as "Which trends are the most important to present the doctor?". We further divide the questions we are looking to answer into two classes: Questions that strictly relate to the individual (ex. How often a patient forgets to take medication), and questions where a demographic class can have an impact on the answer (ex. The effect of different types of medication on the patient).

To summarize our approach, we identify the main questions our solution system should be able to answer (Section A.):

- When is the best time to administer medication?
    - **Binary Formulation:** Is it more effective to administer the medication in the morning or evening?
    - **Approach:** Data-preprocessing + Demographic Clustering + Health state prediction (Section IV)

- How efficient is the administered medication?
    - How long does the effect of medication persist?
        - **Approach:** Anomaly detection (Section IV.3.a), ex. Effect of medication = Average time period of daily recurring “shocks” at the expected time of medicine intake
    - Which dose should the doctor prescribe?
        - **Binary Formulation:** Is it more effective to prescribe multiple small doses or one large dose based on patient demographics and vitals?
        - **Approach:** Data-preprocessing + Demographic Clustering + Health state prediction (Section IV)
    - Are medications adversely affecting each other?
        - **Approach:** Anomaly detection (Section IV.3.a) + Trend analysis (Section IV.3.b), ex. Successive “anomalies” (medication intakes) with a sudden drop in vitals or a long-term drop (change in trend) caused by medication interaction.
- Are there any long-term changes in the patient's health?
    - **Approach:** Trend detection (Section IV.3.b), ex. Long-term ramp-up in blood sugar => Possible onset of diabetes
- How is the patient's ability to ingest medication?
    - Do they often forget to take their medication?
        - **Approach:** Anomaly detection (Section IV.3.a): Assuming normal behavior is medication intake then forgetting to take medication would be modeled as an anomaly in some of the vital signals.

VII. Evaluation Criteria

Evaluation criteria provide the system and context to evaluation the effectiveness of our system. We consider how a variety of factors, such as the type of users and the type of data, are handled by our approach. With each factor, several relevant questions were composed to thoroughly assess the system as well as its the performance compared with other approaches. Our evaluation criteria are as follows:

- Use cases
  - Can the algorithm handle the different use cases?
  - Are different use cases handled separately or is the system generic and flexible enough to handle multiple cases in the same way?
- Users
  - Does the main algorithm originally target elderly users? If not who is the targeted users?
  - How can the algorithm be adapted for the elderly?
- Input data
  - What physiological signals & vital signs can be monitored?
  - Are the signals collected continuously or periodically?
  - If periodically, how often are they measured?
- Data processing
  - How is noise handled?
- Data analysis
  - How are trends and patterns recognized?
  - How to distinguish between everyday routine, newly developed habits, and a single-time event?
  - How to identify something as significant or meaningful?
  - When to administer blood pressure medication (morning, evening)?
  - How to gather data about possible long-term changes (e.g. early onset of diabetes)?
  - How are insights about the efficacy of medications gathered? E.g.
    - how long does the effect of insulin persist?
    - multiple small doses or a single large daily dose?
    - are the medications adversely affecting each other?

    - How do we monitor the ability of the patient to ingest medications (e.g., how often do they forget to take the medication)?
- Learning
    - How does the algorithm/system learn from doctors' feedback?
- Overall
    - What are the strengths of the system?
    - What are the limitations of the system?

C. Algorithms

I. Previous Attempts

Prior to settling on our final algorithms, our team considered various other types of techniques to solve this problem. This section explores these techniques and the reasons as to why our team decided not to move forward with these techniques in our system.

1. Random Forest Classifier

ViSiBiD [1] is a prognostic model that can accurately identify dangerous clinical events of a patient in advance using knowledge collected from the patterns of multiple vital signs from a large number of similar patients. The data collected is populated in cloud storage and analyzed using the high processing power of cloud computing technology in order to discover useful patterns for future behavior prediction. They continuously accumulated vital signs, such as heart rate, respiratory rate, body temperature, blood pressure and oxygen saturation, in order to monitor any changes in multiple vital signs which indicates symptoms of diseases and can be utilized for early diagnosis. To do this, a data mining algorithm is used in which groups of input are classified into different classes. Data mining algorithms are very effective and accurate if they are allowed to learn on a large amount of training data. ViSiBiD utilized the training data the MIMIC-II database of MIT Physiobank archive which contains records of 23,180 ICU patients. This database consists of physiological signals and vital sign time-series data captured from patient monitors of different ICUs [1]. ViSiBiD uses this database to train a Random Forest Classifier to handle thousands of feature vectors whose nature is a learning method that produces many decision trees. These decision trees classify incoming data of multiple vital signs of a patient with an unknown clinical condition. The continuous data of patients are sent to this classification engine so that continuous classification results are obtained over time. This way, if an abnormal clinical event is predicted by the classifier, a notification is sent to the patient's physician who makes the final decision on what predictive action to take.

The ViSiBiD system collects multiple real-time vitals from a patient's, send this data to the cloud where it identifies different clinical events and extracts features from the data using a Random Forest Classifier. This sort of system which applies data mining algorithms using cloud platforms seemed like the type of architecture we should use for our system. The classifier runs efficiently on large data sets and cloud platforms allow for fast processing capability and scalable learning. However, our system is not provided with any sort of training data. Without a provided set of training data that accurately reflects the same type of data that will be realistically collected by the watch, the accuracy of the classifier diminishes. Additionally, the ViSiBiD system is used to predict severe clinical events, such as heart attacks, which does not align with the goal of our system. Despite ViSiBiD being a fast and accurate system within its domain, it does not share enough similarities with our system that we could easily utilize it.

2. Associative Classifier

Classification using association rule mining is a major predictive analysis technique that aims to discover a small set of rules in the database that form an accurate classifier [5]. Association rules in the form of implications, A → B, are generated from a training set of data with a certain support and confidence threshold, and then these rules are tested and pruned using the remaining dataset. A and B can be any set of attribute values within the database. Example associative classifiers mined from a medical database include implications such as: {(Age, “>62”), (BMI, “45”), (Blood_pressure, “95-135”)} → Heart_Disease [5]. Associative classifiers can also be applied to temporal, fuzzy, and weighted datasets.

Associative classification data mining was not used in our solution for a variety of reasons. Based on the interpretation our team held of the prompt, we chose a solution which associative classifiers did not serve a clear purpose towards. Our solution attempts to answer binary questions and improves using human feedback. Associative classifiers do not directly describe attributes we choose and instead of relying on human feedback, requires large training sets and leftover example data to check itself against. Also, the output implications rely on support and confidence thresholds whose values would be unclear in a highly sensitive field like medicine.

3. ID3 Algorithm

One of the more end-to-end solutions we have looked into included the use of the Iterative Dichotomizer 3 (ID3) algorithm. This algorithm belongs to the class of classification algorithms and internally, it aims to generate a decision tree based on multiple observations for various attributes and their associated outcome. We aimed to use this algorithm to generate learned decision trees that model every binary data insight we are intended to provide to the doctors (we would, for example, obtain a trained decision tree that outputs whether the patient is showing improvement if the medication is taken in the morning vs in the evening). We first considered this algorithm for the fact that decision trees effectively model the way doctors learn to make decisions based on sensor data (given a sensor reading, say blood sugar level, doctors are usually taught thresholds for which the readings are normal /anomalous /fatal). For instance, systems such as the APACHE II Scoring System [13] (Knaus et al) leverage "scores" associated with physiological measurements to obtain a mortality rate of patients admitted to the ICU. However, through our exploration of the algorithm, we determined that the data we are providing it with is simply too large for any decision tree and needed to be heavily pre-processed through a lossy approach. In addition, our input data contained not only sensor readings but also demographic data which means that the algorithm would need to handle data of multiple types, frequencies of observation and significance, all of which are factors that GDLM handles well.

II. Overview of the Final Design

Overall, this system collects a patient's data from the sensors on the watch-like technology, sends this data to a cloud server where algorithms detect trends, patterns and anomalies, and the outcome is sent to the patient's health professional to be assessed. More specifically, the process of data collecting, data processing, and data mining takes place over a certain period of time and within a clinical and nonclinical setting shown in *Figure 1*. Once the physician has initialized the watch with the patient's personal and medical information, it is administered to the patient and it starts collecting data regarding the patient's vital signs. This data is continuously collected over a certain period of time and then sent to a cloud platform. Cloud computing consists of a network of remote servers hosted by the healthcare corporation for large storage and fast processing capability. Within the cloud, the large amount of data generated by the sensors can be processed using appropriate data processing, and mining techniques. After a certain time period of collecting and aggregating data, it goes through a preprocessing stage where data compression techniques produce a vectorized version of the most important features of the data. This vector of data is fed into three different data mining and data analysis techniques which address different questions from the problem statement. The Cluster Data technique uses the vector of data from a patient to assign the patient to clusters. The Predicting Health State Technique utilizes the vector of data from a patient, along with their assigned clusters, to compare the individual vitals to the average vitals within their various cultures to address their health. The Trend Analysis and Anomaly Detection technique utilized the vector of data from a patient to address medicinal questions and detect unusual trends. Prior to the next doctor’s appointment, these techniques formulate a report which is provided to the patient's healthcare provider to aid them in diagnosing and monitoring their health and optimizing their medical prescriptions. The healthcare provider provides feedback for this report which the system uses to learn and adapt.

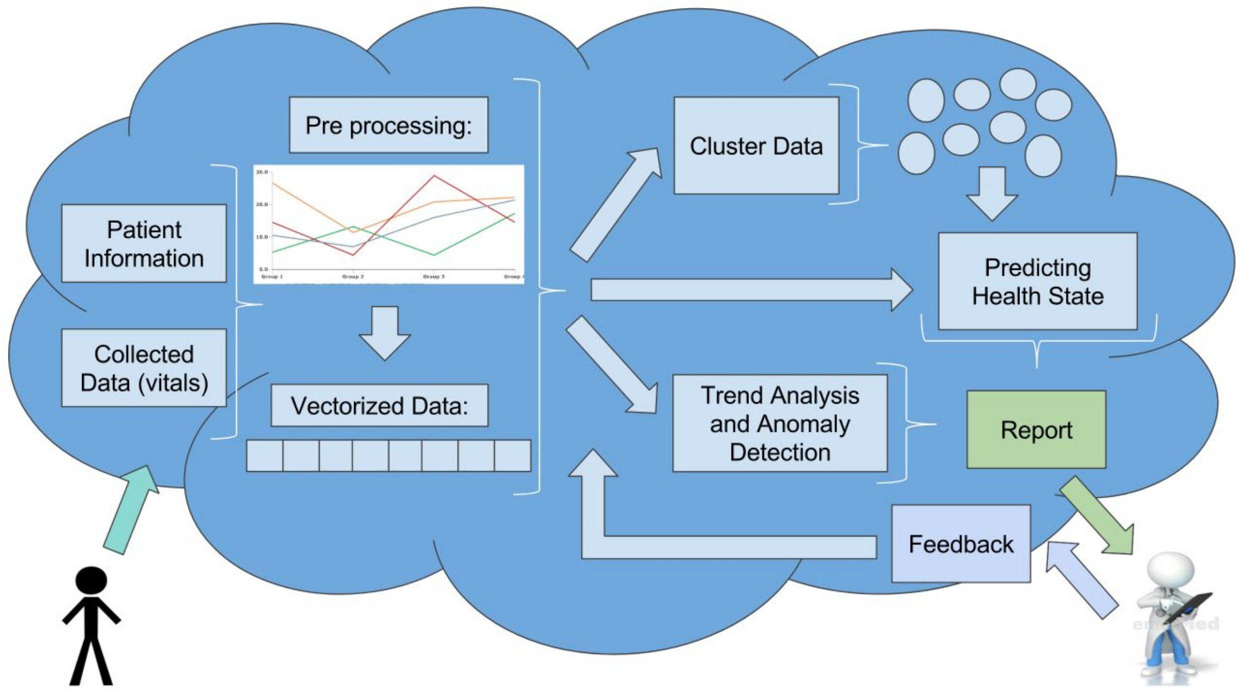

Figure 1: High-level overview of the system.

III. Preprocessing and Knowledge Discovery

Before data pertaining to a patient can be exploited for decision-making processes, it must be simplified through feature extraction to make computation on these massive datasets feasible.

1. Data Collection

Data collected on a patient starts with user information (or metadata) like ethnicity, gender, age, etc. This information is stored within the user's file at an external database. The wearable medical device itself may collect data on any non-invasively measurable and numerically quantifiable vital. Examples include temperature, blood pressure, heart rate, and blood glucose levels. Vital signs are measured at an, essentially, continuous rate. This is done to maximize precision for input into the feature extraction algorithm. An example storage requirement for this method: assuming a 2-byte integer is used to represent a vital sample, with 5 vitals, at a tenth of a second sample rate; over one day, under 8.64 MB of integers would be stored.

2. Feature Extraction

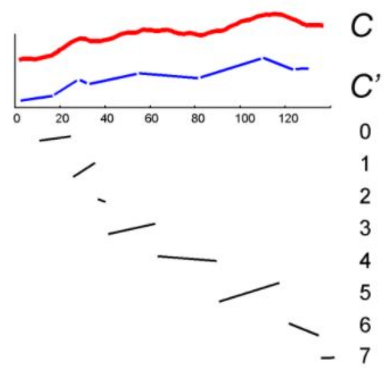

Figure 2: A visualization of the PLA algorithm using K = 7

On the completion of a day, the continuous samplings of each vital are given as input to a piecewise linear approximation (PLA) algorithm. Piecewise linear approximation is ‘described as a way of feature extraction, data compaction, and noise filtering of boundaries of regions of waveforms’ [7]. Example output of PLA can be seen in Figure 2. PLA simplifies a continuous curve into a variable-sized set of line segments where the error norm is minimized. The remaining component necessary to apply this approximation is the choice of the K value, or the number of line segments used to represent the curve [6]. In our application, the K value should relate to the length of time the dataset encompasses, a day, and the variability of the signals, which may be abstracted into the rate at which an individual changes activity. Between fifty and one hundred line segments would effectively represent the changing activity of an individual over a day while still significantly simplifying the data.

3. Data Representation and Organization

a. Wearable

Data approximated into a variable-sized, time-ordered set of lines can be represented as a vector of triples: {Slope, Intercept, Magnitude}. These daily vectors of approximations can, themselves, be stored as a time-ordered vector, increasing the dimensionality of the dataset to two. Upon the next visit to a clinic, this two-dimensional vector will be offloaded to the user's entry in the external database, restoring space for future samples and approximations. Over a period of D days between clinic visits, $K \cdot D$ triples will be stored by the wearable.

b. External Database

Vitals are stored in the external database as a three-dimensional vector of V periods of time between clinic visits, each containing D_V days of data, each of which holds at most K triples representing line segments. Since historic data is naturally less relevant to a patient's state when compared to current data, these sets should also be less significant to a decision-making algorithm. To satisfy this consideration, at the clinic visit binding time period V_X, historical time periods, V_1 to V_{X-1}, should be reapproximated with a lower accuracy. This reduces the significance and size of these historical datasets.

Example Implementation:

Let:

$V_X = Current\ visit\ binding\ time\ period$

$V_1 = First\ visit\ binding\ time\ period$

$K = original\ number\ of\ line\ segments\ used\ to\ approximate\ current\ vitals$

For each period V_Z where $Z = X - Y$ and $Y < K$:

Reapproximate all days of V_Z using $K - Y$ line segments

The resultant multidimensional vector existing on the external database and available for decision-making processes consists of metadata and at most K time periods, each containing D_V days and at most K line segments per day. The vector is represented in *Figure 3*.

Meta data	Current Period V_X: D_X days: K lines: {Slope, Int, Len}	V_{X-1}: D_{X-1} days: K-1 lines	...	V_{X-Y}: D_{X-Y} days: K-Y lines	...	V_{X-K}: D_{X-K} days: K-K lines: 0 Entries

Figure 3: Resultant multidimensional vector

IV. Main System

After being constructed, multidimensional feature vectors are fed into the main system and used for various tasks, including three main subproblems: detecting groups of patients, predicting patients' health states, and detecting trends and anomalies. Several algorithms are integrated into our system to effectively mine a large amount of data, extract useful insight, and better inform the physicians of the patients' health conditions. The detail for each of subproblem is as follows.

1. Demographic Clustering

The first subproblem that our system needs to handle is identifying groups of similar patients. The assumption is that patients with similar backgrounds and health conditions share similar characteristics in terms of their health states. Thus, dividing patients into groups can help better study the common (or uncommon) patterns of each group, more effective than we could have if treating all patients as homogeneous entities. With this approach, more relevant and useful insight could be extracted from the statistics of the group members.

We model this patient grouping problem as a clustering problem: given a set of various patients and their background information, identify groups of similar patients based on their demographic and medical profiles. This problem models a patient as a vector of background features, which is essentially a portion of the output feature vector from data representation process, excluding the real-time vital-reading part. The algorithm we use is called Dimension-based Partitioning and Merging (DPM) [10]. DPM is a fast and scalable clustering algorithm that is suitable for handling large-scale high-dimensional datasets with an unknown number of clusters. An overview of the approach is provided below.

Input/Output for Algorithm 1

- Input:
    - A set of patients with feature vectors consisting of

- medical history (e.g., diseases, medications, food allergies, injuries, etc.)
- demographics (e.g., sex, gender, birth, ethnicity, postal-code, blood-type, etc.)

- Output
  - K clusters of patients with similar backgrounds
    - medical history (e.g., diseases, medications, food allergies, injuries, etc.)
    - demographics (e.g., sex, gender, birth, ethnicity, postal-code, blood-type, etc.)

Algorithm 1: Demographic clustering

1. Step 1: Data partitioning
   - Input: data as
   - $n \cdot d$ matrix
     - n : number of samples or patients
     - d : number of features or dimensions
   - Output: M dense partitions
   - Procedure:
     - $\forall$ dimension D_x , $x \in [1, d]$, generate a histogram H_i
     - Identify m_i dense intervals I_y , $y \in [1, m_i]$, from H_i
     - $\forall\ i, j \in [1, n]$, create a partition if patients p_i, p_j share the intervals I_y in all dimensions D_x
2. Step 2: Noise detecting and filtering
   - Input:
   - M dense partitions
   - Output: M' dense partitions, $M' < M$
   - Procedure:

- $\forall$ partition P_x, $x \in [1, M]$, count z number of dimensions D_x such that $density(D_x) < d_{thres}$
- If $z \geq (d-1)$, mark P_x as noise. Remove P_x

3. Step 3: Merging
   - Input: M' dense partitions
   - Output: K cluster, $K < M'$
   - Procedure:
     - $\forall\ x, y \in [1, M']$, if partitions P_x, P_y are neighbors, merge.
     - $Dist\,(x,y) = max\left[\beta_j\,(x_j - y_j)\right]\ \forall\ j \in [1, d]$

Dimension-based Partitioning and Merging does clustering in 3 steps [10]. First, the algorithm partitions patients into small dense volumes by evaluating the histogram of each dimension. The input dataset for this step is in the format of a $n \cdot d$ matrix, with D is the number of features or dimensions while N is the number of patients. With each dimension D_x with $x \in [1, d]$, DPM constructs a histogram all patients using t arbitrary bins and detect all dense intervals as illustrated in *Figure 4a.* These intervals are then evaluated across all dimensions to construct dense partitions: for all $i, j \in [1, n]$, if patient P_i and patient P_j belong to the same dense intervals in all dimensions, they are considered to be in the same partition.

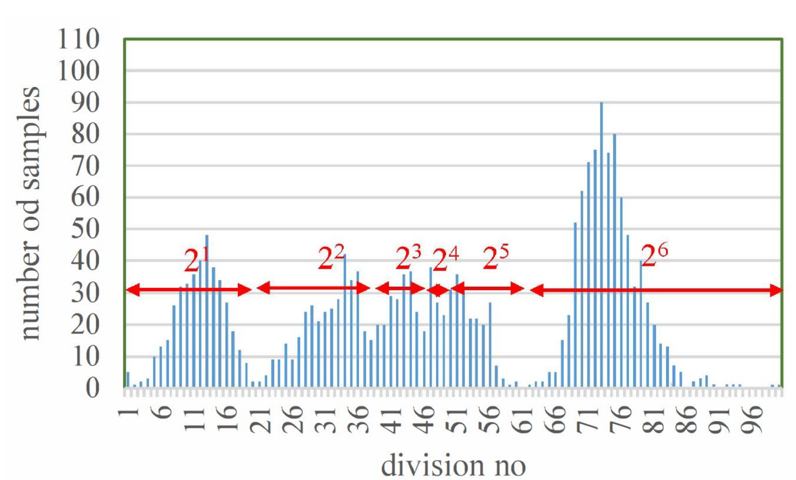

Figure 4a. An illustration of the histogram for data in dimension d_i using $t = 100$ [10]

Second, noise detection and removal is invoked. For each partition, DPM calculates the densities of all its dimensions. The dense partition would be considered as noise if most of its dimensional densities are lower than a threshold n_{thr}. Particularly, given a dense partition X, if $(d-1)$ or more dimensions of X have densities less than n_{thres}, i.e., X is sparse in almost all dimensions, then X is detected as noise and will be removed (*Figure 4b*).

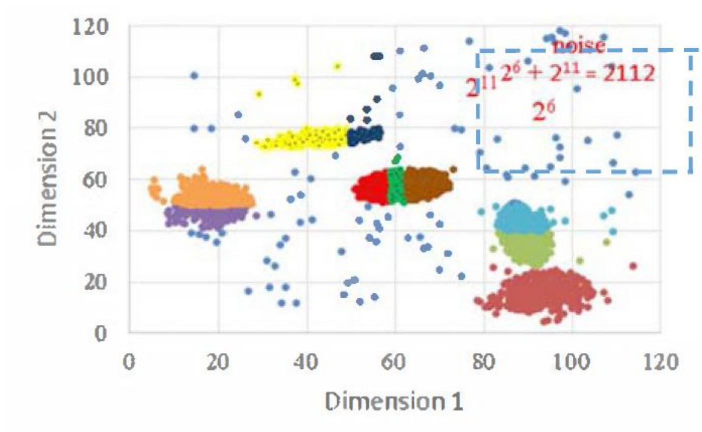

Figure 4b. An illustration for noise detection [10] (top right)

Finally, dense partitions are merged into clusters using nearest test for neighbor groups: if partition p_i passed the nearest test of partition p_j, the two dense partitions would be joined together, creating a new neighbor group. The distance used in the nearest test is defined as the longest distance across all dimension:

$$dist(x, y) = \max_{1 \leq i \leq d} |\beta_i(x_i - y_i)|$$

Thanks to this approach, clusters of patients are automatically formed based on distances between dense partitions, requiring no known number of clusters in advance.

It is shown in [10] that DPM algorithm is significantly faster and more robust compared to other clustering methods such as PROCLUS [11], DOC [12], and DBSCAN [13]. With its divide-and-conquer approach for both data partitioning and dimension processing, DPM not only provides us a way to cluster our large-scale, high-dimensional datasets, but also speed, robustness, and scalability.

2. Predicting health state

Predicting the health state of the user is accomplished utilizing an expectation maximization algorithm applied to a series of general dynamic linear models. Expectation maximization (EM) algorithm allows for the estimation of values in a model dependent upon unobservable variables [4][8]. In our case, these unobservable variables would be information such as lab results and doctor's diagnoses that need to be considered when determining if a trend in vitals is significant to report but cannot be tracked by the sensors in our watch. General Dynamic Linear Models (GDLMs) are a subclass of generalized linear models which utilize time series data but allow for variability in parameters as well as observation windows. In our system, we utilize the system of EM applied to a series of GDLMs to map the tracked vitals with inferred unobservable variables to calculate the probability that a trend is significant to report to the doctor at the next visit by the patient [4].

GDLMs are utilized for their unique properties allowing variable observation windows, unobservable variables, and associations between variables to learn connectivity between variables to determine the importance of trends. GDLMs also allow for a variety of quantitative and qualitative variables as input, and new variables and associations can be added to each model during the learning phase of the system. In the case of tracking the health state of a patient, each vital sign cannot be considered in isolation to determine importance or significance; vital signs inform others as well as the significance of trends associated with each. For example, abnormally high blood pressure in isolation is less significant than abnormally high blood pressure combined with a trend of abnormal blood sugar levels. GDLMs allow for this cross consideration through associations between variables, both observed and unobserved. If variables are not tracked continuously, their values can be inferred from observations to help better inform the system in determining the importance of a trend [8].

We build upon and extrapolate the mechanisms of a previous system utilized to predict patient mortality in ICUs as the framework for our system [4]. The basis of this system was the GDLM mapping a variety of information, such as patient vital information, doctor's and nurse's notes, lab results, and most importantly the previous mortality state of the patient to a probability that the patient is dead since the last observation, with the purpose of predicting negative trends in health for intervention. After the information undergoes a pre-processing step of filtering and smoothing before input into the GDLM, after which EM is applied to the model. Unobserved variables' values are inferred from the previous states and learned values and associations between variables are accounted for in the EM before a probability is generated from the model. Associations and variables in the GDLM are modified based upon doctor's notes, as well as the state of the patient (dead or alive), from the next check in with the patient. The system modifies its state predictions at each iteration based upon the previous states, observable information, and inferred unobservable information [4].

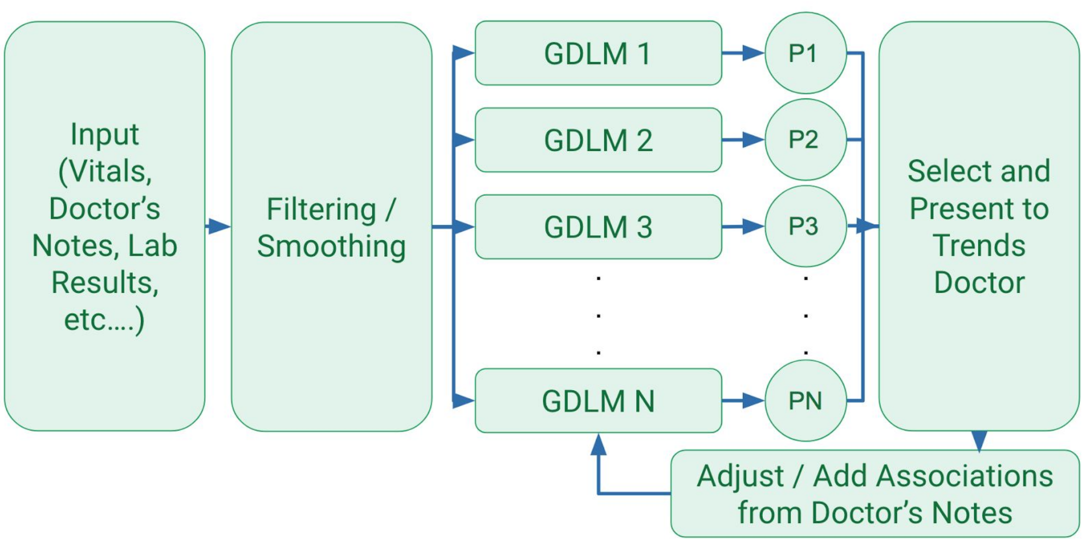

Figure 5. General GDLM System Flow

From this baseline, we extend the number of GDLMs utilized by the overall system to include models for the trends or information that we would report to the doctor. The trends are broken down into multiple models per question if needed, as each requires a binary value from the previous state of the patient. For example, the question of whether it would be better to administer a drug in the morning or the evening would be mapped to two different GDLMs with output probabilities of “the data suggests administration in the morning is impactful on the target vitals” and “the data suggests administration in the evening is more impactful on the target vitals”, respectively. When reported to the doctor, the model with the higher probability of effect is selected to be presented to the doctor. If the doctor agrees with the presented report, then a 1 is assigned as the previous state for the next calculation into the associated GDLM, if the doctor does not agree then a 0 is input into the next calculation as the previous state.

Each GDLM contains observable variables, such as the heart rate, blood pressure, and blood sugar information from the previous observation window, and unobservable variables such as EKG results and lab results. Static variables such as ethnicity and previous diagnosis of disease are also included in the model. The previous output of the model, state of the patient is included as a variable in the model; it is assigned a binary value based upon its significance. This is determined as follows: if the associated trend was reported to the doctor and was determined to be useful or was acted upon the state is set to 1 in the next model, if the information is reported and not deemed important or acted upon the state is set to 0. If the state is not reported but the doctor records he wants the values, the next state is set to a 1 and if the state is not reported and not requested the variable is set to 0 in the next model iteration. Associations between variables denote their importance in determining the significance of a trend.

Vital information in the form of the vector generated in the preprocessing and clustering steps is input into the system in addition to doctor’s notes, lab results, and other variables such as previous diagnosis, ethnicity, and previous state for that model. After smoothing and filtering of the data, each GDLM is passed the input. The EM algorithm is applied to each model, inferring missing unobserved variables based upon previous model states, returning a probability for each model [4][8]. It has been displayed to accurately extrapolate noisy data into predictions and classifications. This probability is the probability that the trend associated with each model is significant to report to the doctor. All probabilities above a value N are selected and reported to the doctor. The notes from the meeting with the doctor are used by the system to refine the associations and association coefficients in each model based upon the doctor's prognosis, possible diagnosis of diseases, as well as their determination of the importance of the information presented in the report [4]. Doctor's notes, diagnosis, and test or lab results are utilized to add new variables to the relevant GDLM if needed. The combination of EM algorithm applied upon a variety of data types allows us to generate a robust and flexible system that expands upon and learns additional associations as data collection continues; the system allows for flexibility in observation window times, as well as extrapolates information further based upon the trends in other tracked vitals, unobserved data, and doctor input.

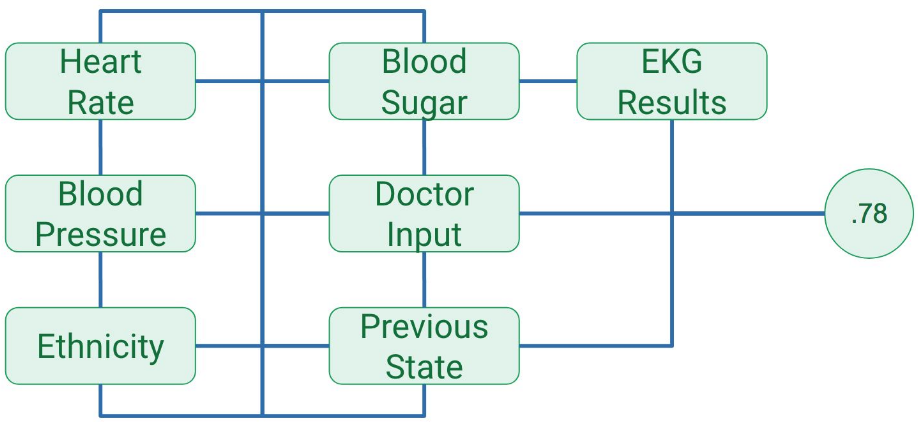

Figure 6. Variable Addition and Association Adjustment to a GDLM in Learning Phase

3. Trend and Anomaly Detection

To answer some of the questions that relate to the single individual, we deemed it impractical and irrelevant to use a learning model that encompasses other patients' information. These questions, traditionally solved by doctors through looking at a patient's vital reports, include:

- Gathering data about possible long-term changes (e.g., early onset of diabetes)
- Gathering data about the efficacy of medications (e.g., how long does the effect of insulin persist, are the medications adversely affecting each other)

- Monitoring the ability of the patient to ingest medications (e.g., how often do they forget to take the medication)

We decided to model these questions as data mining/analysis problems and tried to solve them using the relevant operations we can perform on time series [6]. We particularly found that the *clustering*, *segmentation*, and *anomaly detection* were of utmost relevance to solving these various sub-problems.

Given the nature of our data (time-bound continuous readings of sensor data), we modeled our collected input as multiple time series (where each continuous reading from every sensor is different a time series). We then needed to understand the nature of our data. We determined that the time series we collect are *deterministic* time series augmented with *additive seasonal patterns*. The former term means that we assume that shocks/anomalies revert to the "normal" behavior in the long term (since our shocks/anomalies are most of the time behavioral and do not have permanent/irreversible effects on the patient's health, shocks/anomalies we might observe include forgetting to take medication, fasting, exercising, etc.). The latter term (*additive seasonal patterns*) means that our data contains fluctuations that happen at a regular/predictable rate and that their variations are relatively constant in amplitude (as opposed to the amplitude increasing with every additional fluctuation) [15].

To answer the questions at hand, we identified two types of anomalies we would like to be able to detect and analyze in every sensor reading: *ramps* and *shocks*. Ramps (or trends) are the slow, long-term change in the sensor data (which would help us in predicting early onsets of diseases) while shocks (or instant anomalies) are the sudden, unpredicted fluctuations in the sensor readings (which would help us predict day-to-day outliers such as exercising, forgetting to take medication, starting to take a new medication, etc.).

a. Shock detection

As mentioned above, *shocks* (or instantaneous anomalies) are fluctuations in the sensor data that happen in a short period of time and that are at the same time inconsistent with the regular patterns seen in the time series. Identifying shocks and characterizing them (amplitude/frequency) will help us answer the third and fourth question of our prompt (since both medication intake, medication abuse and medication effects are all reflected in the sensor data as short-term anomalies). To identify shocks, we combine various time series operations as detailed in *Figure 7*.

Algorithm 2: Shock Detection

1. Start by applying *Time-based Segmentation* on the relevant data (X successive readings, X to be determined through experimentation)
    - We use a sliding time window to identify relevant segments while keeping track of their position in the time series

2. Classify the obtained segments into k classes that denote useful patterns that appear in the patient's data using the *k-means clustering algorithm* (explained below). Given that this process is unsupervised, we cannot label the learned patterns, however, we anticipate that we will see different patterns for sleeping, fasting, eating, taking medication, etc.

3. Once a model of recurring patterns is learned, we can perform anomaly detection by fitting the input data to the learned patterns and observing differences (given the fact that the learned patterns denote the average shape of the sensor data, thus any input that does not classify as one of the previously learned patterns is indeed an anomaly).

4. After each iteration, if the observation is not reported as anomalous (through the doctor's report), use it to fine-tune the learned patterns for any new behaviors / habits.

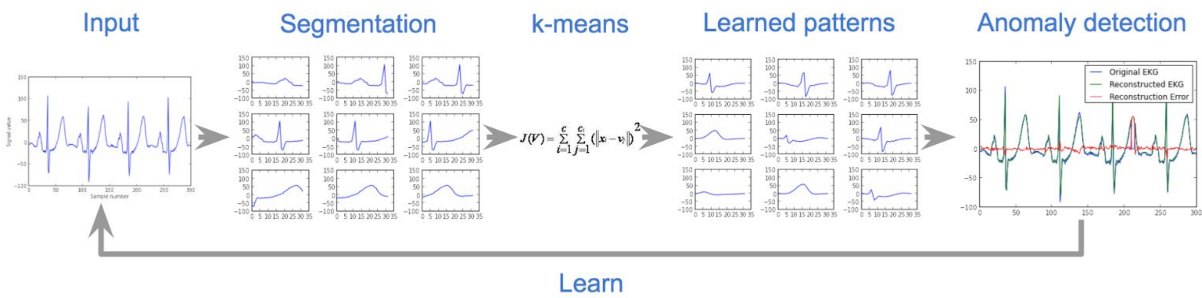

Figure 7. Shock detection algorithm overview

K-Means clustering [16] is a simple algorithm that aims to partition n observations into k "clusters" or classes by using means as a measure of "closeness" to the determined cluster/group. The k-means clustering algorithm is highly suitable for our use case since it makes use of an unsupervised learning process, meaning that data points do not need to be labeled beforehand (which is appropriate since we do not have prior information about the patterns we are going to see in the time series). At the same time, k-means clustering provides us with the centroids of the k clusters (which can be extremely useful for analysis). The algorithm for k-means clustering is straightforward:

Let $S_i(t)$ denote the set of observations associated with cluster i at iteration t. Given a set of initial means (can be randomly chosen from the observations themselves) denoted $m_1(0)$, $m_2(0), \ ..., \ m_k(0)$, the algorithm tries to minimize the distance of each observation to the mean of its assigned cluster using two steps:

1. Assign each observation to the cluster with the "nearest" mean (calculated through a simple Euclidean distance) as follows:

$$S_i(t) \ = \ \{x_p : ||x_p - m_i(t)||^2 \leq ||x_p - m_j(t)||^2 \ \ \forall j, \ 1 \leq j \leq k\}$$

2. Calculate the new means of the centroids using their new associated observations

$$m_i(t+1) \ = \ \frac{1}{|S_i(t)|} \sum_{x_j \in S_i(t)} x_j$$

The algorithm will then converge to a solution and there will be no more significant change in means when clusters are determined [16].

b. Trend Detection

Trend (or ramp) detection is an easier task to perform than shock detection since we are simply looking for an overall increase/decrease in the time series combined with a measure of the long-term rate of change. The algorithm we are using leverages linear regression and time series summarization to fit a line to the data (over time) and obtain a moving slope/intercept indicating the direction and the rate of change of the sensor measurements.

Algorithm 3: Trend Detection

1. Perform dynamic *summarization* of the time series (more recent data points have greater effect)
2. Perform *linear regression* (using the method of least squares) on the time series
3. Obtain a starting value and a slope indicating if the data is increasing/decreasing and at what rate

V. Algorithm evaluation

Criteria	Our solution
Use cases	
Can the algorithm handle different use cases?	- Yes. Refer to Section B. VI. for more details
Are different use cases handled separately or is the system generic/flexible enough to	- The system is designed to answer all the questions that were specified earlier in Section VI. but is also flexible enough to adapt to answer novel binary questions (through training the

handle multiple cases in the same way?	model for good and bad behavior based on the doctor's report).
Users	
Does the main algorithm originally target elderly users? If not who is the targeted users?	- The main algorithm for predicting the health state of the patient originally targeted patients admitted into ICUs.
How can the algorithm be adapted for the elderly?	- Yes, the algorithm could be adapted for the elderly, as ICU patients tend to be a subset of the elderly population, just in more dire health states. Adapting the original algorithm is accomplished by modifying the GDLMs in the system by training on general health improvement/deterioration prediction rather than mortality prediction.
Input data	
What physiological signals & vital signs can be monitored?	- The algorithms are flexible enough to account for a wide range of vital signs, as long as the physical device can collect and vectorization of data is allowed. - Examples of these signals are temperature, blood pressure, heart rate, and blood glucose levels.
Are the signals collected continuously or periodically? If periodically, how often are they measured?	- Vitals are measured at an essentially continuous rate. Although they cannot physically be measured continuously and vitals will not be expected to change within fractions of a second, if a vital is stored as a 2-byte integer, measured every tenth of a second, only 1.7 MB of integers would be generated over a day, and raw data is only stored over single day periods, making high rates of sampling feasible.

Data processing	
How is noise handled?	-Noise generated in raw data from sensor variability will be smoothed and filtered when simplified using Piecewise Linear Approximation into a finite set of line segments [7]. - Clustering algorithm filters noise by evaluating partitions' dimensional densities.
Data analysis	
How are trends and patterns recognized?	- Trends and pattern analysis is detailed in Section IV.3, we use linear regression for trends and k-means clustering for anomalies.
How do we tell when to administer blood pressure medication (morning, evening)?	- Determining when to best to administer medication is accomplished by breaking the problem into two separate GDLMs, one which outputs the probability that the medication is affecting the relevant vitals in the morning, and one which outputs the probability that the medication is affecting the relevant vitals in the evening. When reporting to the doctor, the two probabilities are compared and the larger value is reported to the doctor, i.e. our data suggests administration in the morning seems to have a more pronounced effect on the relevant vitals.
How is data about possible long-term changes (e.g. early onset of diabetes) gathered?	- Long-term changes are inferred from the trend analysis algorithm (with ramp-ups and ramp-downs after a certain threshold being reported to the doctor).
How are data about efficacy of medications gathered?	- Data about the efficacy of medication is tracked by generating the necessary GDLMs for the medication.

How do we monitor the ability of the patient to ingest medications (e.g., how often do they forget to take the medication)?	- The patient's ability to ingest medication is determined through either anomaly detection (assuming taking the medication is the expected behavior) or trend detection (assuming taking the medication would change the trend of the signal reading in direction of the better/normal reading).
Learning	
How does the algorithm/system learn from doctors' feedback?	- The algorithm learns from doctor feedback following the patient's visit. The doctor's notes and feedback are utilized by the system to refine the coefficients of associations between variables in each individual GDLM, add new associations between variables, or add variables to the GDLM as by the adapted system [4]. Further learning occurs on the next calculation of the GDLM, which takes into account the previous state of that model, whether the associated trend was reported to the doctor and if the doctor found that information valuable or useful.
Overall	
What are the strengths of the system?	- The system's strength lies in its ability to infer unobservable information to increase the accuracy of predictions and calculations, as well as its ability to factor in the associations and connectivity of information in determining the importance of a trend. Additional strengths of the system are its ability for variable observation windows, integration of a wide range of information, both quantitative and qualitative, as well as quickly learn from doctor feedback.

What are the limitations of the system?	- GDLM's training relies on doctor's feedback - Since DPM identifies dense intervals from the histogram in each dimension to create dense partitions, those intervals of binary data such as sex might be less useful than those of non-binary data.

VI. Future work

Below is some directions and ideas for follow up works to improve the system.

- In the long run, as we get more and more data, we hope to rely less on doctor feedback. In an ideal solution, the system would be able to independently analyze and extract insight from patients' background data and real-time readings with minimal or no help from the doctor.
- It may be worth comparing DPM – a density-based clustering algorithm – with other types of clustering algorithms such as connectivity-based or centroid-based.
- Investigation of superior methods of factoring temporal significance into our solution, which can accurately classify the importance of historic data, may improve decision probabilities.

D. Conclusion/Ethics

Artificial Intelligence has been regarded as a double-edged sword considering it has the power to greatly aid yet severely hurt people and society. Before implementing and deploying a system, it is imperative to consider the advantages and the disadvantages that the system may bring about. In health monitoring systems, there is a focus on developing intelligent algorithms to perform a variety of tasks such as pattern recognition, anomaly detection and decision-based systems which can handle subject-specific models and personalization. Creating a system for a large retail pharmacy and healthcare company, such as Walgreens, that can utilize the vitals collected from the human body in order to give physicians clear insight into their older patients' health will benefit all parties involved. The utilization of a system that continuously monitors a patient's health will be chiefly beneficial to the patients themselves, who would be able to more easily and effectively receive the preemptive care they need. Individuals will be able to receive personalized care as a result of the system managing their specific individual data. Using this system, physicians will be able to make data-driven decisions. They will be provided with the proper evidence needed to prompt a possible diagnosis and will be able to make real-time decisions when it comes to administering medications. Healthcare providers can develop more insightful diagnosis and treatments, resulting in lower costs and with better outcomes. With this system, medical care will be able to become a 24/7 hour operation that can be done while outside of a clinical setting. Finally, as the data grows and the learning algorithms improve, more patients from a wide range of backgrounds and profiles will be able to benefit from the collective knowledge gained from the data.

On the other hand, there are many aspects of a system like this that frightens many people. This system cannot replace the human interaction people have with their doctors or nurses, who are able to provide a human connection and a sense of care and empathy that a health notification or a watch cannot. While this possibility may occur far ahead in the future, it is possible that advanced systems like these replace the need for licensed medical professionals which would replace vital jobs and increase the unemployment rate. If reliance on healthcare were to switch from humans to algorithms, the quality of the care received would diminish. Additionally, it would cut down other professions related to the medical industry which would be very detrimental to a wide range of people. This system would produce a vast amount of medical data which would need to be stored somewhere safe in order to reduce the risk of data breaches.

Overall, perhaps implementing this system is the natural progression of the medical field and the healthcare industry. This increase in nonclinical environments using vital signs provided by wearable sensors will lead to physicians being able to detect diseases early and provide patients with personalized medical doses. However, the future could bring a time where licensed physicians are no longer needed to diagnose diseases and administer medication. Fortunately, humans are still trusted more than algorithms in many domains such as decision making, handling difficult situations, and bringing a sense of empathy into a clinical setting. Legally speaking, there is an extent as to which software systems and algorithms can be involved in a patient's medical records and health decisions. Despite this, the long years of medical training, the heavy costs of a medical education and the burdensome weight of insurance could lead to advancements in cutting out medical professionals in order to provide patients with the care they need. In conclusion, we believe that our system efficiently provides a helpful mechanism to aid physicians in gaining clear insight into their patients' health.

Team Members

- Agathe Benichou
- Wassim Gharbi
- Greg Shindel
- Nick Turney
- Thanh Vu

References

[1] Forkan, Abdur, Khalil, Ibrahim, Atiquzzaman, Mohammed. *ViSiBiD: A learning model for early discovery and real-time prediction of severe clinical events using Vital Signs as Big Data.* February, 2017. doi:10.1016/j.comnet.2016.12.019.

[2] Rohloff, Kurt, Polyakov, Yuriy. *An end-to-end security architecture to collect, process and share wearable medical data.* In Proceedings of 17th International Conference on E-Health Networking, Applications and Services (HealthCom). October 14-17 2015.

[3] Banaee, Hadi, Ahmed, Mobyen and Loutfi, Amy. *Data Mining for Wearable Sensors in Health Monitoring Systems: A Review of Recent Trends and Challenges*. Center for Applied Autonomous Sensor Systems December 17 2013.

[4] Barajas, Karla L. Caballero, and Ram Akella. *Dynamically Modeling Patient's Health State from Electronic Medical Records.* In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15, 2015, doi:10.1145/2783258.2783289.

[5] Ranganatha, S., et al. *Medical data mining and analysis for heart disease dataset using classification techniques.* National Conference on Challenges in Research & Technology in the Coming Decades (CRT 2013), Sept. 2013. IEEE, Xplore Digital Library, doi:10.1049/cp.2013.2485.

[6] Ralanamahatana, Chotirat Ann, Jessica Lin, Dimitrios Gunopulos, Eamonn Keogh, Michail Vlachos, and Gautam Das. 2005. *Mining Time Series Data.* In *Data Mining and Knowledge Discovery Handbook*, 1069–1103. Springer, Boston, MA. doi:10.1002/9781118445112.stat05517.

[7] T. Pavlidis and S. L. Horowitz. *Segmentation of Plane Curves*. in *IEEE Transactions on Computers*, vol. C-23, no. 8, pp. 860-870, Aug. 1974, doi: 10.1109/T-C.1974.224041

[8] Ghosh, Sujit K., et al. *Generalized Linear Models: a Bayesian Perspective*. Marcel Dekker, 2000.

[9] Subramaniyam N.P., Tronarp F., Särkkä S., Parkkonen L. (2018) *Expectation–maximization algorithm with a nonlinear Kalman smoother for MEG/EEG connectivity estimation.* In: Eskola H., Väisänen O., Viik J., Hyttinen J. (eds) EMBEC & NBC 2017. EMBEC 2017, NBC 2017. IFMBE Proceedings, vol 65. Springer, Singapore

[10] Ghanem, T. F., Elkilani, W. S., Abdelkader, H. M., & Hadhoud, M. M. (2015). *Fast Dimension-based Partitioning and Merging clustering algorithm*. Applied Soft Computing, 36, 143-151. doi:10.1016/j.asoc.2015.05.049

[11] Charu C. Aggarwal , Joel L. Wolf , Philip S. Yu , Cecilia Procopiuc , Jong Soo Park. *Fast algorithms for projected clustering*. ACM SIGMOD Record, v.28 n.2, p.61-72, June 1999.

[12] C. M. Procopiuc, M. Jones, P. K. Agarwal and T. M. Murali. *A Monte Carlo Algorithm for Fast Projective Clustering*. In Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, 2002.

[13] A. Hinneburg, E. Hinneburg and D. A. Keirn. "*An Efficient Approach to Clustering in Large Multimedia Databases with Noise*. In Proceeding of the 1998 International Conference on Knowledge Discovery and Data Mining (KDD '98), New York, 1998.

[14] Knaus, W. A., E. A. Draper, D. P. Wagner, and J. E. Zimmerman. 1985. *APACHE II: A Severity of Disease Classification System*. *Critical Care Medicine* 13 (10): 818–29.

[15] Gould, Phillip G., Anne B. Koehler, J. Keith Ord, Ral Snyder, Rob J. Hyndman, and Farshid Vahid-Araghi. 2008. *Forecasting Time Series with Multiple Seasonal Patterns.* European Journal of Operational Research 191 (1): 207–22.

[16] Hartigan, J. A., and M. A. Wong. 1979. *Algorithm AS 136: A K-Means Clustering Algorithm.* Journal of the Royal Statistical Society. Series C, Applied Statistics 28 (1). [Wiley, Royal Statistical Society]: 100–108.