Quantified Self: Building a Multi-Time Window Analytical

Workflow for Clustering Wearable Data Streams

by

Luke McCully

B.Sc.E. Geomatics Engineering, University of New Brunswick Fredericton, 2018

A Thesis, Dissertation in Partial Fulfillment
of the Requirements for the Degree of

Master of Science in Engineering

in the Graduate Academic Unit of Geodesy and Geomatics Engineering

Supervisor: Monica Wachowicz, PhD, Geodesy and Geomatics Engineering, UNB

Examining Board: Suprio Ray, Department of Computer Science, UNB
Ian Church, PhD, Geodesy and Geomatics Engineering, UNB

This thesis is accepted by the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

January 2021

ABSTRACT

A new research domain known as the Quantified Self has recently emerged and is described as gaining self-knowledge through using technology to acquire spatio-temporal data on our behavior. Wearable technology is widely used in this domain since it generates a large amount of wearable stream data, which contains information on self-monitoring activities and physical health related problems. However, very little is known about which stream clustering algorithms should be used and which time windows can reveal individuals' spatio-temporal patterns that can yield new self-knowledge insights. This thesis proposes an analytical workflow developed to reveal self-quantified patterns that can be used to understand physical activity behavior. It consists of six phases that are devised to support tasks including retrieving, processing, and clustering wearable data streams. The streaming k-means clustering algorithm, based on an online/offline approach using both sliding and damped time window models, is proposed to uncover self-quantified patterns. An intervention experiment with 15 participants is used to gather Fitbit data logs and implement the proposed analytical workflow. The clustering results reveal the impact of a time window model has on exploring the evolution of micro-clusters and the labelling of macro-clusters to accurately explain regular and irregular individual physical behavior.

DEDICATION

This thesis is dedicated to my parents for always being supportive throughout my undergraduate degree to this point in my academic career. While constantly keeping me motivated and eager to continue, they were also always there for me to offer their complete compassion and empathy.

ACKNOWLEDGEMENTS

This research was supported by the NSERC/Cisco Industrial Research Chair [Grant IRCPJ 488403-14]. We would also like to thank the Cisco-Flinders Digital Health Design Lab. in Australia for providing us with the Fitbit data logs.

I would like to specifically acknowledge and express my gratitude toward my supervisor Monica Wachowicz for her guidance and assistance throughout the duration of my time in graduate school. Her expertise in an assortment of areas helped me grow as a student and gave me irreplaceable experience to guide me through my career. I would also like to thank Hung Cao for his eagerness to assist not just myself, but everyone around our lab. I feel privileged to be able to call myself part of the People in Motion Lab as I learned so much in such a short duration of time due to being surrounded by bright-minded students and a mentor with so much to learn from.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Continuously-worn wearable devices are becoming more prevalent in society for quantifying oneself through collecting data for the monitoring of physical health related problems. Examples of such are blood pressure, sugar level, and obesity, which are usually associated to chronic diseases like cardiovascular disease and diabetes, and early detection of neurodegenerative disorders (Jo, Coronel, Coakes, & Mainous III, 2019). There has also been considerable research using clustering algorithms for analyzing wearable device logs since a variety of information about the individuals' activity is reported, including the calories consumed, sleep patterns, steps walked, distance moved, and stairs climbed; despite the fact that there might be a specificity issue with this information related to low accuracy.

Previous research has shown that the k-means algorithm is the most commonly partitioning based approach using historical wearable device logs (Zuraini, Ismail, Hendradi, & Justitia, 2020). However, very few attempts can be found in the literature in clustering wearable data streams (Park, Lee, Han, & Cha, 2019).This is mainly because analyzing them brings along a research challenge in which the rate of the data compiled and stored is not being optimized to each use case, which is where time window models come into effect. When the wearable data streams are continuously brought into the k-means algorithm, it is challenging to retrieve any insight since previous and future data streams are needed to provide context. For example, a user may have a one minute peak in heart rate while not having any steps taken during the one minute interval, but looking

at the previous minute timestamps may provide important context such as the user may have just sprinted during previous minute. This detracts from the possibility that this sudden heart-rate spike could be due to a health issue. This contextual information is what makes the time window models an important factor to take into account in the streaming k-means clustering algorithm when analyzing wearable data streams.

Different time window models can be coupled within the streaming k-means clustering algorithm, including sliding, landmark, damped and pyramidal (Mansalis, Ntoutsi, Pelekis, & Theodoridis, 2018). Each of these models aims to handle the evolution of the distribution of the data streams over time, and as a result, they determine at which time frame the streams are stored and analyzed, and when the previous historical streams are discarded (Carnein & Trautmann, 2019). With regards to wearable devices, this can become an issue since historical data streams might be as important as new incoming data streams. It is paramount to understand what the impact these windows have on generating self-quantified patterns over time.

This thesis proposes an analytical workflow to reveal self-quantified patterns by using a streaming k-means clustering algorithm based on finding online micro-clusters from the wearable data streams and offline macro-clusters from re-clustering these micro-clusters. The sliding time window model is used to understand micro-cluster evolution, which plays an important role in distinguishing actual novel self-quantified patterns from possible existing outliers. Meanwhile, the damped time window model draws on micro-cluster scalability, defined here as the maximum number of current and historical data streams which guarantees context consistency that is needed to compute micro-clusters. Consequently, self-quantified patterns are inferred from the *k* macro-clusters that are

computed by re-clustering the set of k_0 micro-clusters using a particular time window model. The labelling of these k macro-clusters is a process aimed to reveal changes in physical activity behavior, targeting on individuals rather than their physical and social environments.

1.1 Research Objectives

The main research goal to be achieved is to understand the impact of time window models on an analytical workflow using a streaming k-means clustering algorithm based on finding online micro-clusters from the wearable data streams and offline macro-clusters from re-clustering these micro-clusters. Both micro- and macro-clusters are aimed to reveal self-quantified patterns from wearable data streams.

The measurable research objectives can be described as follows:

- Build an analytical workflow to achieve adequate storage capacity in the online component, so the wearable data streams results remain constant in the streaming k-means clustering algorithm, as well as immediately storing the processed micro-clusters for further re-clustering.
- Implement the proposed analytical workflow using two time window models: the sliding and damped window models despite the complexity of integrating two different time window models.
- Evaluate the clustering results for finding self-quantified patterns from wearable stream data.

1.2 Scientific Contributions

The scientific contributions of this thesis can be described as follows:

- A new multi-window analytical workflow for streaming k-means clustering is proposed since previous research work has neglected the role of time window models in cluster evolution and cluster scalability.
- The damped time window model has never been used for clustering wearable data streams before.
- The empirical results of this research work are expected to advance the understanding of the Geospatial Data Science community about the impact of a time window modelling might have in finding self-quantified patterns from wearable stream data.

1.3 Organization of Thesis

The thesis is organized as follows.

Chapter 2 presents the research background on wearable technology, the subject of quantified yourself, and an overview of clustering approaches.

Chapter 3 summarizes previous research work in this research domain.

Chapter 4 introduces the proposed multi-window analytical workflow.

Chapter 5 provides a summary of its implementation.

Chapter 6 includes a discussion of the clustering results.

Finally, Chapter 7 concludes and outlines future research work.

Chapter 2: Background

To provide further context, this Chapter 2 describes the wearable device used for this research and its means for capturing data. Along with the device, this chapter also provides the definition of "Quantifying Yourself" and explains how clustering analysis is used to tie the two together.

2.1 Wearable Technology

Fitbit is a type of wearable devices that has become increasingly popular throughout the world. In this shared space by nearly every tech giant (e.g. Apple, Alphabet, and Microsoft), wearable devices are proving to be a staple product in today's society. The wearable market was valued at 15.74 billion dollars (USD) as of 2015, and expected to raise to 51.60 billion by the year 2022 (Markets and Markets, 2020). This growth trends directly with the growing popularity of the internet of things and the desire to gain further insight into one's health.

There is no denying that wearable devices are here to stay, which is why the data of these devices are a major area of research today. A large portion of these wearables consist of similar if not the same sensors, but with the difference between products being the way the data is analyzed/presented after being captured (Aroganam, Manivannan, & Harrison, 2019). Whether these wearables are being used at the individual level or a macro-level in healthcare, the need to enhance insight will always be present.

The device used in this study is the “Fitbit Charge 2” a midrange activity tracker ,which was released in 2016 (Peckham, 2019). The specifications can be found in Table 1 below.

Table 1: Fitbit Charge 2 Specifications (Westenberg, 2016).

Fitbit Charge 2	
Display	1.5-inch multi-line OLED Tap Display
Heart Rate Monitor	Optical Heart Rate Monitor
Sleep Tracking	Automatic
Estimated VO2 Max	Yes
Battery Life	Up to 5 days
Sensors	Optical Heart Rate Monitor 3 Axis Accelerometer Altimeter Vibration Motor
Price (2016 relative)	$149.95

The Fitbit Charge 2 tracks and collects the data by using a combination of the sensors within the device (Figure 1). To monitor heart rate Fitbit justifies: “When your heart beats, your capillaries expand and contract based on blood volume changes. To determine your heart rate, the optical heart-rate sensor in your Fitbit device flashes its green LEDs hundreds of times per second and uses light-sensitive photodiodes to detect these volume changes in the capillaries above your wrist. Then your device calculates

how many times your heart beats per minute (BPM). The optical heart-rate sensor detects a range of 30-220 BPM (Fitbit, 2020, p. 1).

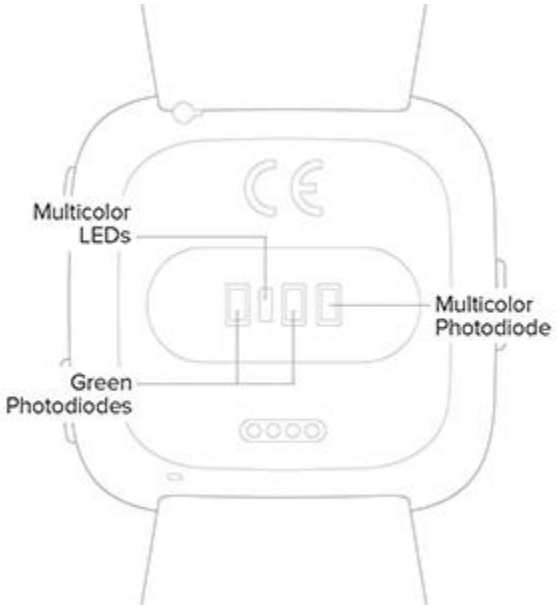


Figure 1: Layout of optical heart-rate sensor (Fitbit, 2020).

One of the main components of any wearable device is the 3-axis accelerometer that determines which way is up, down left or right, while also providing detailed information about frequency, duration, intensity, steps taken, distance traveled, calories burned and sleep activity (Fitbit, 2020). As can be seen in figure 1, the use cases revolving around the 3-axis accelerometer are plentiful, but these are all derived from transforming physical movement data into digital form and then analyzing the information to draw conclusions such as step count. The true power of the results come from Fitbit's internal algorithms. Fitbit states that their devices have "finely tuned algorithms for step counting", alluding that the algorithm is looking for motion patterns that are most indicative of people walking and determines whether a motion size is large

enough by using thresholds (Fitbit, 2020). No in-depth technical details regarding the algorithms have been published by Fitbit.

The 3-axis accelerometer in combination with the optical heart-rate monitor are used to determine sleep patterns and stages. If a user remains motionless for around an hour, the device assumes the user is asleep, and short movements such as rolling over helps confirm that they are asleep (Fitbit, 2020). The optical heart-rate monitor is used to track/estimate what stage of sleep the user is in by heart rate variability (HRV), which fluctuates when transitioning to light sleep, deep sleep and REM sleep stages.

2.2 Quantify Yourself

> "The quantified self is any individual engaged in the self-tracking of any kind of biological, physical, behavioral, or environmental information. The self-tracking is driven by a certain goal of the individual with a desire to act upon the collected information." (Swan, 2013, pp. 85-89)

The concept of quantifying oneself comes from the new world fact that sensors are now all around us, whether it be a watch, a smartphone or anything else that is capturing data. These sensors capture enormous amounts of data with a variety of use cases, the current issue being is how to retrieve meaningful information from the all the noise also being captured (Hoogendoorn & Funk, 2018). One faucet of this new reality is how to use this data for one's own advantage. One popular example of quantifying oneself can be someone wanting to track how fast and far they ran during their exercise,

so they use their smart phone which utilizes an assortment of sensors to be able to do so. The data captured by the phones sensors during their exercise is summarizing and giving meaningful information back to the user while cutting out the noise. This one example is extremely popular and has watches and other smart devices strictly to give more accurate and precise information back.

When the use cases is explicitly stated before the activity (e.g. track how far/fast I will run), the sensors can be tailored to that environment to reduce incoming noise. However, in a use case like ours where the experiment takes place over a 2-month period, the noise will vastly overpower the meaningful information. This is where quantifying oneself becomes more difficult to achieve and needs the introduction of creating a process of data cleaning, identifying strong/weak parameters and utilizing unsupervised machine learning models to enrich the output data of these sensors.

Unsupervised clustering is currently the most popular category for wearable information due to the ability to define clusters in an unlabelled dataset and its simplicity. Clustering is a focal point in the 2018 textbook “Machine Learning for the Quantified Self” by Mark Hoogendoorn and Burkhardt Funk dedicating a chapter to the usage of clustering with wearable data (Hoogendoorn & Funk, 2018).

2.2 Clustering Approaches

There are numerous clustering algorithms defined today in general, that work with the notion of distance between data points (Hoogendoorn & Funk, 2018). There are many possible distance metrics used to determine where a cluster may lay, and which

points belong to one. Three popular distance metrics are the Euclidean distance, Manhattan distance and Minkowski distance, each with a different impact on any clustering approach.

The Euclidean distance is the most commonly used distance metric and what is utilized in our research with the k-means algorithm. Euclidean distance computes the root of square difference between a co-ordinate of a pair of objects (Singh, Yadav, & Rana, 2013). This can simply be described as the distance between points.

$$\textit{Euclidean distance } (x_i x_j) = \sqrt{\sum_{k=1}^{p} (x_i^k - x_j^k)^2} \quad (1)$$

The Manhattan distance differs from the Euclidean distance as it determines the distance between points based on that you can only connect points based on a horizontal or vertical movements and not diagonally, as Euclidean distance (Hoogendoorn & Funk, 2018). Instead of computing the root of square difference, it takes the absolute difference between a co-ordinate of a pair of objects (Singh, Yadav, & Rana, 2013). This is also known as the "City Block Distance" as it works in a grid-like path.

$$\textit{Manhattan Distance } (x_i, x_j) = \sum_{k=1}^{p} \left| x_i^k - x_j^k \right| \quad (2)$$

The Minkowski distance is a generalization of the two and can be manipulated to use either the Euclidean or Manhattan distance metric by switching the *p*-value. If $p = 1$ then it becomes the Manhattan formula, where $p = 2$ changes it to the Euclidean distance metric (Hoogendoorn & Funk, 2018).

$$\textit{Minkowski Distance } (p, x_i, x_j) = \left(\sum_{k=1}^{q} \left| x_i^k - x_j^k \right|^p\right)^{\frac{1}{p}} \quad (3)$$

Each of these distance metrics are primarily dataset dependent and should be chosen based on what type of data is needed to be used in the clustering algorithm. If there are any categorical attributes used these listed distance measurements will not suit the need. There is the possibility to convert categorical information to a categorical state but other distance metrics are better sought out.

As previously noted, we are using the Euclidean distance with our k-means algorithm. The k-means algorithm dates back to the 1950's and is still widely used today due to its computational efficiency and teamed with Euclidean distance makes it even more so (Singh, Yadav, & Rana, 2013). K-means works by taking initial points of a dataset and putting them into an initial but not optimal cluster. Then it relocates each point to the nearest center (Cluster centroid) and updates the cluster centers by calculating the mean of the member points to eventually placing each point into its optimal cluster. This process can be performed in multiple iterations defined by the initial parameters (Jin & Han, 2010).

K-means can also be used not just on offline (static) datasets, but also on online (dynamic) datasets. This is what can be defined as stream clustering. Stream clustering is used to provide fast responses where new data points coming into the data stream are analyzed in near to real-time. This provides insight almost immediately and can be largely beneficial with wearable data.

Stream clustering with k-means begins with clustering the initial dataset (if applicable) and then updating the location of the clusters as new data enters the stream to provide more optimal cluster centers. A few issues arise when stream clustering, one of the main problems comes from the changing distribution of the data as the dataset changes over time. This is also known as concept drift (Carnein & Trautmann, 2019). To assist with concept drift, time windows are used in order to summarize data into certain defined time-based windows.

2.3 Time Windows

Time windows in data streams are used to assist with the changing distribution of data over time. There are four main time window models currently popular in literature; the sliding, landmark, pyramidal and damped time window (Silva & Faria, 2013). Each with their own unique way of handling the incoming data.

The sliding time window is used by only considering the most recent data points, based on either a fixed or variable window length. After new points come into the stream, the older points that fall out of the window length are discarded offline. Having a small or large window length can dramatically change the resulting micro-clusters, where smaller windows can handle concept drift better but may lack in cluster accuracy in comparison to larger windows (Carnein & Trautmann, 2019).

The damped time window model utilizes a decay rate in order to fade older data points in the stream. With a set decay rate, the micro-clusters and points within will gradually diminish on its impact on the next micro-cluster. This is computationally more

expensive but supposedly provide more accurate micro-cluster results due to the data still residing in the stream (Silva & Faria, 2013).

In addition to these models, there is the landmark and pyramidal time window. The landmark model is similar to sliding window model in that it breaks up the data stream into groups. Instead of having a window length that cuts off incoming data, it can also use “events” (Carnein & Trautmann, 2019). For example, an event could be someone passing above a heart rate of 100, where a new data stream is created to generate micro-clusters above that threshold. The pyramidal model uses different granularity levels based on the recency of the data (Aggarwal, Han, & Wang, 2003). This works by aggregating the older data points and keeping the most recent points at the highest level of detail. In our research we will be specifically focusing on the damped and sliding time window models to test against each other.

Chapter 3: Literature Review

Although seemingly a modern invention, wearable technology has been around since the early 1960's when Claude Shannon and Edward Thorp invented what is known as the first wearable computer in order to beat a casino game of roulette (Thorp, 1998). Today wearable devices are more prevalent in our society and used for a plethora of different reasons, from fashion, safety, sleep tracking and activity monitoring (Haghi, Thurow, & Stoll, 2017). With the advancement of the hardware behind these wearables, it is evident that these devices will continue to become a common place in our society due to the endless use cases available and the allure of integrating these sensors to acquire more data on the users' behavior.

One of the main use cases comes from the health sector. Devices such as Fitbit make use of sensors through a wearable watch to track the health and fitness activities of the user. Invented in 2007, Fitbit has become the current market share leader for wearables across the world (Liu, 2019). Currently, Fitbit has seven different models in the market, all varying in price, health features, exercise features, smart features and design style. All the current models feature an accelerometer to measure acceleration and determine orientation used to compute step count, where five types (i.e. Versa 2, Versa, Ionic, Charge 3, Inspire HR) utilize a heart rate sensor to measure a user's heart beats per minute (Fitbit, 2020). With as many as 27 million people using these devices, the influx of activity related to data availability and data rates increase tremendously (Liu, 2019). This leads too many questions regarding its actual accuracy in terms of step count, heart rate, and activity recognition, among other variables collected.

Recent research has tested different Fitbit models to determine whether they are effective at monitoring physical activity, and whether they can potentially be used by healthcare professionals to guide decision making and treatment plans. Feehan et al. (2018) evaluated 67 studies and experiments carried out by other researchers in the field to evaluate the data reliability of Fitbit devices. They found consistent evidence indicating that these devices would meet an acceptable accuracy for step count only half the time, with a tendency to underestimate steps in a controlled setting, while overestimating in a real-world setting. They further describe the accuracy rates for different activities such as jogging, sleeping and slow walking in comparison to research grade accelerometers. When measuring a user's sleep activity, such as sleep time and time in bed; the Fitbit devices provided similar measurements in comparison to the research grade accelerometers such as an Actigraph. They recommend using discretion when considering using Fitbit devices as an outcome measurement tool in research and making health care decisions, bearing in mind this is less so in adults with no mobility issues.

Due to the data reliability, the use of Fitbit devices have been limited to physical activity monitoring to produce acceptable accurate results. Koolean et al. (2019) proposed a method to relate physical activity to physical capacity. This was done by using a quadrant method to place individuals into different categories based on one variable (i.e. step count) to represent physical activity, and one variable (i.e. 6 minute walk distance (6 MWD) to represent physical capacity. If an individual had a high step count but low 6 MWD, then he would be categorised in "Can't do, do do" which represents that he does not have capacity to do what he is doing, and vice versa for the

other categories. The notion that physical activity can be represented by step count is currently accepted in the Quantified Self domain, however, it is still an issue of debate coming back to how reliable the wearable devices are, and also how just one variable can accurately represent a user's capacity level.

From an analytical perspective, Carnein et al. (2019) provide an extensive survey on stream clustering algorithms, outlining how each algorithm performs during a streaming process by delineating their advantages and limitations. The overall strategy is based on a two-phase clustering approach, having an online phase, which uses a time window model to capture the data streams and then computing micro-clusters (i.e. preliminary clusters within each time window). The second phase is carried out offline as the micro-clusters are re-clustered to generate the macro-clusters after the entire stream data is processed. The use cases revolve largely around clustering sensor based data streams due to the need of supporting real-time communication between the sensors themselves and the resultant output. More in-depth investigation is needed for supporting multi-window analytical workflows, identifying which stream clustering algorithm should be used, and which time window actually reveals interesting self-quantified patterns from a vast amount of wearable data streams.

When developing stream clustering algorithms, the focus has also been on extending current approaches, which have been based on density, distance, and partitioning strategies, having a time window model being embedded within. The partitioning based stream clustering algorithms are of the most popular types of algorithms due to their simplicity and low parameter tuning in comparison to grid and distance based approaches (Carnein & Trautmann, 2019) . One of the most popular

approach is the streaming k-means clustering algorithm, with an assortment of different time window models utilized, but rarely has the focus been on the impact of a particular time window on the discovered patterns.

The streaming k-means clustering was chosen due to its ease of use and overall popularity amongst clustering algorithms. Originally published in 1955, k-means has stood the test of time due to its simplicity and overall insightful results associated with many use cases (Jain, 2010). Research revolving around the usage of k-means can be found ranging from the 1960's to today in 2020 (MacQueen, 1967). The idea behind k-means can also be traced back to 1957 from polish mathematician Hugo Steinhaus (Steinhaus, 1956). With the nature of wearable stream data, an unsupervised learning method is needed to further gain new insights, and using k-means meets this requirement as well as provides easy comparisons to the numerous amounts of use cases that have as well applied k-means in the past.

From a temporal perspective, time windows have been used to extract small, quasi-static subsets from the data streams (Hahsler, Bolanos, Forrest, & al., 2017). The main time window models proposed in the literature are damped, sliding, landmark and pyramidal. The damped model assigns a weight to each time frame, while over time the older data decays by a function, which gives a higher importance to recent data. The sliding time window model only considers the most recent data where the older data is removed once new data is available. The landmark time window model separates the data based on a set time interval (e.g. 1 hour) or by event (e.g. every 10 steps), where after a landmark is reached, new data starts to be captured in a separate window. The pyramidal time window model uses time granularity to easily summarize past data into a higher

level of granularity (e.g. 1 hour data rate summarized into daily records). Figure 2 illustrates the main characteristics of these time window models.

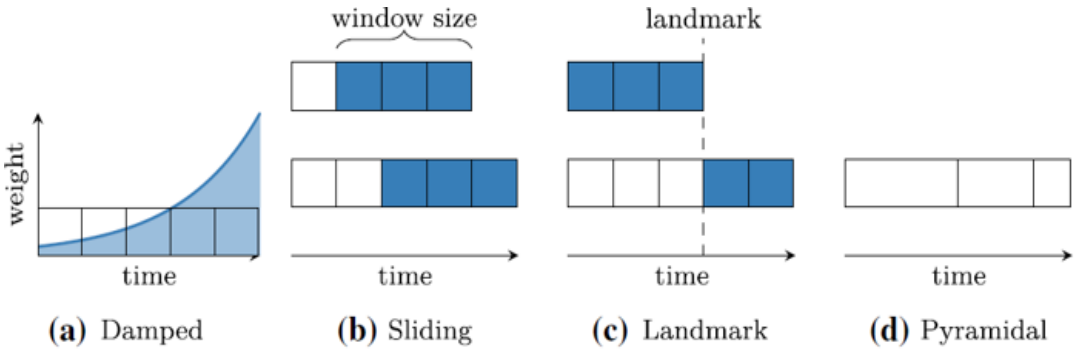


Figure 2: Time window models (Carnein & Trautmann, 2019)

The sliding time window model has been previously proposed to improve clustering results from wearable data streams. Park et al. (2019) provide empirical results showing the main limitations of considering a wearable device log as one whole snapshot rather than considering accumulated wearable data streams using a time window. They were able to find insightful consecutive insomnia-activity clusters of individuals with similar sleep-related dysfunctions by coupling sliding time windows of an 8-day period of daily intervals with a neural-net based unsupervised method, using various information modalities from smart bands. Transforming wearable device logs into multiple sets of binned data has also been proposed for improving clustering results that reflect both local and global patterns (Jang, Oh, Lim, & Cheung, 2020).To the best of our knowledge, no previous research work has been focused on exploring wearable stream data by coupling different time window models with streaming k-means clustering.

Interestingly, Keogh and Lin demonstrate that clustering time series sub-sequences is meaningless while using a sliding window. They state that since the output is independent of the input that a time window is meaningless. Their research can be considered as the first disclosure of the importance of investigating the use of multiple types of time window models to ensure that clustering results gathered have meaning (Keogh & Lin, 2005).

In summary, this chapter has elaborated on the current state of stream clustering in general and examples of current use cases for wearable information along with the difficulties with analyzing this type of data. The potential going forward in this increasingly popular research domain is immense considering its wide reach into different areas of research and industry. Using time windows to further enhance the usage of wearable data is a steppingstone into gaining more insight into quantifying oneself as well as others.

Chapter 4: Multi-Window Analytical Workflow

This Chapter describes the proposed analytical workflow that was developed to cluster wearable stream data using the sliding and damped time window models. The workflow consists of six phases as shown in Figure 3, which are described as follows:

- Data Collection Phase: The process of gathering wearable device logs from individual users.
- Data Preprocessing Phase: Encompasses the cleaning, transforming, and encoding tasks.
- Data Stream Simulation Phase: The process of generating streams from a user's wearable data log.
- Online Micro-Clustering Phase: The tasks that are necessary to compute the micro-clusters using the sliding and damped time window models.
- Offline Macro-Clustering Phase: Generates the macro-clusters by re-clustering the micro-clusters found in each time window model.
- Quantified-Self Phase: The process of visualizing the clustering results to outline the self-quantified patterns of a user.

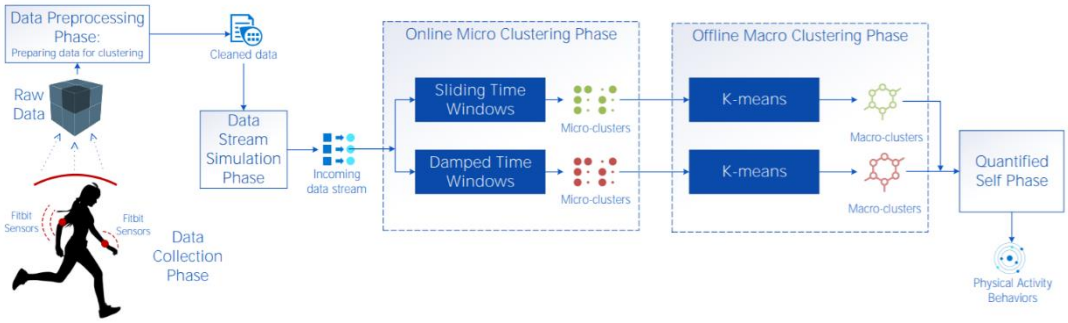


Figure 3: Main phases of the multi-window analytical workflow.

4.1 Data Collection Phase

This phase consists of retrieving Fitbit device logs, which are usually stored within the device and can only allow for retrieval once connected to a computer or synced to a third party cloud platform. Initial pre-sets are needed to account for individual tracking, such as personal information such as age, weight, height and sex. Other sensors connected to the Fitbit devices also generate data such as heart rate, steps, temperature, and location, which contribute to a different range of physical outputs.

Fitbit device logs are usually fetched as an offline data package. Third party platforms such as Fitabase can allow wearable device logs to be retrieved in its raw format in comparison to usual summarized data from device manufacturers. In conjunction with this, if continuously synced to the platform the offline data can be monitored in real-time, acting as a data stream on itself. Once the offline logs are fetched, the Data Preprocessing Phase is initiated as described in the next section.

However, retrieving raw data streams directly from the devices online software, brings many technical issues since the information is currently generated to be as

simplified as possible for the end user. The implementation of new capabilities need to be developed by the manufactures to allow access to raw wearable data streams that are essential for the next generation of multi-window analytical workflows.

4.2 Data Preprocessing Phase

Data preprocessing is an important phase in the proposed analytical workflow. The ultimate goal of this phase is to clean, encode and transform the Fitbit data logs (i.e. raw data) into a revised format that is easily readable by a machine in such a way that data points can be easily processed by the streaming k-means clustering algorithm. Guaranteeing data quality and providing accurate data points is key to the success of the subsequent analytical phases in the proposed workflow. However, due to usage deviation, limitations of Fitbit devices, or flaws in the data collection phase, it is not realistic to expect that the raw data will always be ready to be analysed. Therefore, five main data preprocessing tasks are designed to deal with the common issues to prepare reliable data points for the next phases. They are described as follows:

- Missing data points: Missing data points may leave significant contextual information out for discovering insightful clustering patterns. Whether they are missing due to connectivity issues, sensors malfunction, or human errors, the missing data points are ignored unless considered significant, then further investigation in the Collection Data Phase is needed to determine the main causes.

- Duplicated data points: When the same data point is collected twice, any duplicate identified via the same timestamp is removed, to alleviate any potential bias introduction.
- Missing variables: There can be many reasons why a variable is missing. Some missing variables are acceptable due to the time frame of collection being different among variables. For example, some variables might be collected using minute rate versus hour rate. In contrast, when variables such as step count or heart rate are missing, it is either a function of device error or during a moment of not wearing the device. The data points containing missing variables are deleted to avoid introducing errors
- Redundant variables: The data points contain a set of unique variables, but there may be syncing errors either from the device or from the platform, that can generate a duplicated and unnecessary variable. In this case, the extra variable is removed.
- Variable selection: This task is also known as Feature Selection. The major task in this phase, since it aims at the selection of relevant variables (i.e. features) which can maximally improve the analytical workflow performance in finding meaningful clusters. For an overview of previous research on developing feature selection approaches for clustering please refer to (Hancer, Xue, & Zhang, 2020).

Once the Data Preprocessing Phase is completed, a target data set is ready to be used by the Data Stream Simulation Phase.

4.3 Data Stream Simulation Phase

Data streams are a countable infinite sequence of data points that can be formalized as follows (Rabl, Sakr, & Hirzel, 2018):

$$T = [t_1, t_2, ..., t_n]$$

where each data point contains many sets of variables as follows:

$$[t_1 = (P_1, S_1, Q_1, X_1, U_1)]$$

$$[t_2 = (P_2, S_2, Q_2, X_2, U_2)]$$

.....

$$[t_n = (P_n, S_n, Q_n, X_n, U_n)]$$

where

- P_n: is a set of categorical variables related to personal information (e.g. age, weight, height, and sex);
- S_n: is a set of numerical variables related to sensor measurements (e.g. temperature, vibration, and location);
- Q_n: is a set of ordinal variables related to ratio scales (e.g. sleep quality and activity intensity);
- X_n: is a set of numerical variables related to physical measurements (e.g. step count and heart rate); and
- U_n: is the identifier of a wearable device.

This research replicated the stream process using the target data set created in the previous phase, since the current manufacturers do not support this capability. To achieve

this, an assortment of frameworks are available to simulate or connect to a data stream, including MOA (Massive Online Analysis), MLFlow, and Stream R (Bifet, Holmes, Pfahringer, Kranen, & Kremer, 2010). The Stream R framework to simulate the data streams, compute the clusters and visualize the results (Hahsler, Bolanos, Forrest, & al., 2017). This framework is further explained in Chapter 5.

4.4 Online Micro-Clustering Phase

For this phase, the simulated data streams of each Fitbit device arrive as a continuous sequence of data points that are accumulated using a time window model. Each time window has the same time frame (e.g. 2 hours). The streaming k-means clustering algorithm requires to incrementally update the computation of the micro-clusters, which are represented by their respective k_0 centroids.

A micro-cluster represents a set of similar data points, created using a single pass over the data currently available within a time window. The algorithm selects k_0 random data points as seeds until clustering converges in such a way that for each time window, any new data point t_i is always assigned to one unique micro-cluster mc_j by minimising the sum of square distances (Singh & Bhatia, 2011). Therefore, a centroid is the center (i.e. the mean point) of a micro-cluster belonging to a specific time window.

Figure 4 illustrates how the k-means algorithm steps work in practice, with a k' = 2. First, the data points belonging to the initial time window are set as shown in Figure 4a, where the randomized k' centroid locations are placed as illustrated by the red and blue crosses in Figure 4b. In Figure 4c, each data point is assigned to the nearest centroid

which creates the two micro-clusters (i.e. red = cluster 1, and blue = cluster 2). Through Figures 4d to 4f, the centroid locations are converging through each iteration and moved closer to the mean of the current data points assigned to this initial time window. Figure 4d shows the final micro-clusters and the location of their centroids. These steps are repeated for the next time windows until the data streams are completed.

There are two approaches for selecting the k_0 partitions for computing the micro-clusters. The first approach is based on applying the elbow method for computing the optimal number of k_0 partitions for each time window. In this case, the number of centroids will vary from one time window to another. The elbow method is further explained in Section 4.5. A second approach consists of applying a fixed number of k_0 partitions for all time windows. In other words, it is assumed that the optimal number of micro-clusters should be the same across the time windows. The choice between these two approaches will depend on the selected variables for performing the clustering.

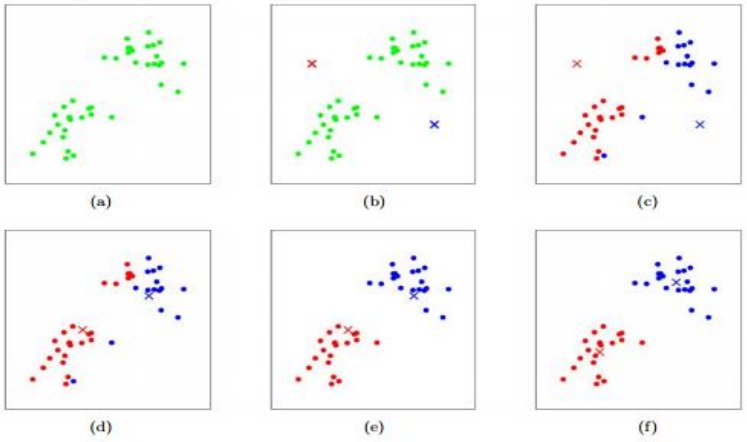


Figure 4: The K-means Algorithm illustrated by (Piech, 2013).

This phase is the most important in the proposed analytical workflow, as the different time windows used have a direct impact on the computation of micro-clusters, and play an important role in finding the macro-clusters in the next phase.

Sliding Time Window Model

The sliding time window model only considers the most recent data point for computing the micro-clusters, since the older data point is removed once a new data point is available. A start window is initiated having an a-priori defined time frame (e.g. 2 hours) and containing the accumulated data points that were streamed during this time frame. As soon as the new data point arrives, the algorithm incrementally updates the micro-clusters. The next window utilizes all the new data points and clusters as the data points enter and exit the stream, storing the micro-cluster and its representative centroid after each completion.

It is important to point out that using a sliding time window model, the minimization of the sum of square distances will often terminate at a local optimum, as expected when using k-means clustering. Therefore, the analytical workflow aims to gather insights on the evolution of micro-clusters rather than a full explanation as to why the data points were grouped under them. The main focus is on exploring the evolution of micro-clusters to distinguish new clusters from outliers, indicating the actual cluster evolution from the wearable data streams.

Damped Time Window Model

The damped time window model continuously adds new data points into the feature space with each iteration lessening the weight of each point, the less weight it has the less it contributes to generating a micro-cluster. This is done to give the highest weight to the most recently captured instances. The streaming k-means algorithm computes the k_0 micro-clusters after a set of data points are damped due to the decay function. As defined in (Cao, Estert, Qian, & Zhou, 2006), the weight of each data point within a damped time window decreases exponentially with time t using the decay function

$$f(t) = 2^{-\lambda t} \tag{4}$$

where λ should be always greater than 0.

The smaller the value of λ, the most important the historical data points are in comparison to the current data points. This makes the damped time window model effective to indicate the cluster scalability. In this research, the cluster scalability is defined as the maximum number of data points, which guarantees context consistency in

the data streams that are needed to compute micro-clusters. This time window model supports the ability of a micro-cluster to grow while conforming within the apriori k_0 partitioning. The outcomes from this phase are two sets of k_0 micro-clusters, one for each type of time window model being used for the computation. They will be re-clustered as macro-clusters in the next phase.

4.5 Offline Macro-Clustering Phase

After the simulated streaming has ended and all the centroids of the micro-clusters have been computed using both time window model, macro-clusters are generated by re-clustering these centroids. The k-means algorithm is again used to compute the final k macro-clusters. The k centroids will be generated from re-clustering the k_0 centroids of micro-clusters found using the sliding and damped time window respectively. The benefit of using the proposed offline clustering is to gain further insight from the entirety of the k_0 centroids after it has finished streaming. The goal being to not add stress to the stream flow itself due to the half the process being performed on a solid state (Ghesmoune, Lebbah, & Azzag, 2016).

Determining the optimal number of k macro-clusters can be obtained by using a well-known method known as the elbow method (Marutho, Handaka, Wijaya, & et, 2018). This method works by running the k-means algorithm on the k_0 centroids of all micro-clusters computed for all sliding time windows as well as damped time windows. It computes the sum of the squared distances from each data point to its assigned centroid, essentially looking at variance and adding another cluster stops for improving the model

(Ketchen & Shook, 1996). The line chart resembles an arm and the elbow represents the point of inflection of the curve as shown in Figure 5. The range used in this case was from 2 to 14 clusters with the optimal *k* value resulting in 5 where the line resembles the arm and the elbow represents the point of inflection of the curve.

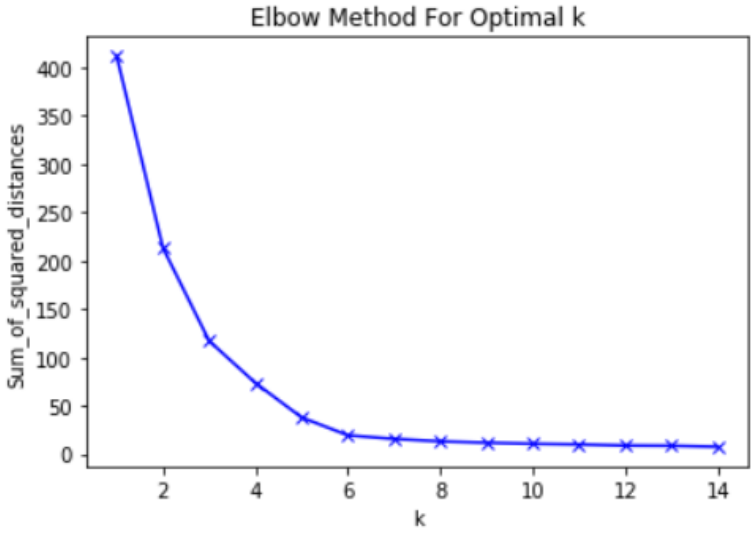


Figure 5: The elbow curve found for a k range from 1 to 14.

4.6 Quantified Self Phase

The Quantified-Self phase begins with plotting each time window k_0 micro-clusters, and comparing them with the final k_0 macro-clusters. Moreover, external variables obtained in the Data Collection Phase are also used to label the final macro-clusters. Each time window will consist of different micro-clusters, which will directly impact the meaning behind their macro-clusters.

The macro-clusters ultimately represent self-quantifying patterns that can be interpreted as regular and irregular physical activity behavior. They may facilitate our understanding of the reasons leading to individual changes in lifestyles and health care settings. Therefore, the outcomes of the proposed multi-window analytical workflow provide empirical evidence that captures the range of self-quantifying patterns on behavior, these outcomes also offer an analytical perspective that may help identifying a range of variables involved in monitoring behavioral changes in physical activities.

Chapter 5: Implementation

5.1 Data Collection Phase

The wearable device logs used in this research were collected from an intervention experiment performed at Flinders University, Australia, where 15 participants continuously wore a Fitbit Charge 2 device for approximately a 2 month period. The raw Fitbit data consisted of approximately 87,600 data points per participant with a data rate of one data point per minute containing 32 variables as shown in Table 2.

Typically, extracting the data directly from the Fitbit device is possible but limited due to the majority of the data being summarised, rather than providing a minute by minute description of the collected data. To address this issue, Fitabase, a third party research cloud platform designed to collect data from Fitbit devices with more diverse options, was used. The major benefit being the raw data can be extracted in a range of formats. Using the third party cloud platform allows the data to be retrieved on a per participant basis, or in a batch with all participants on one spreadsheet. To keep the participants separate, the raw data was retrieved on a per participant basis, and transformed to multiple .CSV files in order to be used in the next phase.

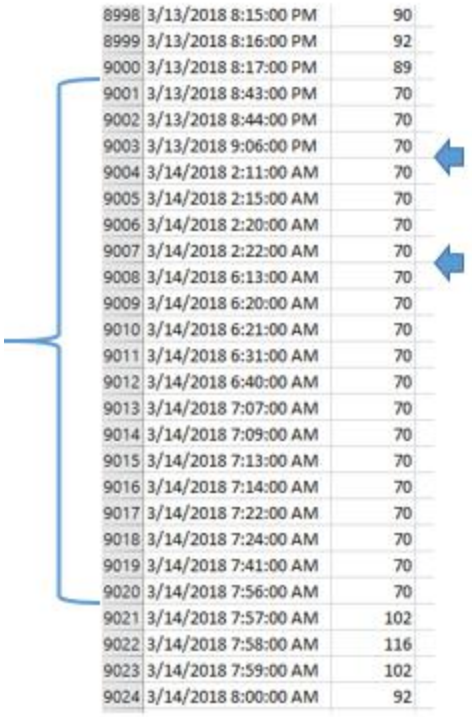
Table 2: List of the variables collected for each participant during the experiment.

Attributes	**Measurement**	**Definition**
Timestamp	1 minute intervals	From start of study period (01/03/2018) to end (30/04/2018)
Gender	0=Female, 1=Male	Gender of Participant
Age	numeric	Age of Participant
Weight	kgs	Weight of Participant
Height	cm	Height of Participant
PID		Participant ID
Steps	1 minute intervals	From Fitabase
HR	1 minute intervals	From Fitabase
Intensity	1 minute intervals	From Fitabase
Sleep_Dur	1 minute intervals	From Fitabase
Temp	15 minute intervals	Temperature indoor, one decimal point. Adjust for Daylight savings
BatteryLevel	0, 1, 2, 3	From Fitabase, measured when device is synced, random, 1=flat, 3=full, 0 means nothing/ignore
Days_elapsed		Days since the last sync event
Hours_elapsed		Hours since the last sync event
DOW	1, 2, 3, 4, 5, 6, 7	Day of the week, 1=Sunday, 2=Monday, 3=Tuesday, 4=Wednesday, 5=Thursday, 6=Friday, 7=Saturday
Hour	1, 2, 3, 4, 5, ... 23	Hour of the day, 1am=1, 2am=2, 3am=3 ... 11pm=23
Date	dd/mm/yyyy	Date
Time	hh:mm:ss AM/PM	Time
Missing	0=no missing data, 1=missing data	Missing = 1 if heart rate date is missing. If missing = 1, worktime/sleeptime/leisuretime = 0. Missing will never be blank
Weekends	0=Weekend, 1=Weekday	Weekend or not weekend, Saturday/Sunday = 1, Monday-Friday = 0
Pub_Holiday	0=Not a pub hol, 1=a pub hol	Public holiday (30/03/2018, 31/03/2018, 02/04/2018, 25/04/2018)
Leave	0=Not a leave day, 1=a leave day	Self reported taken a personal leave day (pub hols are also counted as leave)
WorkfromHome	0=Not a WFH day, 1=a WFH day	Self reported that worked from home (can be a weekend/pub hol and still a WFH day)
WorkTIME	Self-reported	Hours of work - self report minus 30 minute from start and end of day, cannot be a weekend or holiday/leave day
Sleeptime	1 minute intervals	As determined by fitbit and downloaded from Fitabase
Leisuretime	1 minute intervals	Time not defined as Sleeptime or Worktime
Check	1	Must be 1 or there is an error. Sum of worktime, sleeptime, leisuretime and missing
AM_transit	Self-reported	Travel to work time +30 minutes, may cross over with work time by one minute
PM_transit	Self-reported	Travel to home time +30 minutes, may cross over with work time by one minute
Lunchtime	Self-reported	Lunch period +30 minutes

5.2 Data Processing Phase

During this phase, a variety of preprocessing tasks were performed to determine the quality of collected data points. A number of issues were detected that allowed for insight on data quality. One instance was a participant who did not wear the device to bed, showing a defined gap in the data. Another instance was a failure to sync to the platform, which lead to missing minute-to-minute values, although the daily step count and heart rate were still collected. This also resulted in a complete loss of sleep data during these intervals.

Due to lack of connectivity, devices not worn, and sensor problems, missing data points occurred during participant data streams. Figure 6 shows the heart rate not being collected for every minute interval as it should, with no possible explanation on why this was happening. Another example being a participant having mismatched variables that should be the same such as sleeping and quality of sleep, it may show in one category that a participant was asleep, but awake in the other. Finally there were instances of times when it was proven a participant was not moving (via video recording or a time use diary) where step counts were recorded.



8998	3/13/2018 8:15:00 PM	90
8999	3/13/2018 8:16:00 PM	92
9000	3/13/2018 8:17:00 PM	89
9001	3/13/2018 8:43:00 PM	70
9002	3/13/2018 8:44:00 PM	70
9003	3/13/2018 9:06:00 PM	70
9004	3/14/2018 2:11:00 AM	70
9005	3/14/2018 2:15:00 AM	70
9006	3/14/2018 2:20:00 AM	70
9007	3/14/2018 2:22:00 AM	70
9008	3/14/2018 6:13:00 AM	70
9009	3/14/2018 6:20:00 AM	70
9010	3/14/2018 6:21:00 AM	70
9011	3/14/2018 6:31:00 AM	70
9012	3/14/2018 6:40:00 AM	70
9013	3/14/2018 7:07:00 AM	70
9014	3/14/2018 7:09:00 AM	70
9015	3/14/2018 7:13:00 AM	70
9016	3/14/2018 7:14:00 AM	70
9017	3/14/2018 7:22:00 AM	70
9018	3/14/2018 7:24:00 AM	70
9019	3/14/2018 7:41:00 AM	70
9020	3/14/2018 7:56:00 AM	70
9021	3/14/2018 7:57:00 AM	102
9022	3/14/2018 7:58:00 AM	116
9023	3/14/2018 7:59:00 AM	102
9024	3/14/2018 8:00:00 AM	92

Figure 6: Examples of missing data points.

Although the Fitbit data logs had a fixed number of variables, there were cases when a new variable was added to a data point during the transport to the cloud platform. For example, in the case of having a set of 32 variables, it has occurred that 33 variables were retrieved instead. In this case, the duplicated variable was removed.

Temperature was being captured from a sensor located at the office of each participant. Considering that the participants were not always in their offices, this variable provided an unrealistic context on the participants' behavior. Adding outdoor weather information may have provided additional context to explaining some self-quantified patterns. However, there was little to no rain during the duration of the 2 month experiment.

Finally, the variable selection task was performed to prepare a target data set ready to be used by the two phase clustering algorithm. For this research, three numerical variables as summarised in Table 3.

Table 3: Selected input variables for clustering the data points.

Variable	Description
HR	Heart rate/min
Steps	Steps/min
HRV	Heart Rate Variability

5.3 R Stream Framework

The R Stream framework was used to simulate the data streams, generate the time windows, and run the online and offline clustering. Therefore, the Data Stream Simulation Phase, the Online Micro-Clustering and Offline Macro-Clustering Phases were implemented using the Stream R framework, seamlessly integrating the extensive existing R packages, including streamMOA, cluster, clusterGeneration, and fpc.

The overall architecture is shown in Figure 7. Initially, the Data Stream Data (DSD) was used due to its ease of use and ability to simulate a live stream from any .CSV file. Table 4 illustrates the .CSV format of our target input data set used for generating the stream simulation.

Figure 7: Overview of the Stream R architecture.

Table 4: The input data for stream simulation.

Timestamp	Steps	HR	HRV	PID
Mar-6-2018 12:00:00 AM	0	71	0	18
Mar-6-2018 12:01:00 AM	0	71	0	18
Mar-6-2018 12:02:00 AM	0	69	-2	18
Mar-6-2018 12:03:00 AM	0	70	1	18
...	...	...	...	...
...	...	...	...	...
Apr-30-2018 11:58:00 AM	0	65	-2	18
Apr-30-2018 11:59:00 AM	0	66	1	18

The Stream R framework was also developed to focus in the domain of data stream clustering which fits well to our need. The Data Stream Clustering (DSC) was used to compute the online micro-clusters, with the option of both a sliding window or damped window, and then passing it on to the offline phase which consisted of generating the macro-clusters.

In conjunction with implementing the time window models using the Stream R framework, we have also explored different time frames that could generate the most meaningful self-quantified patterns whilst not being computationally expensive were also explored. The initial selected time frame had one hour time intervals (60 data points), and

was expected to be an acceptable time frame due to the ability to visually recognize clusters.

However, after adding more time for the time frame, it was discovered that a two-hour interval was more appropriate. Figure 8 demonstrates that the micro-clusters are more diversified with the addition of an extra hour of data. Further, that there was more distinction between areas of no steps being taken, low amount of steps taken (< 15 steps) and medium to high amount of steps (> 15 steps). The 2-hour time frame was selected for implementing both time window data models since it provided a richer context for generating the micro-clusters that have optimised and further investigated to infer the self-quantified patterns for each participant.

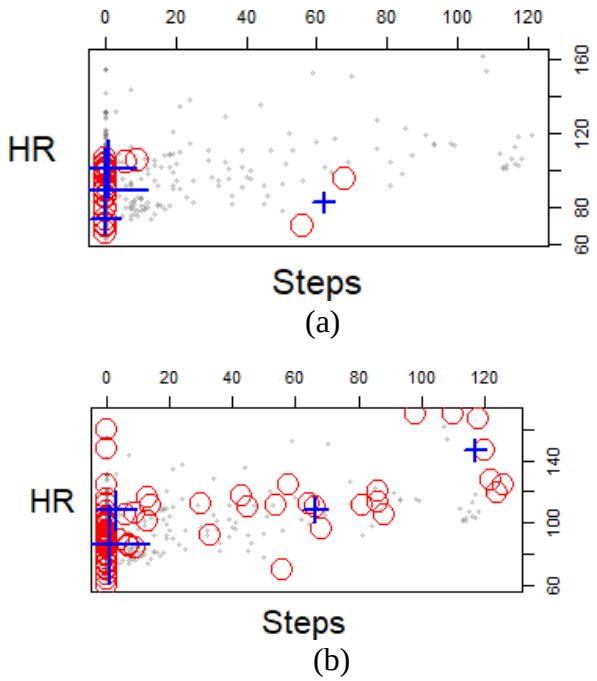


Figure 8: Macro-cluster centroids (blue crosses) and micro-cluster centroids (red circles) results using the sliding time window model: (a) with 1-hour time frame, and (b) with 2-hour time frame.

For the sliding time window model, the online clustering initialises with 120 data points within the initial window where gradually over time new data points are introduced and old data points are disposed. Micro-clusters were generated in each of these windows using the DSC function generation, and stored once the sliding window passed over the 120^{th} data point relative to the starting data point.

For the damped time window model, micro-clusters were continuously generated after a set of input data points were damped due to the decay function of $\lambda = 0.033$. This model like the sliding time window model, was also implemented using the DSC function generation.

With a fixed $k' = 4$ for each time window (i.e. sliding and damped), once the data stream was finished and the micro-clusters stored, the online clustering was finished. The DSC function generation was again used for the computation of the macro-clusters in the Offline Macro-Clustering Phase. The current version of the DSC does not support the streaming elbow method yet, therefore, the optimal k value was computed separately using R, and later used as the input parameter for the DSC function.

Chapter 7: Discussion of the Results

It was determined that the most relevant variables in our analytical workflow to be steps and heart rate/minute (HR) due to these variables being the most accurate numerical values in the collected data streams. Heart rate variability (HRV) was also used as in an input for the streaming k-means algorithm. The initial $k_0 = 4$ micro-clusters results from clustering data points can be seen in Figure 9, using the first four sliding time windows of stream data, and the steps and HR variables for comparison.

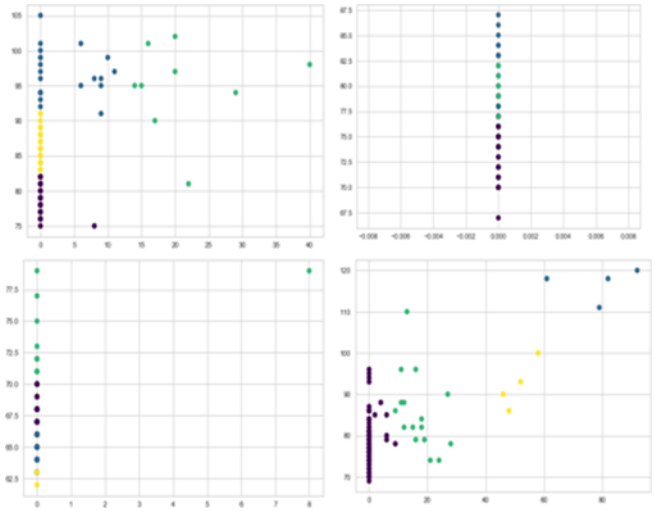


Figure 9: Initial micro-cluster results for the first four consecutive sliding time windows (x = steps, y = heart rate/min).

The evolution of the micro-clusters can be observed between these sliding time windows, in particular the linear and constant micro-cluster patterns when steps = 0 across a distributed range of HR values during a period of four hours. It is also possible to distinguish new random turning shape patterns that have occurred when steps >0, within

more specific ranges of HR values (e.g. from HR >90 to HR >100). These types of patterns have emerged throughout the time windows of several participants' data streams. For example, in the first and last sliding time windows, the new random turning shape patterns indicate movement patterns of a participant after a period of 4 hours staying still, rather than being outliers. These results provide empirical evidence that the variables should be targeted for influencing behavior change when devising interventions, and that the evolving patterns of actual novel micro-clusters represent a different context.

A selection of four participants are used here to illustrate the macro-clusters results and their respective self-quantified patterns that were found using both sliding and damped time window models. Table 5 provides an overview of their personal information.

Table 5: Participants in the experiment.

Participant	Description
12	59 year old female
18	30 year old female
19	28 year old male
20	60 year old female

The macro-clusters results of Participant 12 can be seen in Figure 10, where the red circles represent the centroids of the micro-clusters, and the blue crosses being the centroids of the macro-clusters, which were found using the sliding time window model. The centroids of the macro-clusters represent the self-quantified patterns, which reveal a balanced relationship between strong regular physical activity behavior that consists of no movement (i.e. steps = 0) with moderate regular physical activity behavior due to

mobility (i.e. steps between 10 to 40). Finally, it is also possible to visually identify the outliers by looking at the off-set centroids of the micro-clusters that have emerged from the data points.

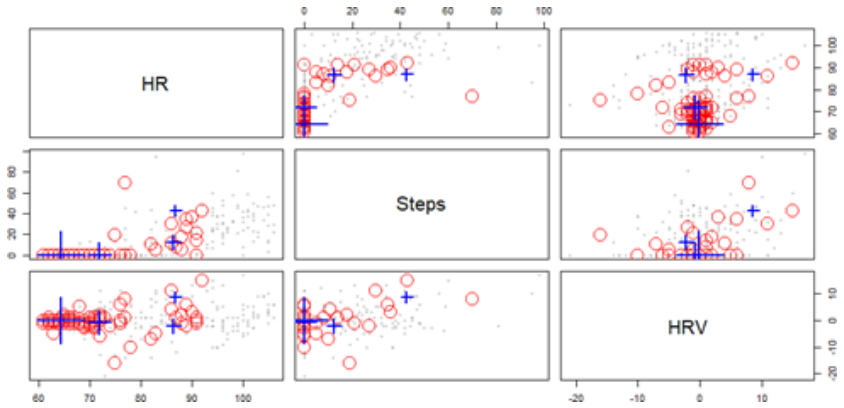


Figure 10: Macro-cluster (blue crosses) and micro-clusters (red circles) results for Participant 12 using the sliding time window model.

The moderate regular physical activity behavior of Participant 12 is more prominent in the macro-clusters results using the damped time window model (Figure 11). It is also interesting to point out that this participant has shown very few outliers: only one outlier micro-cluster in both time window models.

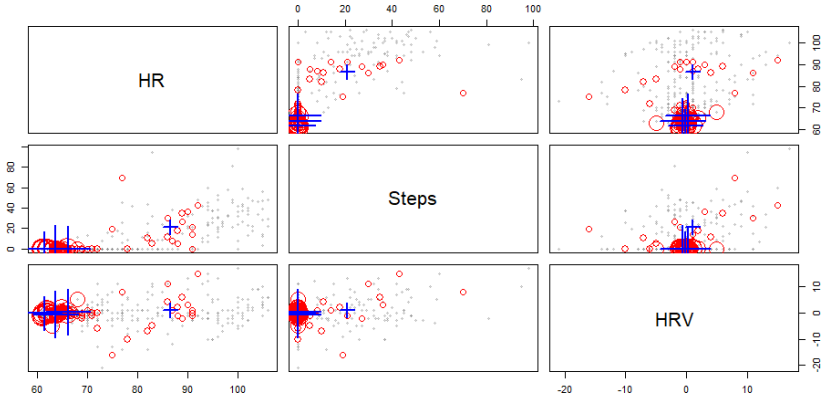


Figure 11: Macro-cluster (blue crosses) and micro-clusters (red circles) results for Participant 12 using the damped window model.

Table 6 provides statistics of the macro-clusters using some explanatory variables. Cluster 3 for example, makes up 79% of leisure time with an average step count of 11.44 steps per minute. This cluster represents time periods when the participant is active. Any new values entering this cluster will fall into the same class. We can also see that the data points belonging to this cluster fall largely on days 3 and 4 of the week (i.e. Tuesday and Wednesday) leading further insight into the participant's lifestyle, such as the participant become more active during these days.

Table 6: Explanatory variables for the macro-clusters found for Participant 12 using the sliding time window model. (Leisure/Sleep/Work time being a percentage of total time (1))

Cluster	Count	Average Heart Rate	Average Step Count	Average HRV	Majority Day	Least Day	Leisure Time	Sleep Time	Work Time
1	28009	77.13	8	0.01	5,6	7,1	0.43	0.41	0.16
2	25755	83.07	11.44	-0.02	3,4	2,7	0.79	0.1	0.11
3	18922	74.73	6.51	0.06	6,5	3,4	0.35	0.46	0.19
4	9885	73.18	5.9	0	1,7	3,4	0.32	0.54	0.15

Macro-clusters 3 and 1 are more generic in nature but still have unique labelling classes. For example, macro-cluster 3 represents the largest portion of work time compared to the other macro-clusters and macro-cluster 1 contains a mixture of values pertaining to the normal. Comparing these macro-clusters to macro-cluster 4, it is noted that this macro-cluster mainly represents 54% and 15% times that the participant is inactive with an average step count of 5.9 per minute, while maintaining a low average heart rate. The data points belonging to this macro-cluster fall largely on days 1 and 7

(i.e. Saturday and Sunday), which shows that the participant may be less active or sleeping more then on other days.

In comparison to the sliding window clusters, the damped window clusters offer a similar but different perspective on the participants' activity level as summarised in Table 7. We can see that the main difference lies in there now being two macro-clusters (1 and 3) that captured high physical activity instances, whereas sliding time windows stored these into just one macro-cluster. Although the count of data points per macro-clusters 1 and 3 are lower, they offer a diverse range of activity with high step counts and heart rate. Macro-cluster 4 offers a higher sleep classification rate, and again, day 7 is recorded as a day with a low amount of leisure time, populated mostly by sleeping time.

Table 7: Explanatory variables for the macro-clusters found for Participant 12 using the damped time window model. (Leisure/Sleep/Work time being a percentage of total time (1))

Cluster	Count	Average Heart Rate	Average Step Count	Average HRV	Majority Day	Least Day	Leisure Time	Sleep Time	Work Time
1	6743	82.87	11.26	0.09	4,6	5,7	0.77	0.11	0.12
2	40136	78.57	8.94	0	5,6	1,4	0.5	0.34	0.16
3	9109	82.96	11.05	-0.08	3,1	7,2	0.86	0.06	0.08
4	26583	74.1	6.21	0.06	1,5	3,4	0.33	0.49	0.18

The damped window model was particularly effective in revealing macro-clusters that could be associated to different physical activity intensity levels. One example was Participant 12 who exhibited the whole spectrum of physical activity intensity levels, ranging from very low (macro-cluster 1) and low (macro-cluster 2) intensity levels; up to high (macro-cluster 3) and very high (macro-cluster 4) intensity levels. Figure 12 illustrates the evolution of these intensity levels during the whole duration of the

experiment. It is important to point out that the highest intensity peaks have randomly occurred at any intensity level, showing a volitional regulatory behavior on different days of the week.

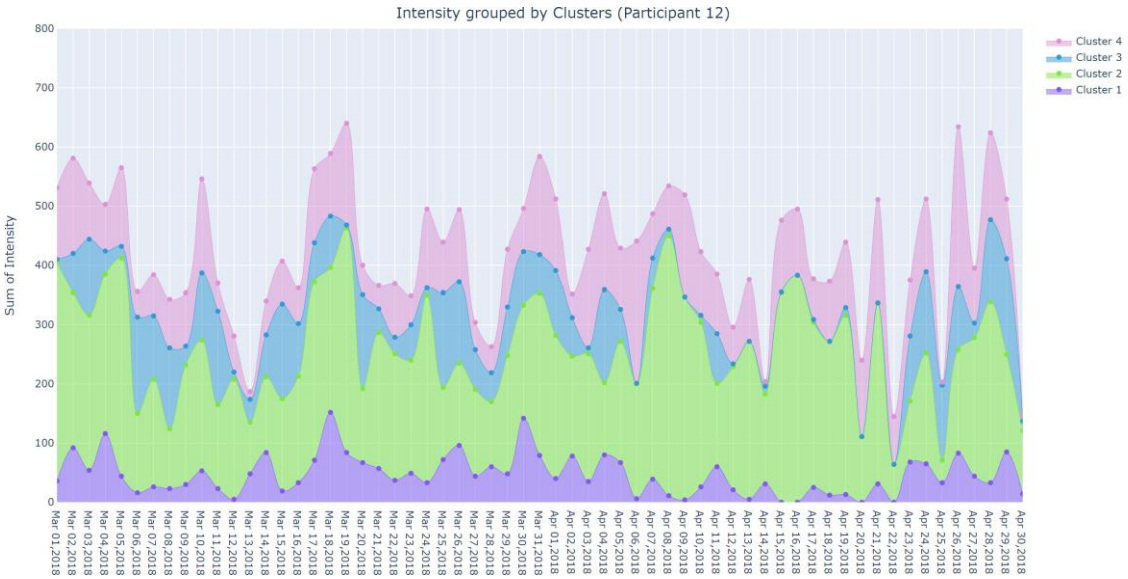


Figure 12: The evolution of intensity activity patterns of Participant 12 using the damped time window model.

A different evolution was observed for the intensity activity patterns of Participant 18 who exhibit few peaks of very low intensity activities in macro-cluster 1, as opposed to a wave pattern for macro-clusters 2, 3, and 4 that reveals intensity peaks that occurred after a wave has passed (Figure 13).

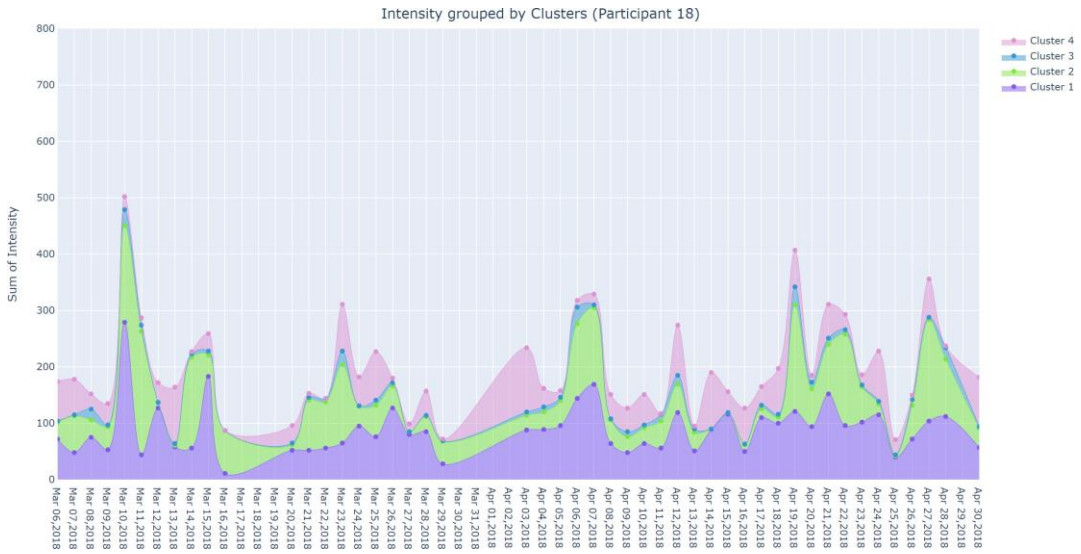


Figure 13: The evolution of intensity activity patterns of Participant 18 using the damped time window model.

We can see from Table 8 and 9 that participant 19 produced similar clustering results from both the sliding window and the damped window. Each window model produces one main physically active cluster, where the make-up is approximately 68% leisure, 14% sleep and 19% working time. This encompasses a large portion of high step count minutes and a higher than average heart rate. Similarly, another cluster results in most of the sleeping time with approximately 46% sleeping time, 44% leisure time and 10% working time. This cluster captures moments of lower step count, lower heart rate and during sleep. In certain instances, like in the case of participant 19 the time windows model will produce similar results due to low diversity in the movement and heart rate of the participant, which can be seen in Figure 14 and 15 (participant 19 damped and sliding plot from R).

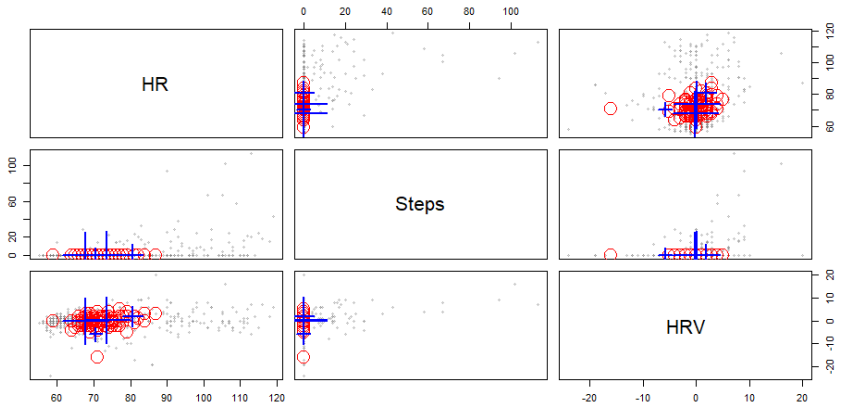


Figure 14: Macro-cluster (blue crosses) and micro-clusters (red circles) results for Participant 19 using the sliding time window model.

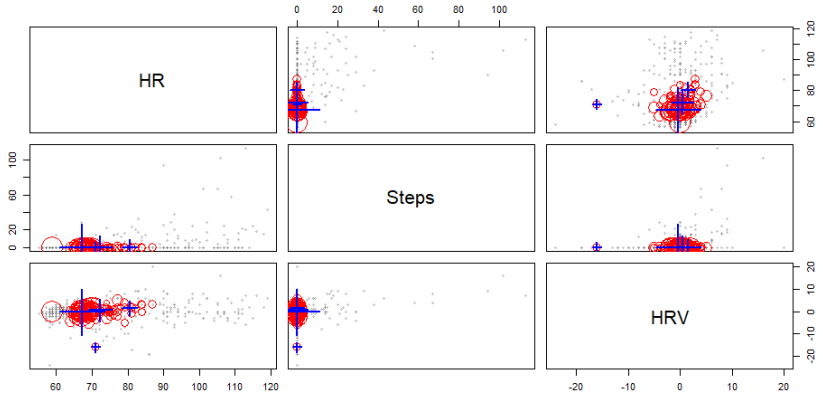


Figure 15: Macro-cluster (blue crosses) and micro-clusters (red circles) results for Participant 19 using the damped time window model.

There is also evidence that the participant had less data captured on days 3 and 4 of the week during the 2-month interval due to every cluster having day 3 and 4 as its least captured day. This could be due to user routine (example: taking off the Fitbit during weekly practices), or random errors (device/human). This case provides a good example of how low diversity datasets provide minimal changes to the results of the time window models.

Moreover, the time window models play an important role discovering self-quantified patterns labelled as regular physical mobility behavior because the actual steps trends during the weeks have been different from each other during the experiment. After analysing all the results of Participant 20, it was clear that the macro-clusters have exhibited regular physical mobility behavior using the sliding time window as shown in Figure 16. In this case, regular physical mobility was associated to the macro-cluster 1 on Monday (DOW=2); Tuesday (DOW=3) and Saturday (DOW=7).

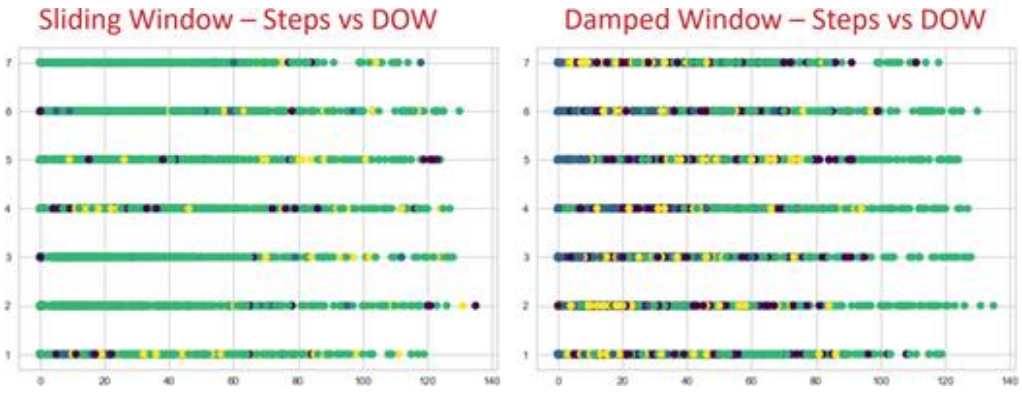


Figure 16: The weekly evolution of the macro-clusters according to the steps taken by Participant 20.

In contrast, the same findings were not found when analysing the macro-clusters using the damped time window model 15. In this case, the macro-clusters results reveal irregular physical activity throughout the various days of the week and macro-clusters. This exposes how challenging it is to differentiate regular from irregular physical activity behavior in self-quantified patterns due to the impact of a time window model being used to compute the macro-clusters.

Finally, the clustering results were visualised using density heat maps in order to compare the global self-quantified patterns amongst the participants. Figures 17, 18, 19, and 20 provide an overview of the variation of the number data points belonging to different macro-clusters of each participants when taking into account the relationship between the HR and steps variables.

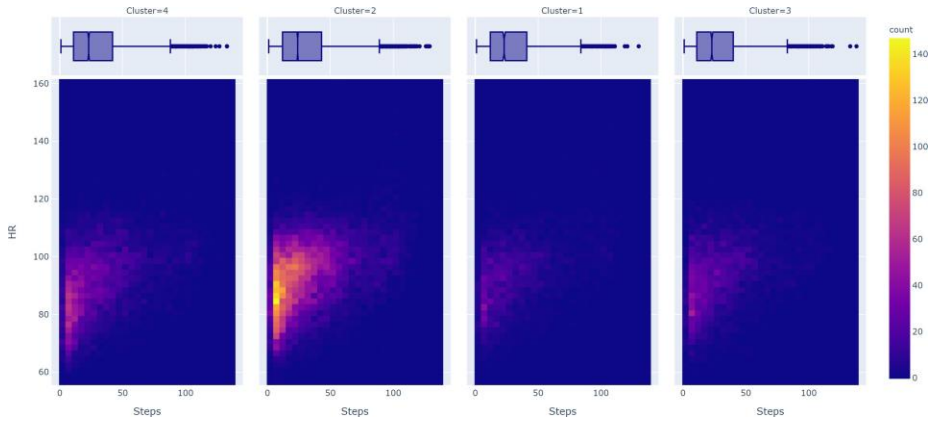


Figure 17: Density heat map for Participant 12.

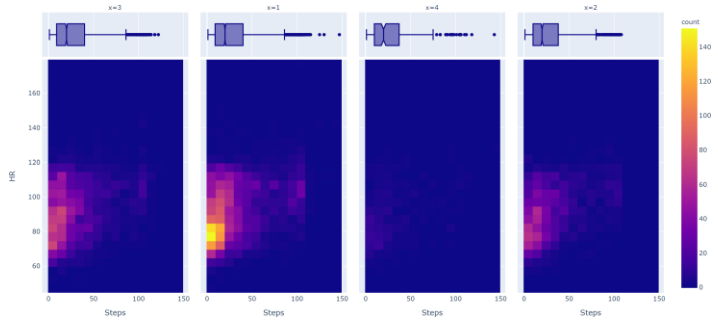


Figure 18: Density heat map for Participant 18.

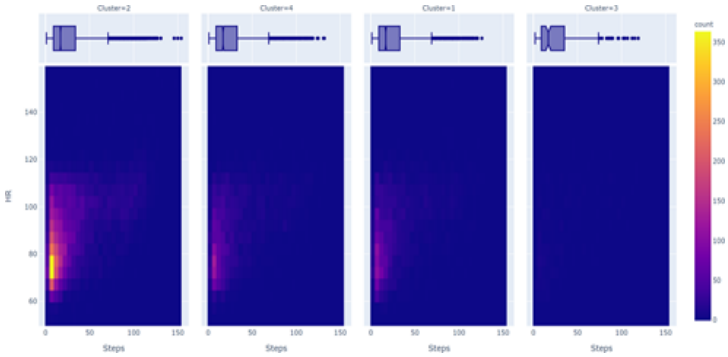


Figure 19: Density heat map for Participant 19.

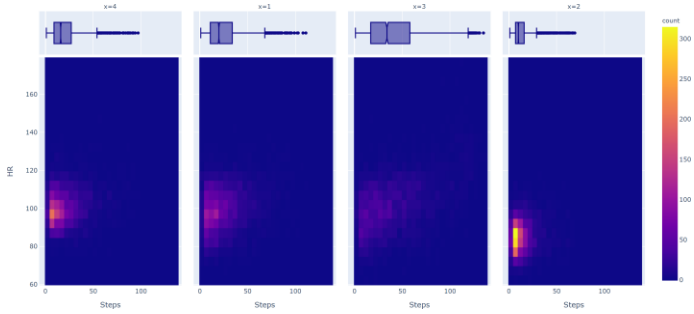


Figure 20: Density heat map for Participant 20.

Chapter 6: Conclusions and Future Work

A multi-window analytical workflow was proposed for improving the streaming k-means clustering algorithm by integrating complementary time window models such as the sliding and damped time window models. Our preliminary results demonstrate the impact they have on finding meaningful patterns and provide insight into classifying users activities based on just heart-rate, steps and heart rate variability, all attributes present in the majority of wearable devices currently on the market .Time window models have also not been researched exclusively, as they have been considered as a minor step in current research on stream clustering algorithms and therefore, they have not been explicitly understood in the required depth until now.

The time window models have also shown that aside from tackling the issue of concept drift, can also improve the results by providing sub-contextual information in snapshots of a user's total gathered data. In order to release the full power of stream clustering wearable data, time windows are a crucial aspect on the process and should continue to be used in conjunction with any stream clustering algorithm.

Outside of the impact that time windows produce on the resultant macro-clusters, we were able to distinguish what each cluster represented and the attributing factors on such. The key parameters used to determine the clusters were steps, heart rate and heart rate variability where the parameters sleep time, leisure time and work time were able to provide meaningful context on what the clusters represent. In our case, we were able to see that in a large case of the macro clusters favored certain activities (Sleep, leisure and work) over the other. These values could be further classified into these categories and

work with new incoming data streams to improve the results; which can be an area of further research.

Future research will explore other time window models (e.g. landmark and pyramidal time window models) coupled with the streaming k-means clustering algorithm in order to develop further our multi-window analytical workflow. For example, our landmark time window model will then be changed from time interval collection to “event” based collection, where data streams will be collected until a set “event” occurs. Initially we will start by setting the event as a drastic change in heart rate, this is dependent on the participant so will be adapted accordingly.

There is no time window that should be considered the most optimal for determining whether k-means is an accurate algorithm to use for both the online and offline phases. It is anticipated that there is to be a point where if the time frame of any type of time window model is too large, noise will always overcome the results and clusters will not be recognisable as self-quantified patterns as would be the case with using an entire dataset. After this point is found, we will be able to accurately explain any regular and irregular physical behavior. It will also be possible that different time window models will use different time frames within the same analytical workflow.

At this point using time windows in conjunction with a stream clustering algorithm has yet to be used as a focal point of the analysis and with our results we anticipate time windows and similar methodologies to be utilized in this pair both in real-time streams as well as in two-phase stream and offline streams in order to retrieve meaningful insight out of any clustering methodology.

References

Aggarwal, C., Han, J., & Wang, J. (2003). *A Framework for Clustering Evolving Data Streams.* IBM T. J. Watson Research Center &UIUC.

Aroganam, G., Manivannan, N., & Harrison, D. (2019). Review on Wearable Technology Sensors Used in Consumer Sport Applications. *Sensors(Basel)*.

Bifet, A., Holmes, G., Pfahringer, B., Kranen, P., & Kremer, H. (2010). Massive online analysis, a frame- work for stream classification and clustering. *Proceedings of the First Workshop on Applications of Pattern Analysis*, 44-50.

Bini, S., Shah, R., Bendich, I., Patterson, J., Hwang, K., & Zaid, M. (2019). Machine learning algorithms can use wearable sensor data to accurately predict six-week patient-reported outcome scores following joint replacement in a prospective trial. *The Journal of arthroplasty 34 (10)*, 2242–2247.

Cao, F., Estert, M., Qian, W., & Zhou, A. (2006). Density-based clustering over an evolving data stream with noise. *Proceedings of the 2006 SIAM international conference on data mining, SIAM*, 328-339.

Carnein, M., & Trautmann, H. (2019). Optimizing Data Stream Representation: An Extensive Survey. *Business & Information Systems Engineering 61 (3)*, 277-297.

Feehan, L., Geldman, J., Sayre, E., Park, C., & Ezzat, A. (2018). Accuracy of Fitbit Devices; Systematic review and narrative synthesis of quantitative. *JMIR mHealth & uHealth*.

Fitbit. (2020). Fitbit, How do i track my heart rate with my fitbit device? [Accessed on 2020-08-11]URL https://help.fitbit.com/articles/en_US/Help_ article/1565.htm.

Fitbit. (2020). *How do I track my heart rate with my Fitbit device?* Retrieved from Fitbit.com: https://help.fitbit.com/articles/en_US/Help_article/1565

Fitbit. (2020). *How do I track my sleep with my Fitbit device?* Retrieved from Fitbit.com : https://help.fitbit.com/articles/en_US/Help_article/1314

Fitbit. (2020). *How does my Fitbit device calculate my daily activity?* Retrieved from Fitbit.com: https://help.fitbit.com/articles/en_US/Help_article/1141#steps

Frey, A.-L., Karran, M., Jimenez, R. C., Baxter, J., Adeogun, M., Chan, D., . . . Hinds, C. (2019). Harnessing the potential of digital technologies for the early detection of neurodegenerative diseases . *10.31219/osf.io/u49z5*.

Ghesmoune, M., Lebbah, M., & Azzag, H. (2016). State-of-the-art on clustering data streams, . *Big Data Analytics 1 (1)* , 13.

Haghi, M., Thurow, K., & Stoll, R. (2017). Wearable devices in medical internet of things: scientific research and commercially available devices. *Healthcare informatics research 23 (1)*, 4-15.

Hahsler, M., Bolanos, M., Forrest, J., & al., e. (2017). Introduction to stream: An extensible framework for data stream clustering research with r. *Journal of Statistical Software 76 (14)*, 1–50.

Hancer, E., Xue, B., & Zhang, M. (2020). A survey on feature selection approaches for clustering. *Artificial Intelligence Review*, 1-27.

Hoogendoorn, M., & Funk, B. (2018). *Machine Learning for the Quantified Self.* Spriner.

Hu, R., van Velthoven, M., & Meinert, E. (2020). Perspectives of people who are overweight and obese on using wearable technology for weight management: systematic review. *JMIR mHealth and uHealth*.

Jain, A. (2010). Data Clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 651-666.

Jang, J., Oh, H., Lim, Y., & Cheung, K. (2020). J.-Y. Jang, H.-S. Oh, Y. Lim, Y. K. Cheung, Ensemble clustering for step data via binning. *Biometrics*.

Jin, X., & Han, J. (2010). *K-Means Clustering.* Encyclopedia of Machine Learning.

Jo, A., Coronel, B., Coakes, C. E., & Mainous III, A. G. (2019). *Is there a benefit to patients using wearable devices such as fitbit or health apps on mobiles? a systematic review.* The American journal of medicine 132 (12) 1394–1400.

Keogh, E., & Lin, J. (2005). Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and information systems 8 (2)*, 154-177.

Ketchen, D., & Shook, C. (1996). The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal 17 (6)* , 441-458.

Koolean, E., van Hees, H., & van Lummel, H. (2019). “can do” versus “do do”: a novel concept to better understand physi- cal functioning in patients with chronic obstructive pulmonary disease, . *Journal of clinical medicine 8 (3)* , 340.

Liu, S. (2019). Fitbit - statistics & facts. [Accessed on 2020-08-11] URL https://www.statista.com/topics/2595/fitbit/.

MacQueen, e. a. (1967). Some methods for classification and analysis of multivariate observation. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* , 281–297.

Mansalis, S., Ntoutsi, E., Pelekis, N., & Theodoridis, Y. (2018). An evaluation of data stream clustering algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 167-187.

Markets and Markets. (2020, Febuary 20). *Wearable Technology Market by Product (Wristwear, Headwear/Eyewear, Footwear, Neckwear, Bodywear), Type (Smart Textile, Non-Textile), Application (Consumer Electronics, Healthcare, Enterprise & Industrial), and Geography - Global Forecast to 2022.* Retrieved from Markets and Markets: https://www.marketsandmarkets.com/Market-Reports/wearable-electronics-market-983.html#:~:text=The%20overall%20wearable%20technology%20market,15.51%25%20between%202016%20to%202022.

Marutho, D., Handaka, S., Wijaya, E., & et. (2018). The determination of cluster number at k-mean using elbow method and purity evaluation on headline news. *2018 International Seminar on Application for Technology of Information and Communication, IEE*, 533-538.

Park, S., Lee, S., Han, S., & Cha, M. (2019). Clustering insomnia patterns by data from wearable devices: Algorithm development and validation study. *jMIR mHealth and uHealth*, e14473.

Peckham, J. (2019, July 01). *Fitbit Charge 2 Review*. Retrieved from Tech Radar: https://www.techradar.com/reviews/fitbit-charge-2-review#:~:text=The%20Fitbit%20Charge%202%20arrived,originally%20appeared%20early%20in%202015.

Piech, C. (2013). *K means, [Accessed on 2020-08-11]*. Retrieved from https://stanford.edu/~cpiech/cs221/handouts/ kmeans.html

Rabl, T., Sakr, S., & Hirzel, M. (2018). Big stream processing systems. *(dagstuhl seminar 17441), in: Dagstuhl Reports, Vol. 7, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik*.

Shah, Y., Dunn, J., Huebner, E., & Landry, S. (2017). Wearables data integration: Data-driven modeling to adjust for differences in jawbone and fitbit estimations of steps, calories, and resting heartrate. *Computers in Industry* , 72-81.

Silva, J., & Faria, E. (2013). *Data Stream Clustering: A Survey.* University of Porto.

Singh, A., Yadav, A., & Rana, A. (2013). *K-means with Three different Distance Metrics.* International Journal of Computer Applications .

Singh, R., & Bhatia, M. (2011). Data clustering with modified k- means algorithm, in: . *2011 International Conference on Recent Trends in Information Technology (ICRTIT), IEEE, ,* 717-721.

Steinhaus, H. (1956). Sur la division des corp materiels en parties. *Bull. Acad. Polon. Sci 1 (804) 801*.

Swan, M. (2013). he Quantified Self: Fundamental Disruption in Big Data Science and Biological Discovery. *Big Data*, 85-89.

Thorp, E. (1998). The invention of the first wearable computer, in: Digest of Papers. Second international symposium on wearable computers. *(Cat. No. 98EX215), IEEE*, 4-8.

Waheed, B. (2019). Utilization of wearable technology: A synthesis of literature review. *Tech. rep., EasyChair*.

Westenberg, J. (2016, October 18). *Fitbit Charge 2 vs Charge HR.* Retrieved from Android Authority: https://www.androidauthority.com/fitbit-charge-2-vs-charge-hr-718766/

Zaharia, M., Chen, A., & Davidson, A. (2018). Accelerating the machine learning lifecycle with mlflow. *IEEE Data Eng. Bull. 41 (4)* , 39-45.

Zuraini, H. H., Ismail, W., Hendradi, R., & Justitia, A. (2020). Students activity recognition by heart rate monitoring in classroom using k-means classification. *Journal of Information Systems Engineering and Business Intelligence 6* , 46–54.

Curriculum Vitae

Candidate's full name: Luke McCully

Universities attended (with dates and degrees obtained):

University of New Brunswick (Fredericton Campus), B.Sc.E. in Geodesy and Geomatics Engineering (2014-2018)

Publications:

Black, K.,McCully,L., Wachowicz, M.
(2017) Utilizing a Random Forest Classifier to Predict Falls Using Wearable Health Monitoring Devices. Fredericton, New Brunswick, Canada.

McCully,L. , Cao, H. ,Wachowicz, M.,
(2020)"Multi-Time Window Analytical Workflow for Clustering Wearable Data Streams" Journal: Big Data Research (under review)