Faculty of Engineering

Mathematical and Data Modelling 3

EMAT30005

Identifying Patient Subtypes in the Intensive Care

November 20, 2018

Authors:

Andrew Corrigan

Gilad Gur Harush

Tomos Morgan

Rizwan Shelim

Farrel Zulkarnaen

Supervisors:

Dr. Christopher McWilliams

Alex Church

Abstract

With the advancements in data science, many industries are now automating decision making processes through the use of machine learning algorithms. Similarly, the following report investigates the application of such methods in a medical environment. From an unsupervised learning approach, we seek to partition patient data from intensive care units into clusters (subtypes) based on similar health features, so that medical professionals can determine suitable personalised treatments. By specifically using data from intensive care patients, we can gather more data due to the patients being monitored more frequently across more medical measurements. To combat the heterogeneous nature of the patient population, we identified patterns in high dimensional data sets by using dimensionality reduction algorithms. We then applied k-means clustering algorithm and used statistical measures on the distribution of features in each cluster to determine meaningful variation. As a result, we found that one of the clusters exhibit extreme signs of liver damage and greater probability of dying in hospital. As such, doctors should be able to react faster and prevent deaths by looking at the signs that have been examined by the algorithms. Similar approaches could be implemented to find other patterns in health records, hopefully resulting in a more general data-driven procedure for deciding appropriate medical interventions. We hope that this will improve the reliability of precision medicine and reduce the workload for medical professionals.

Contents

1 Introduction

The medical process has traditionally been one determined through the training and decisions of the individual medical professional, with treatments often not being tailored to the individual needs of the patients. Doctors typically treat specific patients with medicine tested on the average patient through a series of medical trials [1] and previous experience. Whilst generalising patients like this may be a good use of statistics it does not respond well to extreme cases such as rare diseases or patients whose health deteriorates rapidly. This method also is particularly sensitive to levels of experience between individual clinicians.

Recently machine learning approaches have been utilised in collaboration with medical professionals to improve medical diagnosis in a range of applications [2]. We propose a modern approach to combat this problem through pattern recognition to subtype patients. The main difference and advantage compared to traditional diagnosis are that subtyping takes into account the conditional distribution of patients within a smaller subset, instead of marginalising patients from the aggregate population. By automating medical prognoses based on similar individuals like this, patient subtyping is theorised to help reduce the uncertainty of an individual's response to medical treatment. This is done by basing the treatment on the patient's subtype's reaction to a similar treatment [3]. Subtyping of patients may also help highlight which patients are more at risk than others.

This model should also result in a reduction of type I (false positive) and type II (false negative) diagnosis errors, lowering health-care cost, and providing a quicker healing process for patients. The ultimate aim of the report is to establish if patients can be safely and effectively subtyped quickly through the methods proposed so that patients can receive the treatment they require as soon as possible, whilst stating the significance of those subtypes on the medical trajectory of a patient.

2 Background

The applications of machine learning in the medical domain has recently become a huge area of growth. Companies such as IBM Watson Genomics have made key developments in the machine learning approach to disease identification and diagnosis [4]. One area that specifically applies to the identification of patient subtypes is through the work done on personalised treatment. This has been applied in real-world situations such as IBM Watson's Oncology system of supervised learning, where the doctor would input the data collected from the patient and consequently be recommended suitable treatments for patients.

Whilst not in practice there have also been several additional studies that compare more directly to the aims of this project. One such example is the report Identifying Distinct subgroups of Intensive Care Unit Patients [5]. Here researchers identified distinct subgroups of intensive care patients. They utilised an unsupervised learning method with clustering analysis to identify six distinct subtypes by splitting the features into four separate domains. These subgroups were the following:

1. Relatively healthy and short-stayed ICU patients
2. Older patients with catastrophic critical illness
3. Post surgical or procedural ICU patients
4. Older ICU patients discharged with long-term care plan
5. Previously healthy patients with prolonged ICU course and good recovery
6. Elderly patient with severe illness and history of life-threatening problems

This differs from the supervised learning approaches used by IBM Watson. Another interesting area of development with the application of machine learning techniques to the medical domain has been the use of neural networks. Examples include the use of Recurrent Neural Networks and a proposed new form Long-Short Term Memory Networks (LSTM) to acquaint for the sequential time-series nature of health records [5].

3 Algorithms

To solve our clustering problem we have chosen three machine learning algorithms that we determined to be relevant and useful to our cause. Two of these are dimensionality reduction algorithms, and the other is a clustering algorithm which was deemed suitable.

Dimensionality reduction is crucial for handling patient's data due to the tendency for recorded health records to be in high-dimensional spaces. Therefore, a lot of our analysis relies heavily on reducing dimensions such that we may ease computational analysis whilst still maintaining as much information.

3.1 t-Distributed Stochastic Neighbour Embedding

This machine learning visualisation method, widely used in many fields [7], is a non-linear dimension reduction algorithm. The benefits of using this algorithm are that it handles the 'Crowding Problem' well and its use of stochastic neighbours. The crowding problem arises from the 'Curse of Dimensionality', where points of similar distances in high dimensions (more than 3D) are mapped to be squashed together in lower dimensions (2D or 3D) [8]. t-SNE prevents crowding by making the optimisation spread out the distances. The stochastic neighbours allow the algorithm to take into account both the local and the global structure of the data by focusing on the local structure without completely ignoring data points that are far away.

t-SNE also has its limitations, one of which being that it is difficult to optimise. This is due to non-convexity (has multiple local minima) and consequently, causes the algorithm to be non-deterministic. Another limitation of this algorithm is in the assumptions made whilst calculating. t-SNE assumes that the local structure of a Manifold (a topological space) is linear and therefore Euclidean distance as the distance metric used between neighbouring points. This assumption affects the output of the algorithm when the Manifold used is increasingly complex in its local structure. Due to the algorithms heavy reliance on calculating the Euclidean distance, the computational cost increases with the number of dimensions in the full dataset. This is partly due to the algorithms time complexity being of order n^2. This algorithm would therefore work well for a number of patient data, but suffer if there are too many variables. Its output is also non-invertible, meaning we cannot re-obtain the original data from our output and as such is only useful in visualising trends in the data. Below is an overview of the algorithm:

1. The algorithm creates a probability distribution that dictates the relationships between neighbouring points in the high/full dimensional space. The probability distribution follows that of a Gaussian. For a point x_i the probability of picking a point x_j as its neighbour is

$$p_{ij} = \frac{\exp(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2)}{\sum_{k \neq l} \exp(-\left\|x_k - x_l\right\|^2 / 2\sigma_i^2)}. \tag{1}$$

   The probability is proportionate to the probability density of a Gaussian centred at x_i, meaning the probability deteriorates quickly for points that are far away, but never reaches 0.

2. The algorithm then creates a low dimensional space that follows the previous probability distribution as much as possible. This step is uses a Student t-distribution with a single degree of freedom

$$q_{ij} = \frac{(1 + \left\|y_i - y_j\right\|^2)^{-1}}{\sum_{k \neq l} (1 + \left\|y_k - y_l\right\|^2)^{-1}} \tag{2}$$

   where y_i is the coordinate in the embedding space. The optimisation of the distribution used in this step is done by gradient descent on the Kullback-Leibler-divergence (J) between the Gaussian distribution and the Student t-distribution, expressed as

$$\frac{\delta J}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \left\|y_i - y_j\right\|^2)^{-1}. \tag{3}$$

   The sign of the gradient represents an attraction (+) or repulsion (-) between particular points. It is this attraction and repulsion that determines where data points settle in lower dimensions.

3.2 Principle Component Analysis

Let us remind ourselves that one feature represents one-dimensional axis; however, not all features may be relevant. If two or more features are correlated, this implies that knowing one over the other does not provide more information as they are both dependent, rendering one of them redundant. PCA (Principle Component Analysis) simplifies the data by choosing a subset of 'independent' features and projecting them to a new set of axes called the 'principal axes.' The effectiveness of PCA is that it reduces the dimension whilst still retaining as much variation present in the dataset, thereby minimising loss of information. PCA does so by taking into account the variances and covariances along each feature dimension. Note that given N number of observations in the dataset (represented as vector points) and d number of feature dimensions in each observation, the shape and orientation of the data are derived from its covariance matrix $\mathbf{C}$ (See Appendix, Equation 11).

Let our dataset of 4,000 patients be a set of points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, such that each vector $\mathbf{x}_i$ represents a vector to a point in d dimensions where $N = 4,000$ and $d = 37$, for i from 1 to N. Also, in vector notation, $\mathbf{x}_i = (x_{i_1}, \ldots, x_{i_d})^T$ where x_{i_k} corresponds to the k^{th} component of the i^{th} vector point, for all k from 1 to d. PCA then reduces the dimension of each $\mathbf{x}_i$ to a new size p where p is smaller than d.

First, PCA calculates the eigenvalues and eigenvectors of $\mathbf{C}$. The independent features that represent the most variance are the highest eigenvalues, and their respective eigenvectors represent the new principal axes called Principal Components (PC) for the reduced dimensional set of the data. PCA sets a new covariance matrix, outlined in Equation 12, such that it becomes a diagonal matrix with the eigenvalues along the main diagonal, and zero covariances to ensure complete independence.

Notice how in Equation 12, the diagonals are eigenvalues of the original covariance matrix ordered from highest to lowest. λ_1 represents the highest variance in the data, λ_2 is the next highest variance orthogonal to λ_1 and so on. The orthogonality ensures that their corresponding unit eigenvectors can represent the direction for a new set of axes. Also notice how now the index of λ ends at p. This encapsulates how PCA has chosen the most relevant features and only saves the features with relative high variance. Typically this is done by setting a benchmark such that the relative percentage of the sum of chosen eigenvalues are above a threshold value normally in the range of 65% to 70%.

This is illustrated in Equation 4.

$$\frac{\sum_{k=1}^{p} \lambda_k}{\sum_{k=1}^{d} \lambda_k} \geq 65\% \tag{4}$$

Third, since we now have a new covariance matrix with no correlation and minimum features, PCA then projects all the vector points into a line that runs through the vector mean $\boldsymbol{\mu}$, in the direction of the chosen unit eigenvectors $\hat{\mathbf{u}_k}$ given by:

$$\mathbf{x}_i^{'} = \boldsymbol{\mu} + \sum_{k=1}^{p} \alpha_k \hat{\mathbf{u}_k} \tag{5}$$

Generally, there are two kinds of dimensionality reduction algorithms, Feature Selection and Feature Extraction - PCA belongs to the latter. Compared to a Feature Selection method whereby individuals need to manually decide on the subset of relevant features, Feature Extraction allows a more robust selection by transforming the raw dataset into a new feature space that is computationally easier while also maintaining as much information as possible. In Feature Selection, since we deliberately discard some features there is a chance that we will lose information, but for Feature Extraction methods like PCA, all features in the feature space are marginalised which linearly transforms the dataset to a more compact subspace. The PCA model is suitable as it performs well for higher dimensional data and does not jeopardise crucial information for sensitive data such as health records. Despite this, one limitation of PCA is that it is a linear feature extraction, this means that it only captures linear correlation in our data.

3.3 k-Means Clustering

To visualise the clusters that represent our patient subtypes, we first chose a deterministic model as groundwork for our observation, the k-means clustering algorithm.

The k-means clustering method is an unsupervised learning algorithm that seeks to find the best partitions for N number of observations into distinct k number of clusters. In doing so k-means outputs k number

of centroids (vector means for each k cluster), and the Squared Sums of within-cluster point-to-centroid Distances (SSD) where these two variables will be the basis for k-means' partition. Specifically, the SSD is computed by a Euclidean distance given by Equation 6

$$SSD = \sqrt{\sum_{i,j} \left(\mathbf{x}_{i_j} - \boldsymbol{\mu}_j\right)^2} = \sum_{i,j} \left\|\mathbf{x}_{i_j} - \boldsymbol{\mu}_j\right\| \tag{6}$$

where $\mathbf{x}_{i_j}$ is the i^{th} data point from the j^{th} cluster, and $\boldsymbol{\mu}_j$ is the centroid from the j^{th} cluster from the set of centroids given by k-means.

The k-means algorithm runs on an iterative refinement technique with 3 steps: initialisation, cluster assignment, and update of centroids.

The algorithm initialises k number of initial cluster centroids i.e $\{\boldsymbol{\mu}_1 \dots \boldsymbol{\mu}_K\}$. Then the algorithm assigns every data point to the nearest initial centroid with the smallest SSD then recomputes and updates the value of the nearest mean as the centroids for each cluster. The algorithm repeats the procedure until there is no change in the values of the centroids.

As k-means runs to find the optimum configuration that minimises the SSD, it also maximises the distance between clusters themselves, resulting in a clearer partition. Therefore, a good initialisation method is key to reducing the time computational complexity of the algorithm, we can do so by choosing the k-means++ initialisation method. k-means++ selects the initial centroids from the points in the dataset and applies different weightings to them based on the distance between each point to its nearest selected centroid. This allows the centroids to converge much faster to the optimum centroids and minimise computational work which makes it the preferred algorithm for centroids initialisation. As a result, k-means is able to converge to find a local optimum despite its NP-hard time complexity.

To summarise, below is a pseudo code outlining the k-means algorithm process.

Algorithm 1 k-Means Clustering

```
procedure CLUSTERING
    function KMeans(Instances, K)
    randomly initialise k vectors with μ1 ... μk;
    repeat
        assign each x ∈ Instances to the nearest μj;
        recompute each as μj the mean of the instances assigned to it;
    until no change in μ1 ... μk; return μ1 ... μk
```

4 Data handling

Before any statistical analysis can be shown, some terms and parameters must be defined:

- *SAPS-I scale*: this is the Simplified Acute Physiology Score used to measure the severity of a person's conditions - a higher number translates to a greater severity [9].
- *SOFA score*: the Sequential Organ Failure Assessment score is used to track the working status of an ICU patient's organs [10].
- *Length of stay*: this is defined as the number of days between ICU admission and the end of hospitalisation.
- *Survival*: this takes a value of NaN if the patient is alive at the time of data collection and, if the patient didn't make it, a positive value corresponding to the number of days between ICU admission and death.

The data is originally stored in the form of 4,000 text files. Each file corresponds to one patient and contains time-stamped CSV data of 37 variables (outlined in Appendix 6.2), taken over the first 48 hours of their

hospital stay. Through reading other approaches to the PhysioNet 2012 challenge [11], we thought it best to create a single, large feature matrix. This feature matrix could then be manipulated and analysed more easily, with all necessary data in that one matrix. Patients' data was represented in the rows, and the wide range of features that had to be analysed were formatted as the columns. In order to initialise the algorithms (PCA and t-SNE), the mean and standard deviation of each medical parameter recorded (listed in 6.2) was used. Our reasoning for choosing these two statistical descriptors is as follows; a higher-than-average heart rate, for example, would indicate a problem and therefore, a higher likelihood of death. Although we speak as data scientists and not medical experts, we feel the same would apply to the means of all parameters, as well as their variances.

4.1 Imputation

The data provided contained many discrepancies and therefore a method to handle corrupted data was needed [12]. t-SNE and PCA prefer different imputation methods; t-SNE works best with mean-imputation - where NaN values are replaced by the mean of the viable data as to not alter the range and distribution of the data; PCA conversely works better when NaN values used for corrupt/erroneous entries because mean-imputation reduces variance and would skew the spread too much in our case.

4.2 Normalisation

In the initial phases of research, much time was spent scrutinising samples of the data within a small script to better understand our dataset and how to best manipulate it. Through doing this we found that the variables' distributions, and more importantly their ranges, had a large degree of variation. This raised an issue as our classification and dimension-reducing algorithms are made less robust by dissimilar ranges and distributions. Consequently, we wrote a script to normalise every variable appropriately to have a zero mean and unity variance, applying the natural logarithm function for variables whose 1st percentile is over two orders-of-magnitude smaller than the 99th percentile. We used these tail-end values instead of the inbuilt max and min functions to avoid the effects of outliers. We adopted the natural logarithm function because the common long-tailed distribution seen in many variables resembles a log-normal shape.

5 Analysis

The data, now taken prior to any dimension-reduction transformations, was parsed through the k-means clustering algorithm to make it easier to analyse and subtype. However, first we were to determine the optimal number of clusters, i.e. the possible number of patient subtypes present in the data.

To validate our value for k we explore the relationship between the sum of squared distances (SSD) of every point to its associated centroid for varying values of k; this is known as the Elbow Method. A graphical representation can be seen in Appendix B, Figure 6. Since the k-means algorithm seeks to partition the dataset by minimising the SSD, one should choose k such $k + 1$ does not provide a significant marginal increase in information. This is depicted at the instance of a gradient increase. As seen in Figure 6, the first gradient change is at $k = 2$; this implies that a possible candidate for the number of subtypes is two. However, we should note that since the SSD is a Euclidean Distance measure, which has signs of inaccuracy in higher dimensions [14], there is still a possibility of losing information if we just choose $k = 2$. Therefore, we will first analyse our results for $k = 2$, and inspect the performance of the clustering. We will then look at higher values of k to try and detect any significant information that could not be found at $k = 2$, and compare and contrast the results.

5.1 Two Clusters

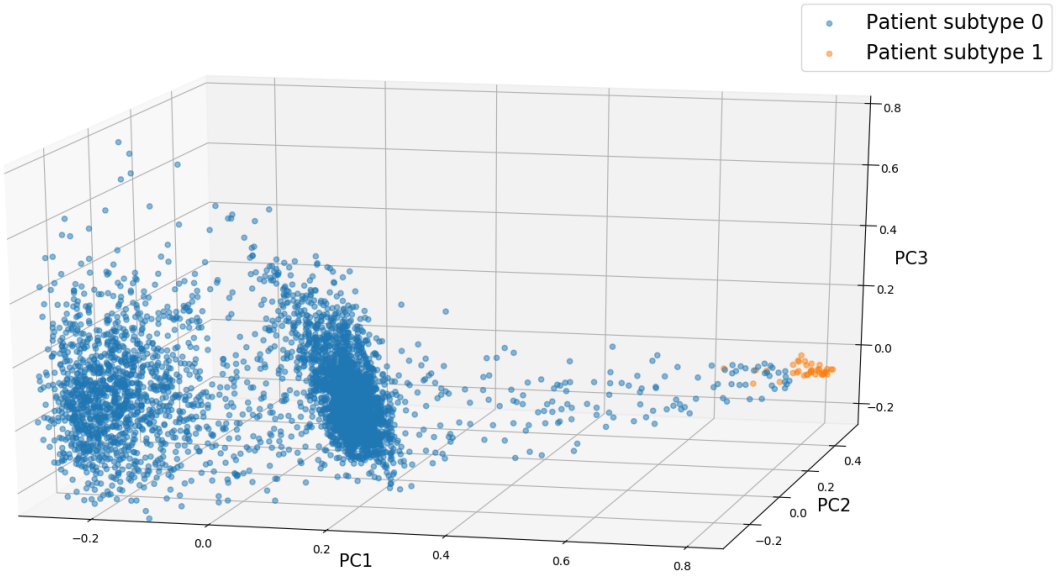
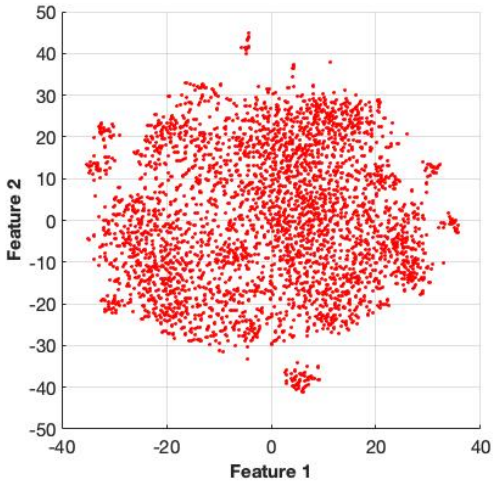


Figure 1: Output of PCA, with $k = 2$. Each cluster depicts the proposed patient subtypes.

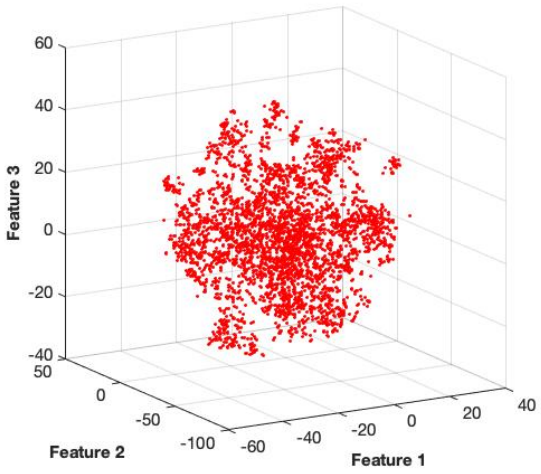
Referring to Figure 1, the two detected clusters are unevenly populated. Looking closely, there seem to be two large groups categorised to one subtype despite a clear separation in the -0.2 to 0.2 range in the PC1 direction. From this, we can infer that the difference between Subtypes 0 and 1 (using the Euclidean distance metric regarding all 37 variables) is greater than the difference between the two apparent sub-clusters within Subtype 0. This implies that at $k = 2$, k-means fails to detect the variation within Subtype 0 to notice those two sub-clusters. To compensate, we increased the value of k until we can detect the separation in Subtype 0 and examine any meaningful information that could not be detected before. We found that in the three principal components from PCA, $k = 5$ manages to detect the sub-clusters in Subtype 0.

5.2 Five Clusters

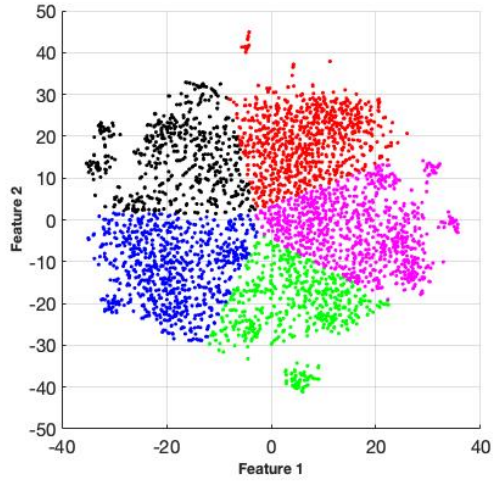
Our initial step in looking at five-cluster separation was to visually inspect it after projection onto two dimensions using t-SNE. As explained in section 3.1, t-SNE is a non-linear technique and so any analysis using this method utilised the k-means function *after* projection. This is because t-SNE relies heavily on the Euclidean distance metric which is known to suffer from inaccuracy and errors when undergoing non-linear transformations. With the graphical output below, this algorithm projects the data to have roughly equal ranges and distributions in both feature axes.



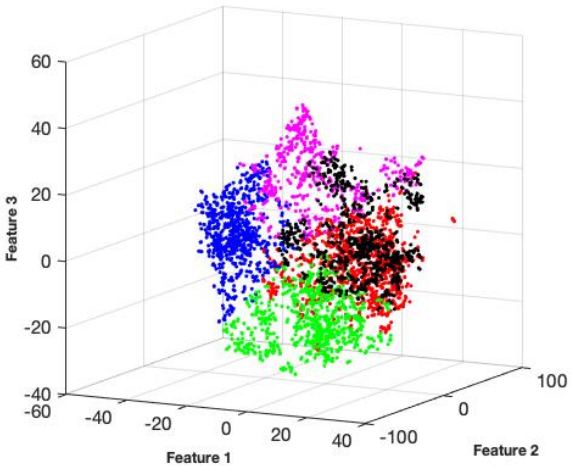
(a) Scattering of the first 4,000 patients.



(b) t=3D scattering of the first 4,000 patients.



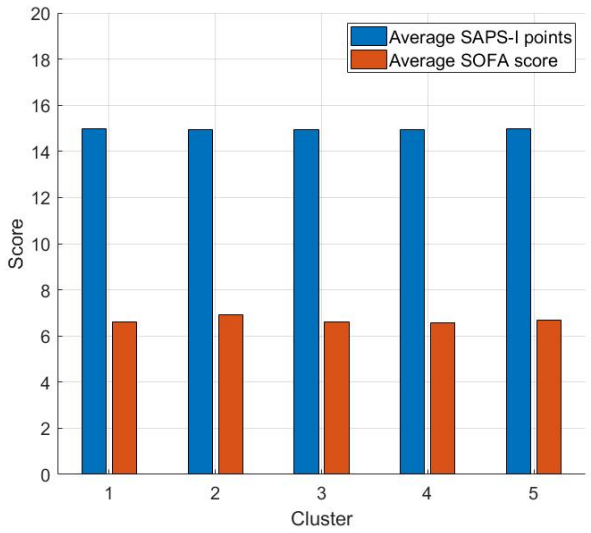
(c) k-means clusters in 2D.



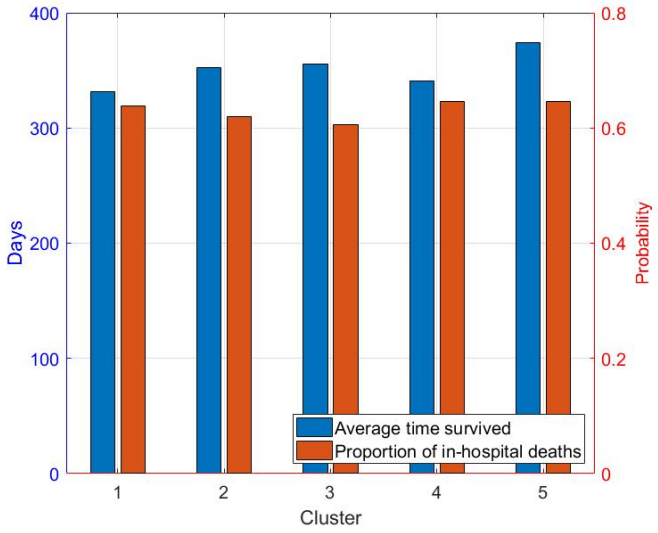
(d) k-means clusters in 3D.

Figure 2: Graphical outputs, in 2D and 3D, of the t-SNE procedure. Shown are the raw outputs and the grouped outputs.

Figure 3b reveals the data to not hold meaningful information with respect to the 'Survival' parameter or any general descriptor (see Appendix Figure 14 and Figure 3b). More medically relevant variables, such as SAPS-I and SOFA scores, were also investigated (Figure 3a). The lack of variation in these variables should be a clear sign of redundancy - the likelihood of death, for all clusters, fluctuates around 65%. The mean amount of time survived after hospitalisation, the SOFA score, and SAPS-I points also don't vary with statistical significance. This was put down to the t-SNE algorithm's non-linearity causing the data to lose any linear correlation it had, resulting in k-means only finding that purely geometric, radially symmetric clustering. A different visualisation technique was adopted to find superior results - namely PCA.



(a) Bar chart depicting the SAPS-I and SOFA scales across five clusters as deemed by k-means on the projected data.



(b) Bar chart plotting the average *Survival*, as defined in Section 4, and proportion of patients that have died, for the five clustered based on the t-SNE projection.

Using PCA and plotting the k-means result for $k = 5$, shown in Figure 4 below, there are two points we can infer from visually inspection. First, notice that the two most blatant out of the five clusters, which were previously held as one subtype when we set $k = 2$, are now differentiated as the blue and purple cluster (Subtypes 0 and 4 respectively). Compared to the other three subtypes, these two are considerably large, meaning that the majority of the patients in the population are in one or the other. Second, as a consequence of the previous statement, the other three clusters are concentrated together on the far right of Figure 4. At first glance, this might seem inaccurate since k-means seeks to maximise the distance between clusters for clearer partition. However, since we applied k-means before dimensionality reduction, this implies that there actually is a difference between those clusters; something we cannot see in three dimensions (and in lower values of k).

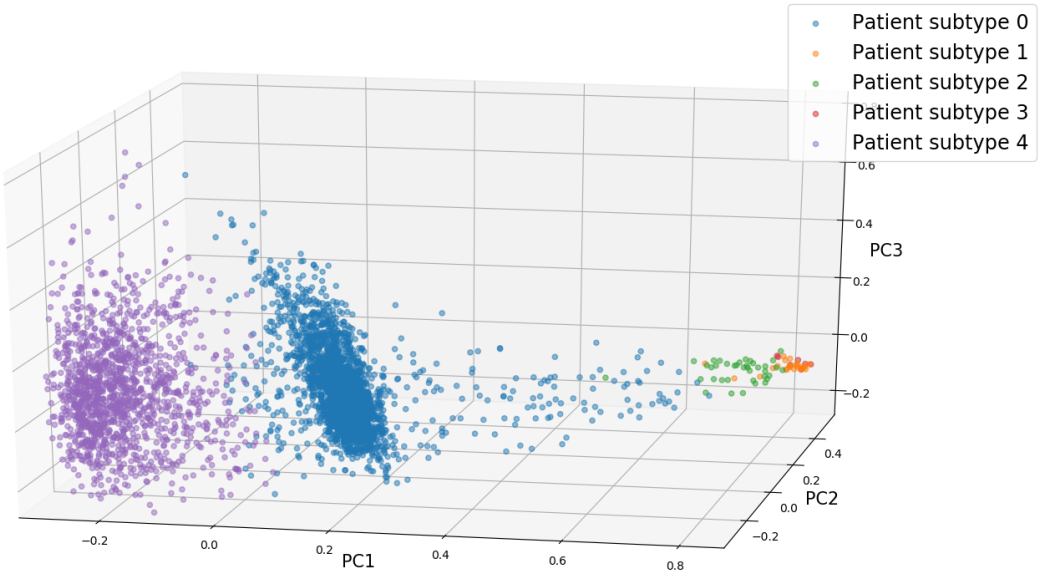


Figure 4: Output using k-means to colour-map the PCA output, with $k = 5$. Here the clusters are more intuitive and are likely to yield more useful results.

To further inspect the differences between the five subtypes, we recorded the location of the centroids for each cluster in 37 dimensions. Note that each cluster's centroid is a coordinate with 37 components ('directions'), where each component represents the mean of said feature within said cluster. This is illustrated below in

Table 1.

Subtype	Age	Height	Weight	ALT	AST	Glucose
0	65	170	82	179	252	136
1	58	167	81	3018	5121	164
2	54	178	81	1708	1819	146
3	33	172	97	6652	8822	179
4	64	170	84	34	47	137

Table 1: Table of cluster centroids showing six example feature components, where each component is the average feature values for each cluster. This information is fully displayed in Appendix Table 2.

Notice how the sample means of most features for both patient Subtype 0 and Subtype 4 are almost identical. However, a notable difference between them is the sample means of their levels of ALT and AST; these concentrations are higher in Subtype 0 than in 4. This is one possible explanation as to why there exists a separation between the two subtypes. This is also explains why in Figure 4, the highest variation in data is along the PC1 direction. Secondly, corroborated by Figure 4 and Table 2, we can see that Subtype 3 is significantly different from the rest. The features that heavily define Subtype 3 are AST, ALT, ALP, and Bilirubin - all of these possess values much higher than other subtypes'. These parameters all measure concentrations of key enzymes in hepatology, hinting at a liver-related cause for their ICU admission. You can also see that the height and weight of patients in Subtype 3 likely contribute to their separation. According to the NHS weight chart [15], given the average height and weight for patients in Subtype 3, they are likely to be obese.

From Figure 7, the two highest-ranking variances in our dataset are those along the AST and ALT features, which both account for 93% of the total variance in our data. Hence, showing why these two features can be used as a suitable marker for identifying patient subtypes in our data. Refer to Figure 8a and Figure 10a to see how ALT and AST levels vary for patients in each subtypes. Also from Figures 8 and 11a, by using a Kernel Density Estimation (which is a non-parametric measure to estimate the maximum likelihood of a random variable's probability density distribution) we can see that each subtype is approximately Gaussian, but subtype 3 corresponds to a greater variance and mean, whilst the other four subtypes are centred around approximately the same mean and with little standard deviation.

Furthermore, from looking at Figure 12 and Figure 13, we can see the subtle difference between all subtypes based on the outcomes of their intensive care admissions. On average, patients in Subtype 3 (which has the highest signs ALT and AST) have less than 10 days survivng after discharge which corroborates to the fact this is the subtype with highest elevated of ALT and AST. As a consequence, Subtype 3 also have the lowest average of length of stay in the hospital with the highest probability of dying in admission, where 43% of patients in subtype dying in the ICU.

To summarise, the 5 different subtypes and their characteristics are the following:

- Subtype 0 - Average aged patients with normal level of health records.
- Subtype 1 - Average aged patients likely to develop liver-related diseases. Longest admission time, but least likely to survive after discharge.
- Subtype 2 - Average aged patients with of signs liver-related problems.
- Subtype 3 - Younger patients with signs of obesity and liver damage. Highest chance of dying in admission and lowest survival rate.
- Subtype 4 - Average aged patients with inadequate level of AST and ALT. More likely to die in hospital compared to Subtype 0.

Whereas for $k = 2$, the two subtypes are categories by the following discribtions.

- Subtype 0 - Average aged patients with normal level of health records.
- Subtype 1 - Younger aged patients with signs of obesity, liver damage and 40% more likely to die in admission compared to Subtype 0

5.3 Effects of time limitation

Through communications with a BRI research associate, we gathered that doctors are looking for ways to extract accurate information using as little time-series data as possible. The motivation for this is so that they can act sooner when making medical decisions. As a result, we looked at the difference between the case where we regard the first few hours of data and the case where we look at the entire dataset. The figure below is a showcase of how the data retains its form despite a time limit. This validation is necessary for our methods of data normalisation for time-bound data; a crucial step in analysing a dataset. If this were not the case then doctors applying our research would require a similar number of hours for our method's results to be significant as to allow the data to take a similar form to what we see at 48 hours - this is not ideal in an intensive care environment.

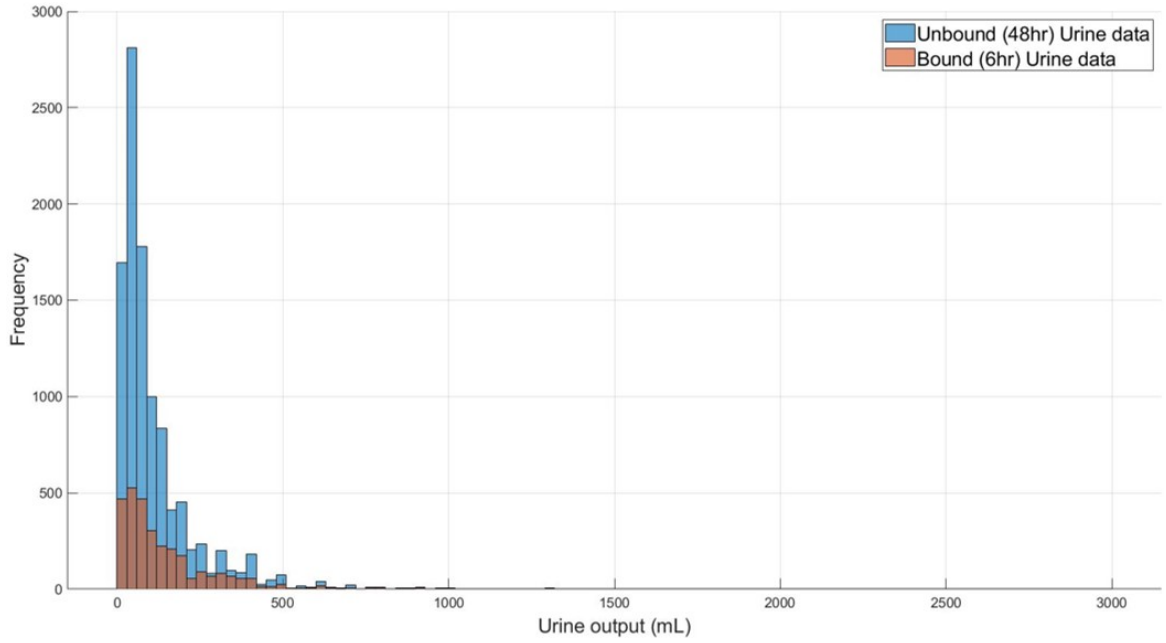


Figure 5: Stacked (not segmented) histograms showing that the distribution shape becomes apparent even within the first six hours of patients' stays.

6 Discussion

Having attempted both PCA and t-SNE methods to reduce the dimensionality of the feature we can draw conclusions as to what can be learnt from the methods and more importantly how useful the methods are at subtyping the patients. We found t-SNE far less effective at reducing the dimensionality of the data as it is a non-linear process. This meant that when the data was reduced to two features a large amount of information was lost. This can be seen in Figure 2d. This shows how there is little to no separation between the clusters identified. This differs greatly from the clusters found using PCA methods. There is clear separation as seen in Figure 4. From this we can infer than most relationships within the data are linear as they are best represented in two dimensions by a linear transformation. In addition to this as the PCA is a linear method so the clusters can be found *before* the dimensionality is reduced meaning even less information is lost and a clearer result can be obtained.

Our methods do also have some associated limitations. The one key issue is the frequency of NaN values with the data set. Every single patient record has at least one missing entry, almost certainly as a result of human error. This means that a considerable amount of the data has been imputed thus reducing the accuracy of the results.

6.1 Future Works

The next step for the project is to test the clusters found so far by using the test data provided by the PhysioNet challenge. This will be done by classifying the new data and seeing if the clusters line up with those previously found. More importantly, the medical trajectories of the patients must be compared. This

will show whether or not the clusters are similar just in terms of data but medical trajectory. If the medical trajectories align we can use that as a measure of success for the grouping.

One area we could explore is reducing the clinical features into domains. This is an approach used in a previous study in the field [5]. This could result in more clear subtyping as the patients can be grouped based on a feature more relevant to their illness. This would enable us to model the medical trajectories based on the domains associated with the illness. This would be effective to reduce the dimensionality of the data.

Another area we could improve upon is the idea of time limitation. We reckon that looking further into the structure of the data when we time-bound the data will show useful information that may be used practically. One potential analysis we could perform is tracking all five clusters' centroids as we vary the time limit. This will provide information regarding how useful our methodology is when doctors only have a few hours of data at hand.

Finally, one report published as a submission to the PhysioNet 2012 Challenge itself, examines a Cascaded SVM-GLM Paradigm [6]. This uses the same dataset as provided for our report, and therefore this would be a great way of comparing the effectiveness of our findings to an existing publication which used a completely different method. Another paper evaluating both of our research might provide useful comparative information that couldn't be found through using just one method.

6.1.1 Gaussian Mixture Model

We considered using another clustering algorithm; the Gaussian Mixture Model, which is an unsupervised learning algorithm. This was to have another way of analysing the data with a probabilistic model instead of a deterministic one. The Mixture Model is probabilistic and might prove to be better in a practical environment, allowing doctors room for judgement. By having probabilistic boundaries and showing the probability of a given patient belonging to each subtype, doctors have more flexibility and are able to have their expertise as an input.

6.2 Conclusion

From our research, we found there to be five key subtypes of patients. The usefulness of all subtypes is yet to be determined and can be clarified with the comparison of the subtypes to the medical trajectories of the patients, if we were given access to that information.

Our findings show that the meaningful variation between groups is mostly due to AST and ALT, indicating that liver-related issues are the main cause of ICU admissions. Essentially, the difference between two subtypes as compared to five is that five clusters provide more in depths analysis regarding the patients in the original two clusters. For $k = 5$, Subtype 1,2, and 3, are sub-clusters of Subtype 1 in $k = 2$, where Subtype 1, 2, and 3 differentiates the severity for patient subtypes with liver problems. On the other hand, Subtype 0, and Subtype 4 for $k = 5$ are the sub-clusters of Subtype 0 in $k = 2$. The difference being that Subtype 4 has below-average levels of AST and ALT, i.e. inadequate amounts of key enzymes resulting in higher in-hospital deaths proportion and lower survival rate compared to subtype 0. Meaninng, subtype 4 showing that patients in 4 requires slightly more intervention than subtype 0. From our analysis, medical professionals can determine which subtypes to act on first and what treatements to apply.

References

[1] Ikaro Silva, George Moody, Daniel J. Scott, Leo A. Celi, Roger G. Mark. *Predicting In-Hospital Mortality of Patients in ICU: The PhysioNet/Computing in Cardiology Challenge 2012.* URL:`https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3965265/` *used on 10-2018*

[2] Jeffrey De Fauw, Joseph R. Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, etc. *Clinically applicable deep learning for diagnosis and referral in retinal disease.* URL: `https://www.nature.com/articles/s41591-018-0107-6` *used on 11-2018*

[3] Anna Goldenberg, Suchi Saria. *Subtyping: What it is and its role in precision medicine.* URL: `https://www.semanticscholar.org/paper/subtyping%3A-What-It-is-and-Its-Role-in-Precision-Saria-Goldenberg/53953ed97fe0977397b1035acca1e1f11f9a7cb8` *used on 10-2018*

[4] *IBM Watson for Genomics.* URL: https://www.ibm.com/uk-en/marketplace/watson-for-genomics *used on 10-2018*

[5] Kelly C. Vranas, Jeffrey K. Jopling, Timothy E. Sweeney, Meghan C. Ramsey, Arnold S. Milstein, Christopher G. Slatore, Gabriel J. Escobar, Vincent X. Liu *Machine learning approach to identifying subtype of ICU patients.* URL:https://www.ncbi.nlm.nih.gov/pubmed/28640021 *used on 10-2018*

[6] Luca Citi, Riccardo Barbieri. *PhysioNet 2012 Challenge: Predicting Mortality of ICU Patients using a Cascaded SVM-GLM Paradigm.* URL: http://www.cinc.org/archives/2012/pdf/0257.pdf *used on 10-2018*

[7] Saurabh Jaju. *Comprehensive guide on t-SNE algorithm.* URL: https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/ *used on 10-2018*

[8] Laurens van der Maaten, Geoffrey Hinton *Visualising Data using t-SNE, Journal of Machine Learning Research 9 (2008) 2579-2605 published 11/08* URL : http://mlexplained.com/2018/09/14/paper-dissected-visualizing-data-using-t-sne-explained/ *used on 11-28*

[9] JR. Le Gall, P. Loirat, A. Alperovitch, P. Glaser, C. Granthil, D. Mathieu, P. Mercier, R. Thomas, D. Villers. *A simplified acute physiology score for ICU patients.* URL: https://www.ncbi.nlm.nih.gov/pubmed/6499483 *used on 11-18*

[10] FL. Ferreira, DP. Bota, A. Bross, C. Mélot, JL. Vincent. *Serial evaluation of the SOFA score to predict the outcome in critically ill patients.* https://www.ncbi.nlm.nih.gov/pubmed/11594901 *used on 11-18*

[11] *Physionet intensive care challenge.* URL: https://physionet.org/challenge/2012/?fbclid=IwAR0bjVwr9ZmwPz_fMu9a1VojyVHia90RQMBvP0ZprLd2jYHWa7ejmSz5FaQ#rules-and-dates *used on 10-2018*

[12] Cheng H. Lee, Natalia M. Arzeno, Joyce C. Ho, Haris Vikalo, Joydeep Ghosh. *An Imputation-Enhanced Algorithm for ICU Mortality Prediction.In Computing in Cardiology 2012, CinC 2012 (Vol. 39, pp. 253-256). [6420378]* URL: https://utexas.influuent.utsystem.edu/en/publications/an-imputation-enhanced-algorithm-for-icu-mortality-prediction *used on 10-2018*

[13] *Missing-data imputation.* URL: http://www.stat.columbia.edu/~gelman/arm/missing.pdf *used on 10-2018*

[14] Shuyin Xia, Zhongyang Xiong, Wei Xu, Yueguo Luo, Guanghua Zhang. *Effectiveness of the Euclidean distance in high dimensional spaces.* URL: https://www.sciencedirect.com/science/article/pii/S0030402615011493 *used on 11-2018*

[15] *National Health Service Height/Weight Chart* https://www.nhs.uk/live-well/healthy-weight/height-weight-chart/

[16] David Pittman. *Data analytics for doctors.* URL: https://www.medpagetoday.com/meetingcoverage/himss/44541 *used on 10-2018*

[17] Luca Citi, Riccardo Barbieri *Predicting Mortality of ICU Patients using a Cascaded SVM-GLM Paradigm.* URL:https://www.researchgate.net/publication/261213761_PhysioNet_2012_Challenge_Predicting_mortality_of_ICU_patients_using_a_cascaded_SVM-GLM_paradigm *used on 11-2018*

[18] Catherine R. Planey, Olivier Gevaert. *A Framework for subtyping across multiple dataset.* URL: https://genomemedicine.biomedcentral.com/articles/10.1186/s13073-016-0281-4 *used on 10-2018*

Appendix A

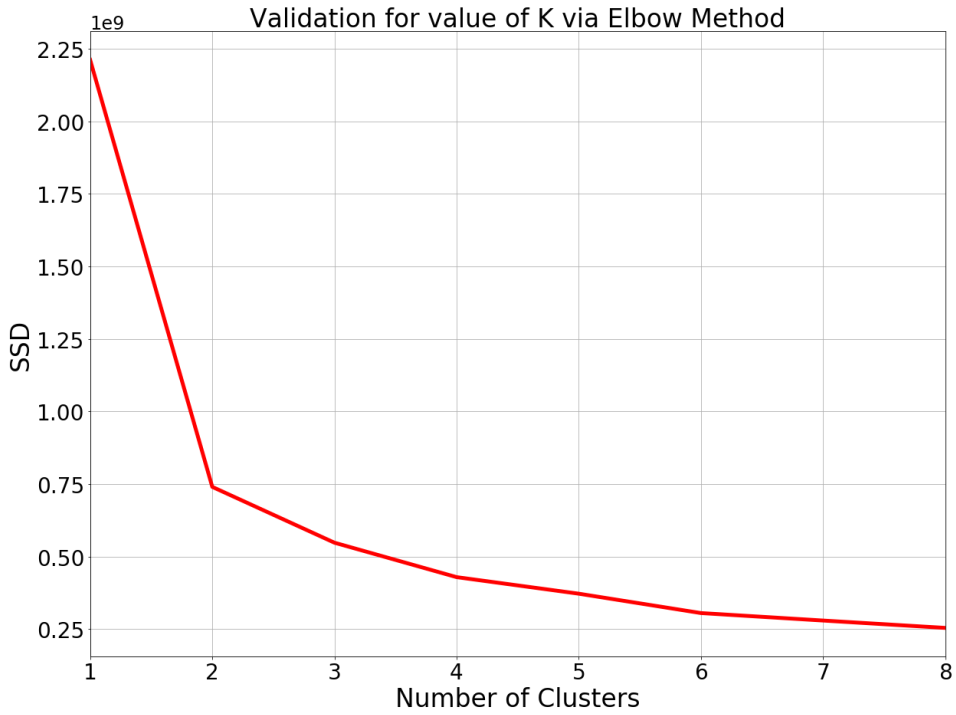


Figure 6: Graph showing the inverse relationship between number of clusters present and the Squared Sum of Distances (SSD) measured. The notable gradient change occurs at $k = 2$.

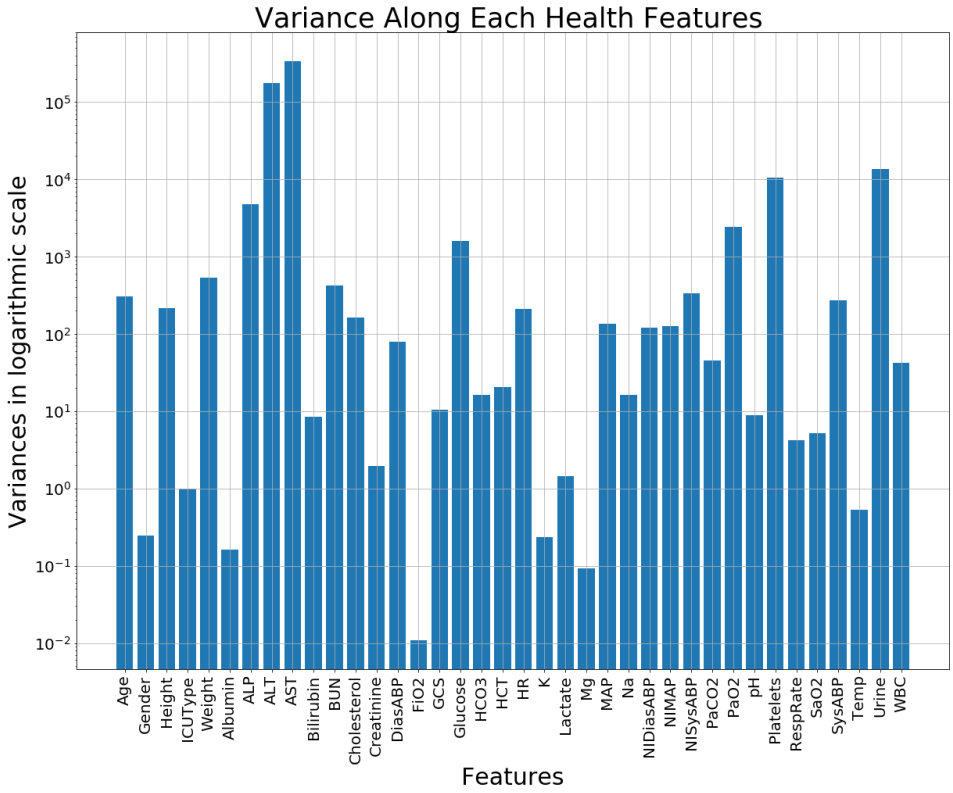
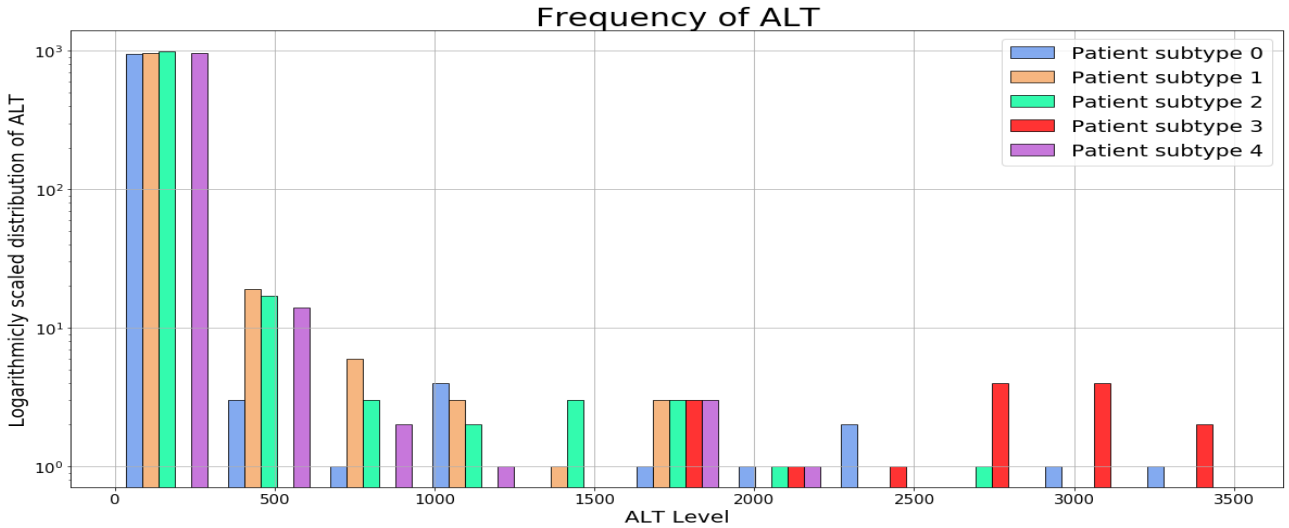
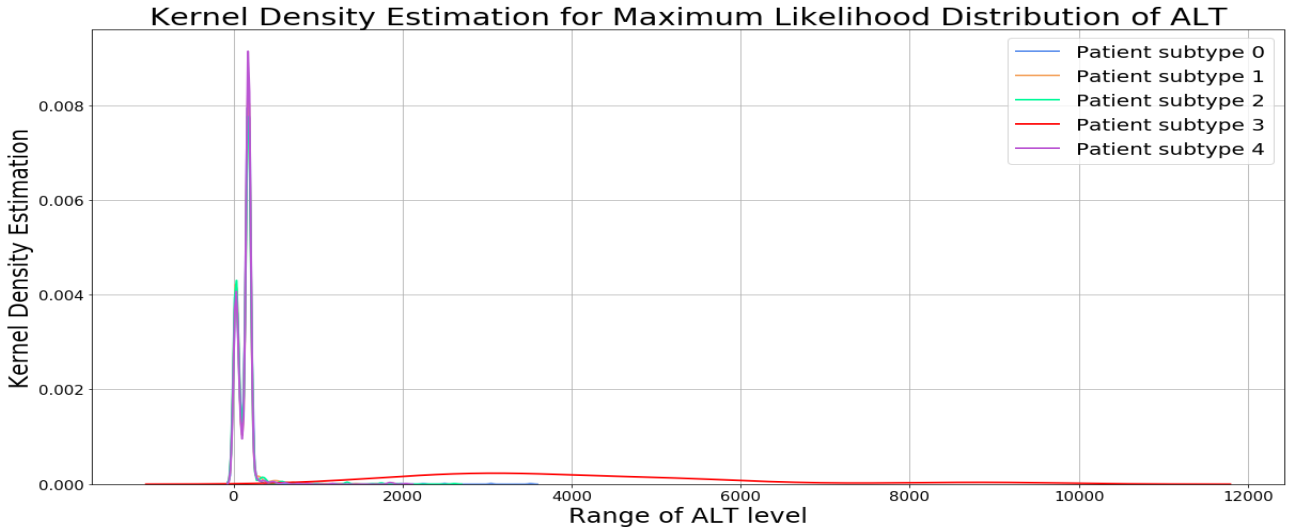


Figure 7: Graph showing the variances for every 37 health features. The two highest variance are ALT and AST, which accounts to 93% of all variances



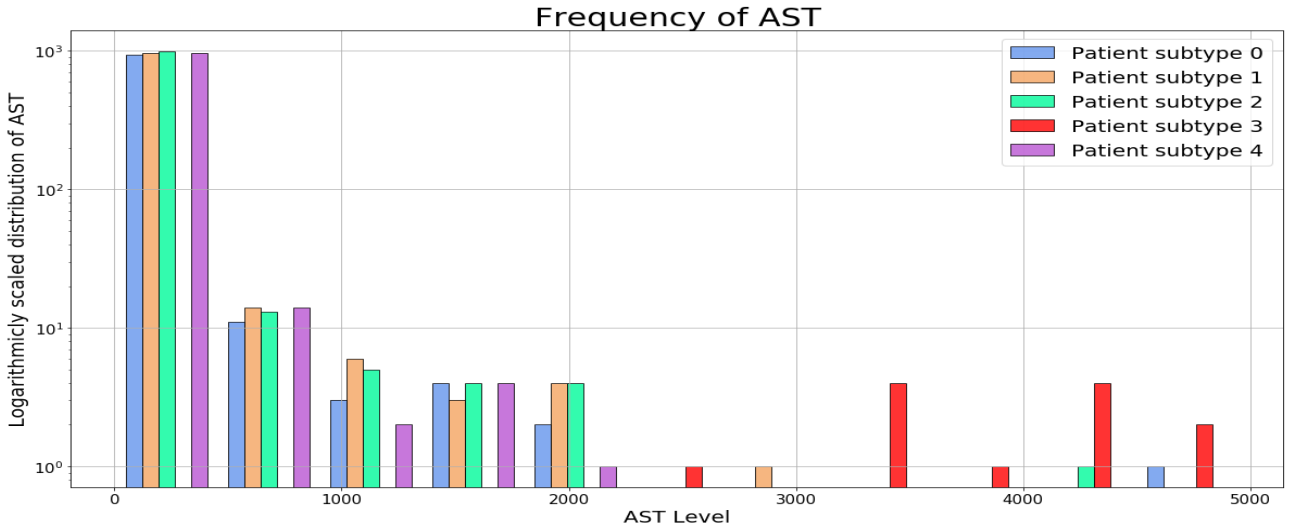
(a) Frequency distribution of ALT

Figure 8: Graph showing the frequency distribution of the second highest-varying feature: ALT. The graph is logarithmicly scaled to show the show the large variance posed by ALT in patient subtype 3



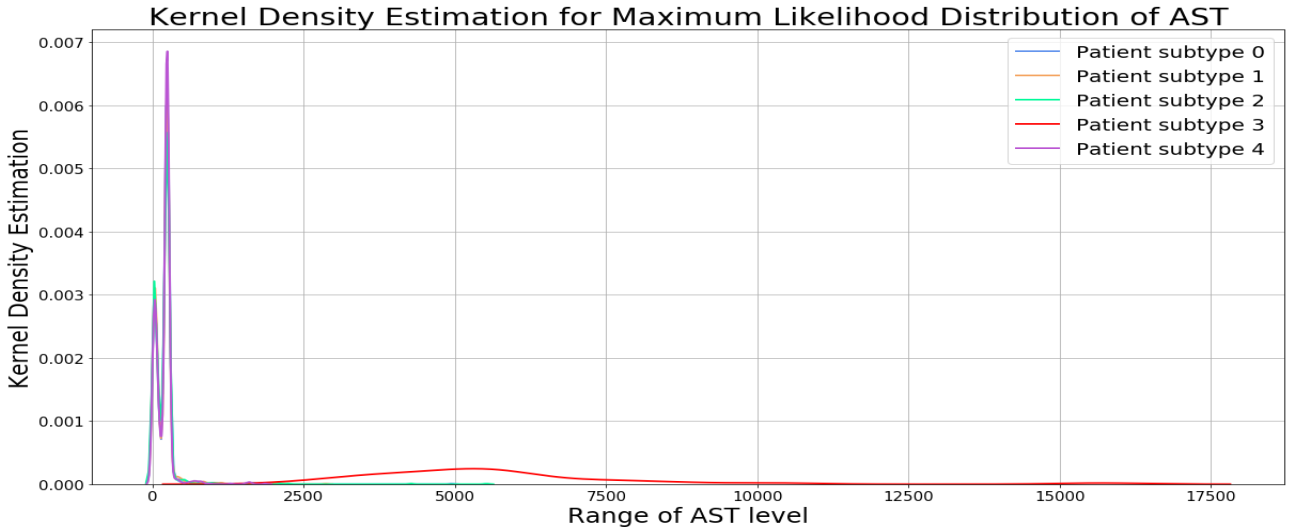
(a) Kernel Density Estimation of ALT

Figure 9: Graph depicting the theoretical maximum likelihood distribution for ALT.



(a) Frequency distribution of AST

Figure 10: Graph showing the frequency distribution of the highest-varying feature: AST. The graph is logarithmicly scaled to show the show the large variance posed by AST in patient subtype 3



(a) Kernel Density Estimation of AST

Figure 11: Graph depicting the theoretical maximum likelihood distribution for AST.

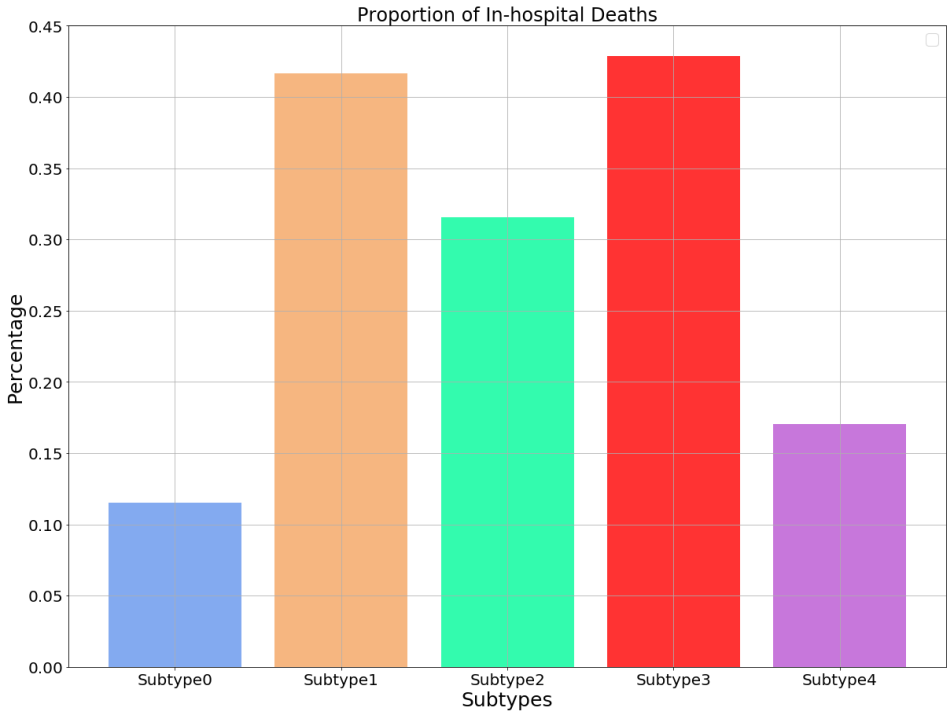


Figure 12: Bar chart showing the percentages of in-hospital deaths

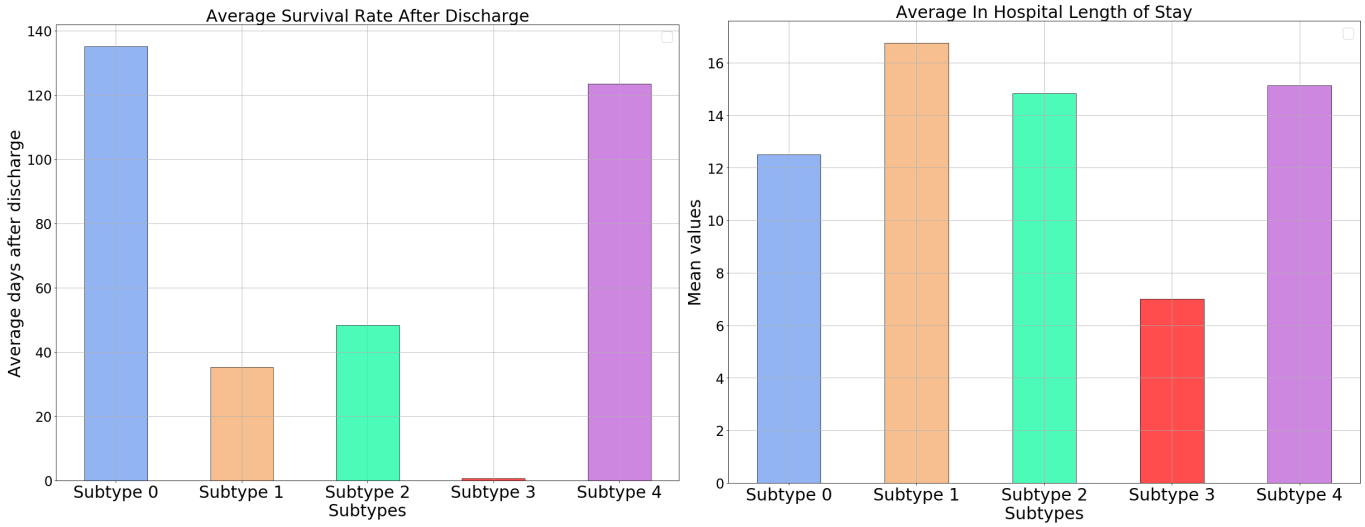


Figure 13: Proportion of average length of stay and survival rate in days for each cluster

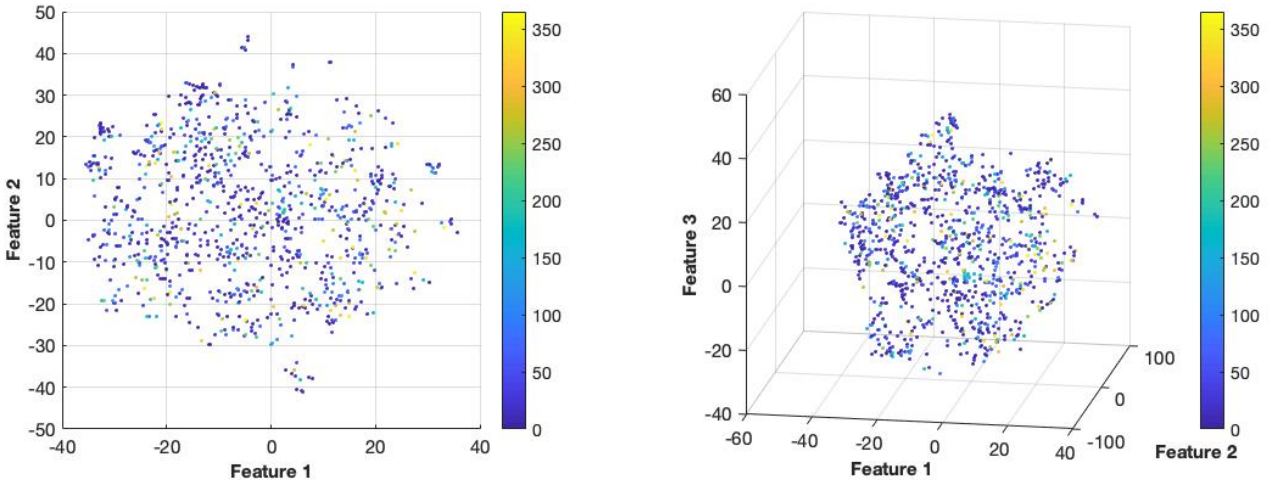


Figure 14: These are the additional results of running the t-SNE algorithm for Set A, in both 2D and 3D, using the *Survival* parameter as a colour mapping. This ignores recorded deaths and patients that survived more than a year - were these data points included the scatter graph would become illegible as over 2,600 of the 4,000 patients (65%) fell into these categories.

Appendix B

$$Var(x_{i_k}) = \frac{1}{N}\sum_{k=1}^{N}(x_{i_k} - \mu_i)^2 \quad Cov(x_i, x_j) = \frac{1}{N}\sum_{k=1}^{N}(x_{i_k} - \mu_i)(x_{j_k} - \mu_j) \quad \mu = \frac{1}{N}\sum_{i=1}^{N}\mathbf{v_i} = \begin{pmatrix}\mu_i \\ \mu_d\end{pmatrix} \tag{7}$$

$$\forall i, j \in [1, N] \quad \text{and} \quad \forall k \in [1, d] \tag{8}$$

$$\mathbf{Cu} = \lambda\mathbf{u} \tag{9}$$

$$\mathbf{v} = \mathbf{m} + \sum_{i=1}^{d}\mathbf{a_i}\mathbf{m_i} \tag{10}$$

$$\mathbf{C} = \begin{bmatrix} Var(x_1) & Cov(x_1, x_2) & Cov(x_1, x_3) & \cdots & Cov(x_1, x_d) \\ Cov(x_2, x_1) & Var(x_2) & Cov(x_2, x_3) & \cdots & Cov(x_2, x_d) \\ Cov(x_3, x_1) & Cov(x_3, x_2) & Var(x_3) & \cdots & Cov(x_3, x_d) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Cov(x_d, x_1) & Cov(x_d, x_2) & Cov(x_d, x_3) & \cdots & Var(x_d) \end{bmatrix} \tag{11}$$

$$\mathbf{C}_{new} = \begin{bmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ 0 & 0 & \lambda_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_p \end{bmatrix} \tag{12}$$

Appendix C

Below is a list of the parameters used in the PhysioNet datasets.

- Albumin: g/dL
- ALP (Alkaline phosphatase): IU/L
- ALT (Alanine transaminase): IU/L
- AST (Aspartate transaminase): IU/L
- Bilirubin: mg/dL
- BUN (Blood urea nitrogen): mg/L
- Cholestrol :mg/dL
- Creatinine :mg/dL
- DiasABP (Invasive arterial blood pressure): $mmHg$
- FiO_2 (Fractional inspired O_2): $0-1$
- GCS (Glasgow Coma Score) $3-15$
- Glucose: mg/dL
- HCO_3 Bicarbonate: $(mmol/L)$
- HCT (Hematocrit): %
- Heart rate: bpm
- K: mEq/L
- Lactate: $mmol/L$
- Mg: $mmol/L$
- MAP (Invasive mean arterial blood pressure): $mmHg$
- MechVent (Mechanical ventilation respiration): $0 : false, 1 : true$
- Na: mEq/L
- NIDiasABP (Non-invasive diastolic arterial blood pressure): $mmHg$
- NIMAP (Non-invasive mean arterial blood pressure): $mmHg$
- NISysABP (Non-invasive systolic arterial blood pressure): $mmHg$
- $PaCO_2$ / PaO_2 (Partial pressure of arterial CO_2 and O_2): $mmHg$
- pH (Arterial pH): $0-14$
- Platelets: $cells/nL$
- Respiration rate: bpm
- SaO_2 (O_2 saturaion in hemoglobin): %
- SysABP (Invasive systolic arterial blood pressure): $mmHg$
- Temperature: $^{\circ}C$
- TropI (Troponin-I): $\mu g/L$

- TropT (Troponin-T): $\mu g/L$
- Urine: mL
- WBC (White blood cell count): $cells/nL$
- Weight: kg

Subtype	0	1	2	3	4
Cluster Size	2544	27	38	7	1384
Age	65	57	56	33	64
Gender	1	0	1	1	1
Height	170	167	178	172	170
ICU Type	3	3	3	3	3
Weight	82	79	81	98	84
Albumin	3	3	3	3	3
ALP	108	161	148	236	95
ALT	178	3016	1537	6652	34
Bilirubin	251	4893	1662	8822	47
BUN	2	5	3	4	2
Cholesterol	24	35	31	31	29
Creatinine	157	157	156	147	156
DiasAPB	1	2	2	3	2
FiO_2	59	63	63	55	60
GCS	1	1	1	1	1
Glucose	12	8	10	9	11
HCO_3	137	158	151	179	137
HCT	24	20	22	18	23
Heart Rate	31	32	33	34	32
K	87	93	95	116	88
Lactane	4	4	4	4	4
Mg	2	7	4	8	2
MAP	2	2	2	2	2
Na	81	83	83	80	82
NiDiasABP	1	1	1	1	1
NIMAP	139	140	140	138	139
NISysABP	57	58	57	58	58
$PaCO_2$	76	76	74	76	76
PaO_2	118	113	112	118	117
pH	41	35	39	31	40
Platelets	152	131	140	120	142
Respiration Rate	8	7	7	7	7
SaO_2	204	146	163	119	211
SysABP	20	20	19	20	20
Temperature	97	94	97	96	97
TropI	119	117	119	95	118
TropT	37	36	37	37	37
Urine	135	75	102	66	134
WBC	12	15	12	14	13
SAPS-I	15	15	15	15	15
SOFA	7	7	7	7	7

Subtype	0	1
Cluster Size	37	3963
Age	52	64
Gender	0	1
Height	169	170
ICU Type	3	3
Weight	84	83
Albumin	3	3
ALP	172	104
ALT	3657	139
Bilirubin	5461	192
BUN	5	2
Cholesterol	35	25
Creatinine	154	157
DiasAPB	3	1
FiO_2	61	60
GCS	1	1
Glucose	9	12
HCO_3	162	137
HCT	20	24
Heart Rate	32	31
K	97	87
Lactane	4	4
Mg	7	2
MAP	2	2
Na	82	81
NiDiasABP	1	1
NIMAP	140	139
NISysABP	59	57
$PaCO_2$	76	76
PaO_2	115	117
pH	35	40
Platelets	132	148
Respiration Rate	7	8
SaO_2	144	206
SysABP	20	20
Temperature	95	97
TropI	112	119
TropT	37	37
Urine	76	135
WBC	14	13
SAPS-I	15	15
SOFA	7	7

Table 2: Table containing the cluster centroids for all parameters used in the dataset along with SAPS-I and SOFA scores for both $k = 2$ and $k = 5$ clusters. This is the full version of Table 1 from Section 5.1. Entry $x_{i,j}$ represents the average ith feature of the jth cluster, rounded to the nearest integer.

Subtype	0	1	2	3	4
Cluster Size	966	32	981	998	1023
Age	64	52	64	64	65
Gender	1	0	1	1	1
Height	169	168	170	170	170
ICU Type	3	3	3	3	3
Weight	83	83	84	82	83
Albumin	3	3	3	3	3
ALP	106	168	103	104	103
ALT	145	3994	138	141	139
Bilirubin	197	5698	192	198	199
BUN	2	4	2	2	2
Cholesterol	25	34	25	26	26
Creatinine	156	154	157	157	156
DiasAPB	1	2	1	1	1
FiO_2	60	62	60	59	59
GCS	1	1	1	1	1
Glucose	12	9	12	12	12
HCO_3	136	156	137	138	138
HCT	24	20	24	24	24
Heart Rate	32	33	31	32	31
K	87	99	87	87	87
Lactane	4	4	4	4	4
Mg	2	6	2	2	2
MAP	2	2	2	2	2
Na	81	83	81	81	81
NiDiasABP	1	1	1	1	1
NIMAP	139	139	139	139	139
NISysABP	57	58	57	58	57
$PaCO_2$	76	76	76	77	76
PaO_2	117	114	117	118	117
pH	41	34	40	41	40
Platelets	148	131	146	148	151
Respiration Rate	8	7	7	7	7
SaO_2	207	143	204	208	205
SysABP	20	20	20	20	20
Temperature	97	95	96	97	97
TropI	119	113	119	119	118
TropT	37	37	37	37	37
Urine	139	74	140	133	127
WBC	12	14	13	13	13

Table 3: Table containing the cluster centroids for all parameters used in the dataset for $k = 5$ clusters. This is table is a time-bounded version of Table 2 above, computed after 6 hours worth of data.