

# Customer Segmentation Using Unsupervised Learning on Daily Energy Load Profiles

J. du Toit, R. Davimes, A. Mohamed, K. Patel, and J. M. Nye  
Eskom, Brackenfell, South Africa

Email: {jacowp357, ricdavimes, ashfaaqmohamed22, kavirpatelkp, nyejon}@gmail.com

**Abstract**—Power utilities collect a large amount of metering data from substations and customers. This data can provide insights for planning outages, making network investment decisions, predicting future load growth and predictive maintenance. One of the requirements is the ability to group similar behaving loads together. This paper provides a comparison between different similarity measures, used in the k-means clustering algorithm, to group daily load profiles together based on metering data. The various methods are compared using two well-known cluster evaluation metrics and the results are then analysed by subject matter experts to determine the validity of the findings. The results, from our particular data set, indicate that various speed improvement techniques can be considered that complement the k-means algorithm without sacrificing intra-to inter-cluster accuracy. A small increase in the optimal number of clusters, using domain expertise, allowed for additional profiles to be extracted that were not explained by algorithmic evaluations. Interplay between both theoretical evaluations and domain knowledge facilitated a preferred number of clusters for practical purposes.

**Index Terms**—daily load profiles, customer segmentation, k-means clustering, similarity measures, non-uniform binary split

## I. INTRODUCTION

Utilities around the world are preparing to use the vast amount of metering and general customer data that will be collected following the rollout of smart metering devices. It is vital that utilities extract value from this data and make use of it to better understand their customers and make improved investment decisions.

Conventional descriptive statistics has its limitations when trying to extract information, make predictions, automate customer monitoring and network anomaly detection. This paper investigates a machine learning approach, using unsupervised learning techniques that allows for more in-depth analytical capability, accurate predictions and processing of high volumes of data.

A well-known unsupervised learning technique, the k-means clustering algorithm, is explored. Four distance metrics for the k-means algorithm are studied and the differences between each algorithm will be analysed

using two different evaluation metrics. The objective is to group similar load profiles, using the different metrics, and then compare the differences between the groups. Principle Component Analysis (PCA) and a suboptimal cluster boundary technique, the non-uniform binary split (NUBS) algorithm, are both used to improve convergence time. The cluster centroids are interpreted and further explained by visualisation and linkage to geographic locations for practical analysis of the various load types.

The paper is structured as follows: Section II covers the research methodology and data preparation, Section III show cases the results and Section IV concludes the paper.

### A. Literature Review

Advantages of segmenting daily energy profiles can span from: achieving more sustainable and efficient urban or rural development, allowing a more flexible electricity market, smoothing of energy peaks and imbalances, and enhancing the exploitation of renewable energy sources [1]-[7]. The shapes of daily load profiles can be grouped into three main types: residential, industrial and commercial [8]. Methods that have been applied and modified to segment temporal data include: variations of Hierarchical clustering, variations of k-means clustering, Self-organising maps, and Fuzzy clustering [9], [10]. Examples of work that make use of techniques related to this study can be found in [11]-[18]. These studies focused on comparing and optimising similarity measures used in some of the techniques alluded to above. Most of these studies were committed to overcoming issues related to clustering temporal data, such as: sensitivity to small changes, sum of distance not capturing the shape of a curve and computational complexity using other similarity measures [19].

Optimisation of k-means with respect to convergence time presents a gap in literature for the application considered in this study. Ref. [20] introduces a low complexity pre-clustering technique known as the Non-Uniform Binary Split (NUBS) algorithm. This can be utilised to initialise the k-means centroids.

In addition to exploring the different similarity measures, this paper combines the same approaches with techniques that can be used to increase convergence time of the k-means clustering algorithm.

## II. METHODOLOGY

### A. Data Extraction and Preparation

In this study, half hourly kVA readings of 11 and 22 kV feeders are analysed. The data for the months of February and March from 2007 to 2014 were selected to limit the variations of the load profiles throughout the year. These months represent the typical summer period where farming and industrial activity is usually at its peak. Each feeder was assigned a unique identification number for post computational reference together with the corresponding locational coordinates for mapping purposes.

All of the load profiles were broken down into individual daily profiles to form an  $m \times 48$  matrix where  $m$  is the total number of daily load profiles. These will later be assigned to the resulting clusters individually. Missing values were interpolated up to a maximum of five consecutive readings so as to maintain data integrity. Days with zero or missing values or those with little variance were removed for algorithmic purposes. One assumption made here is that variance is a measure of information and thus low variance signals do not justifiably contribute to information gain, but rather act as a hindrance to algorithm convergence.

Finally, the load profiles were standardised row-wise as follows:

$$x_{new} = \frac{x - \mu}{\sigma} \quad (1)$$

where,  $x$  is each individual profile value,  $\mu$  is the row mean and  $\sigma$  is the row standard deviation. This ensured that all data were of the same scale and unit variance, allowing for more accurate relationship exploration.

### B. K-means Clustering

The clustering technique described in this section was selected due to its simplicity (low complexity advantage for the production environment), ability to easily modify the similarity metrics and compatibility with the NUBS algorithm to speed up convergence.

The aim of clustering is to find good centroids that represent unique information about the load profiles. The algorithm starts with an initial estimate of the centroids and alternately determines clusters from centroids, and centroids from the cluster means. The following steps formally describe the k-means algorithm:

1. Start with an initial set of  $K$  centroids  $[y_1, \dots, y_K]$ , or choose  $K$  vectors at random from data set  $[x_1, \dots, x_n]$ ,
2. Split data set into  $K$  clusters  $\{X_1, \dots, X_k\}$ , where data point  $x_n$  belongs to cluster  $X_i$  if  $y_i$  is closest centroid  $X_i = \{x_n \mid d(x_n, y_i) \leq d(x_n, y_j), j = 1 \dots K\}$ ,
3. Update the centroids based on new clusters  $y_i = c(X_i), i = 1 \dots K$ ,
4. Calculate total distortion as sum of distances to closest centroids  $D = \sum_{n=1}^N d(x_n, y_{i(n)})$ , where  $i(n) = k$  if  $x_n \in X_k$ ,
5. Repeat steps 2–4 until stopping criterion.

A stopping condition entails reaching a maximum set number of iterations or a test that establishes significance in the relative drop in distortion between iterations (e.g., when a minimum criterion is reached for the distortion difference where  $\Delta D = \frac{D_{prev} - D_{curr}}{D_{prev}} < 10^{-4}$ ) [20].

A typical distance metric,  $d(x, y)$ , used in k-means is the Euclidean distance. The following section describes the distance metrics considered in this study.

### C. Distance Metrics

In k-means clustering the objective is to represent a data set by a set of clusters that should ideally lie in areas of feature space with high data density. The cluster centroids should capture the internal structure of the data tied to an appropriate distance measure or metric. The metric selection criteria and a formal explanation of each are shown below.

**Pearson Correlation Distance:** This measure is useful for time series comparison where the similarity between the shapes of two profiles needs to be investigated. This is performed using the Pearson correlation coefficient, where the normalized angular separation is calculated by centering the dataset by its mean as follows:

$$r_{ij} = \frac{\sum_{k=1}^d (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{\sqrt{\sum_{k=1}^d (x_{ik} - \bar{x}_i)^2} * \sqrt{\sum_{k=1}^d (x_{jk} - \bar{x}_j)^2}} \quad (2)$$

where,  $\bar{x}_i = \frac{1}{d} \sum_{k=1}^d x_{ik}$ . The distance measure can then be defined as:

$$d(x, y) = 1 - r_{ij} \quad (3)$$

**Cosine Similarity:** Cosine distance, like Pearson correlation distance, is a calculation of the angular separation between two vectors without considering the length of the respective vectors. This serves as a suitable measure when signal amplitude is unimportant and signal shape similarity is required when determining similarity. Cosine distance is calculated as:

$$\cos(\theta)_{ij} = \frac{\sum_{k=1}^d (x_{ik})(x_{jk})}{\sqrt{\sum_{k=1}^d (x_{ik})^2} * \sqrt{\sum_{k=1}^d (x_{jk})^2}} \quad (4)$$

Note that in the case of data normalisation in the Pearson correlation distance measure, both distance functions yield the same result.

**Dynamic Time Warping (DTW) Distance :** This distance metric addresses the issue of slight shifts along the time axis (speed/frequency variations) being misleadingly large in distance calculation and therefore mitigates the effects of distortion [21]. Let  $a = \{a_1, \dots, a_m\}$  and  $b = \{b_1, \dots, b_n\}$  be two time series signals where the distance between the two is required and  $M(a, b)$  is a square matrix of shape  $m \times n$ . Point-wise distance measures are then calculated between  $a$  and  $b$  as  $M_{ij} = (a_i - b_j)^2$ . The warping path, defined as  $P = [(x_1, y_1), \dots, (x_m, y_n)]$ , is found that encapsulates the traversal of the matrix [11] according to the following rules:

1.  $(x_1, y_1) = (1, 1)$  and  $(x_m, y_n) = (m, n)$ ,
2.  $0 \leq x_{i+1} - x_i \leq 1$  for  $i < m$  and  $0 \leq y_{j+1} - y_j \leq 1$  for  $j < n$ .

The distance between two signals is defined as:

$$D_l(a, b) = \sum_{i=1}^s l_i \quad (5)$$

where,  $l_i$  is the pointwise distance  $M_{a_{x_i}, b_{y_i}}$  for any given warping path. The DTW measure is given by the path that results in the minimum distance from the set of all possible distances within the constraints defined above.

**Euclidean Squared distance:** This measures the vector displacement between two points. This is the most common approach when considering continuous signals and can be formulised by the following equation [1]:

$$d(x, y) = \sum_{k=1}^m (x_k - y_k)^2 \quad (6)$$

where,  $x$  and  $y$  are two feature vector points.

Clustering with the Euclidean Squared distance metric is faster than the normal Euclidean distance metric as it does not take into account the square root. An important consideration is that the Euclidean Squared distance metric fails to take into account vector correlation as it neglects to consider the direction and angle of the feature vectors.

#### D. Principal Component Analysis

Principal component analysis (PCA) is an exploratory tool with the aim of simplifying large and complex data sets by reducing its dimensionality, and subsequently removing some noise. Accordingly, it is considered in this study as a pre-processing step to k-means. PCA summarises high dimensional temporal data by creating orthogonal (uncorrelated) “artificial” variables (new signal components in this case). These components are linear combinations of the original data and are constructed by exploiting the variability (spread) of each variable and the correlation between variables. The principle components are derived by finding the eigenvectors and corresponding eigenvalues of the correlation or the covariance matrix between the variables in the original data set.

Formally, from the definition of a covariance matrix,

$$\Sigma = (X - \mu)^T (X - \mu) \quad (7)$$

where,  $X$  denotes a  $(n \times p)$  matrix of input data and  $\mu$  is a  $(1 \times p)$  vector consisting of the mean of each variable. PCA is a decomposition of the covariance matrix:

$$\Sigma = (X - \mu)^T (X - \mu) = P \Lambda P^T \quad (8)$$

where,  $P$  is a  $(p \times p)$  orthogonal, rotational matrix, with the columns of  $P$  representing the eigenvectors of the input covariance matrix and  $\Lambda$  is a  $(p \times p)$  diagonal matrix representing the corresponding eigenvalues.

The principle components represent a rotation of the original data onto a new axis, with the new axes best revealing patterns or structures within the data. The rotated data, which can be represented in a lower dimensional space depending on the number of principle

components selected (i.e., columns of  $P$  selected) is represented by the following equation:

$$C = (X - \mu)P \quad (9)$$

The components are ordered according to their variance (size of their corresponding eigenvalues) with the first component encapsulating the highest relative variability in the data.

Literature suggests that the number of components selected should be such that sufficient variance remains as is applicable for the specific application. In most cases, most variance can be explained by a largely reduced number of dimensions. For practical and production purposes this reduces computation time and allows the clustering algorithms to run faster when applied prior to k-means.

#### E. Non-Uniform Binary Split (NUBS)

The k-means algorithm is sensitive to the initial choice of centroids and can be slow when running on large data sets. Normally, it requires multiple runs and inspections to validate good centroids and clusters. The objective of the NUBS algorithm is to find suboptimal cluster boundaries and use the centroids to initialise the k-means algorithm. These initialisations are used to give k-means a good start by placing the initial centroids at selected regions of lower comparative distortion and serves as coarse clustering to be redefined by k-means [20]. The steps in the algorithm are as follows:

1. Start with all data points  $x_n$  in one cluster  $X_1$  with associated centroid  $y_1 = c(X_1)$ , and let cluster counter  $k = 1$ ,
2. Repeat steps 3 to 6,  $K-1$  times, to obtain  $K$  centroids,
3. Choose cluster  $X_i$  with largest distortion, as measured by average distance of points in cluster to centroid,  $D_j = \frac{1}{N_j} \sum_{n=1}^{N_j} d(x_n^{(j)}, y_i)$ , with  $X_j = \{x_n^{(j)} \mid n = 1 \dots N_j\}$ , and  $D_i \geq D_j$  for  $j = 1 \dots k$ ,
4. Split selected cluster  $X_i$  into two sub-clusters  $X_a$  and  $X_b$  by using k-means with  $K = 2$ ,
5. Replace centroid  $y_i$  and add new centroid by letting  $y_i = c(X_a)$  and  $y_{k+1} = c(X_b)$ ,
6. Increment cluster counter  $k \leftarrow k + 1$ .

Instead of initialising k-means with random data points, the centroids found by NUBS were used in an effort to increase convergence time. The algorithm has a binary search order complexity  $O(\log(K))$ . The results from utilising this technique were compared with standard k-means.

#### F. Evaluation Metrics

This section highlights the two evaluation metrics used for finding the optimal number of clusters for each distance metric. These two metrics were selected based on the findings in [22].

**Davies-Bouldin (DB) Index:** This index produces a ratio of the sum of *intra-cluster* dispersion to the sum of *inter-cluster separation* [1]. The intra-cluster dispersion is calculated as:

$$S_i = \frac{1}{T_i} \sum_{j=1}^{T_i} d_i \quad (10)$$

where,  $T_i$  is the number of vectors in cluster  $i$  and  $d_i$  is the distance metric used to calculate the distance between the sample vector and its corresponding centroid. The inter-cluster separation is then calculated as the distance between cluster centroids and is denoted by  $l_{ij}$  where the distance is calculated using the various distance metrics alluded to above. Finally, the DB index is calculated as:

$$DB = \frac{1}{K} \sum_{i=1}^K R_{i,qt} \quad (11)$$

where,  $R_{i,qt} = \max_{j,j \neq i} \left\{ \frac{S_{i,q} + S_{j,q}}{l_{ij,t}} \right\}$ ,  $K$  is the number of clusters and the aim is to minimise the DB index.

**Silhouette Index:** For each sample,  $i$ , in cluster  $K_j (j = 1, \dots, c)$ , the Silhouette evaluation metric computes a quality measure  $s_i (i = 1, \dots, m)$  also known as the *Silhouettewidth*, which gives an indication of confidence that such a sample belongs to cluster  $K_j$  and is defined as:

$$s_i = \frac{a_i - b_i}{\max\{a_i, b_i\}} \quad (12)$$

where,  $a_i$  is the mean distance between sample  $i$  and all other samples in  $K_j$  and  $b_i$  is the minimum mean distance between sample  $i$  and all samples in the other clusters. It is thus notable that all values of  $s_i$  lie on the interval -1 and 1. A value close to 1 indicates a correctly clustered sample while a value close to -1 suggests an incorrectly clustered sample. A value close to 0 indicates that the sample could have been assigned to a neighbouring cluster. For each cluster a *cluster silhouette*,  $S_j$ , is calculated as the sum of the individual *silhouette widths* in the cluster,  $K_j$ . This measure gives an indication of the cluster isolation properties. Finally, the *silhouette index* is calculated as:

$$S_u = \frac{1}{c} \sum_{j=1}^c S_j \quad (13)$$

where,  $c$  is the number of clusters. The *silhouette index* also gives an indication of the optimal number of clusters, which is regarded as the value of  $c$  that gives the highest *silhouette index*.

### III. RESULTS

This section explains the results and is divided into subsections for convenience.

#### A. Distance Metric and Initialisation Comparison

The optimal numbers of  $K$ , for each distance metric using the Davies Bouldin index (at the elbow point), and the Silhouette index (at the maximum) are indicated in Table I. A comparison between random initialisation of k-means and using the centroids extracted from the NUBS algorithm is also presented.

From Table I and Table II, it is evident that the use of NUBS yields the same, or similar, results as random initialisation. However, as is indicated in Fig. 1, the convergence time and variance using NUBS is

substantially less than randomly initialising the cluster centroids and shows an increasingly optimised performance as the dataset being considered grows.

TABLE I. CLUSTERING RESULTS

Distance Metric:	Initialization	Davies Bouldin:		Silhouette:	
		Metric	No. Clusters	Metric	No. Clusters
Correlation	Random	16.1749	2	0.4479	2
	NUBS	16.1749	2	0.4479	2
Cosine	Random	16.1749	2	0.4479	2
	NUBS	16.1749	2	0.4479	2
DTW	Random	1.359	2	0.2944	2
	NUBS	1.359	2	0.2944	2
Euclidean	Random	1.3573	2	0.4749	3
	NUBS	1.3572	2	0.4749	3

TABLE II. CLUSTERING RESULTS USING PCA PRE-PROCESSING (8 COMPONENTS 95% VARIANCE)

Distance Metric:	Initialization:	Davies Bouldin:		Silhouette:	
		Metric	No. Clusters	Metric	No. Clusters
Correlation	Random	12.1632	3	0.5132	3
	NUBS	12.1362	3	0.5132	3
Cosine	Random	24.9963	3	0.3305	3
	NUBS	22.4995	3	0.3663	3
DTW	Random	1.2764	2	0.3167	2
	NUBS	1.2764	2	0.3167	2
Euclidean	Random	1.2235	3	0.5124	3
	NUBS	1.2235	3	0.5124	3

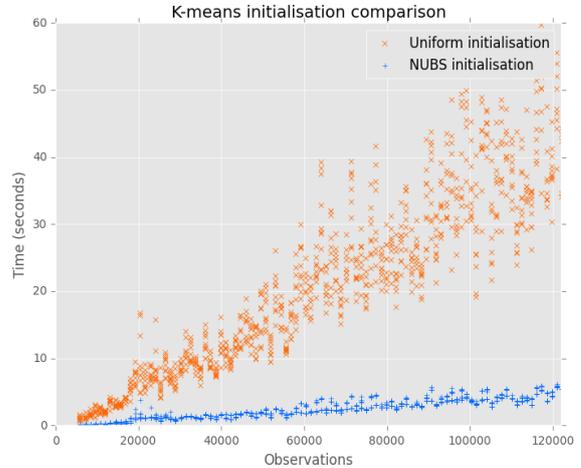


Figure 1. Speed comparison between k-means Random Initialisation and NUBS Initialisation.

The employment of PCA, prior to k-means, yields cluster quality metrics superior to raw k-means. This is

due to noise reduction, which is suitable for centroid approximations.

**B. Clustering Results**

The cluster results were obtained using k-means applied to the PCA scores with the Euclidean distance metric. The number of clusters,  $K$ , was selected based on the results obtained from both the DB and Silhouette indices combined with domain expertise. While the two cluster quality measures suggested three clusters, it was found that allowing for five clusters, additional information was obtained that would otherwise not have been evident.

The shape of the different clusters is shown in Fig. 2 to Fig. 6. These results correspond well with the expected load profiles that were derived in [8].

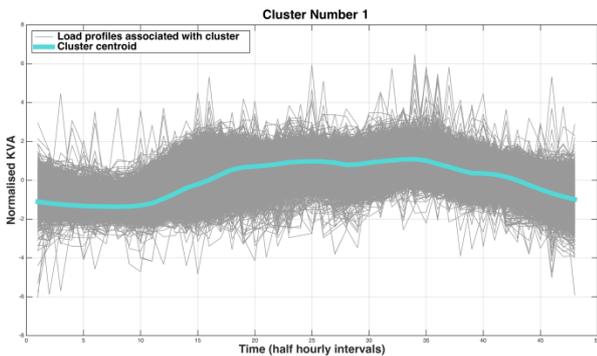


Figure 2. Cluster one (Industrial and Farming weekday).

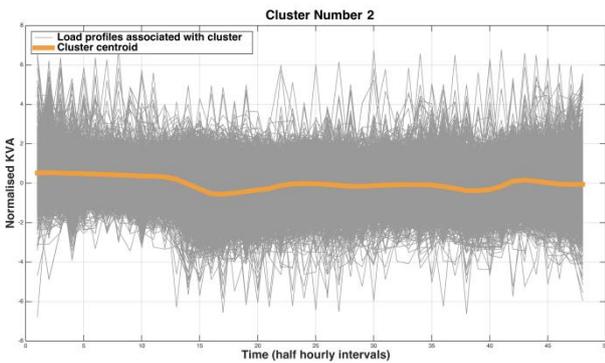


Figure 3. Cluster two (Pumping and Farming).

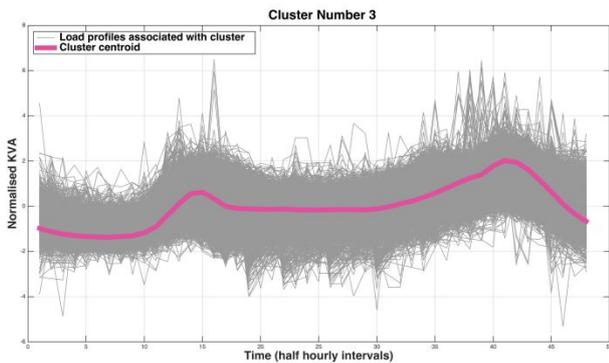


Figure 4. Cluster three (Residential weekday).

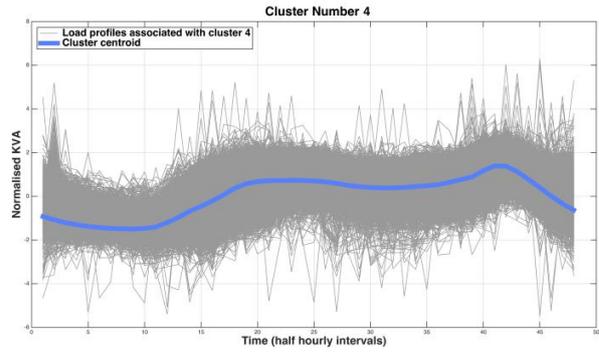


Figure 5. Cluster four (Residential weekend).

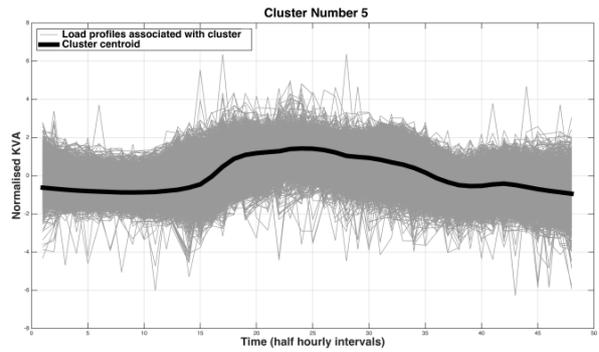


Figure 6. Cluster five (Industrial and Commercial weekend).

A more comprehensive description of these clusters will be shown in the following section. Some of the shapes of the clusters look similar, which explains the optimal  $K$  (two or three), determined by the evaluation metrics. The differences can, however, be explained by relating each cluster to a real world scenario. As an example, cluster one and five look similar, however, in cluster five the load decreases earlier in the day. This, as will be shown in the following section, can represent days where people typically leave work earlier. Cluster two is typical of pumping loads where pumps are operated at times of the day when electricity is cheapest. Finally, cluster three and four are typical residential profiles with cluster three representing a weekday and cluster four a weekend.

**C. Cluster Analysis and Practical Interpretation**

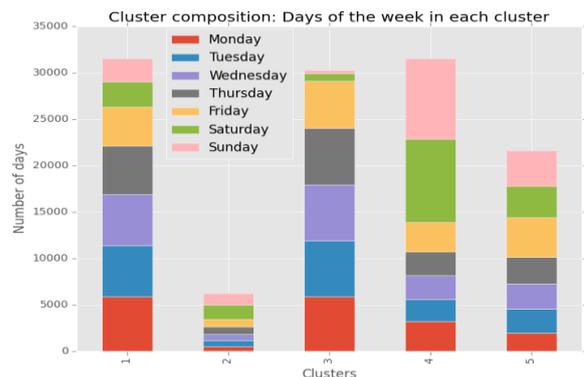


Figure 7. Breakdown of the number of days in each cluster grouped by the day of the week.

To determine whether the clustering results reflect the typical understanding of load types, a breakdown of the days of the week in each cluster is shown in Fig. 7. Clusters one and three represent typical weekday loads and cluster four represents the majority of weekends. Cluster five represents a typical weekend load including Friday. This corresponds with the fact that many customers or areas would have a noticeably different load during the week and on weekends.

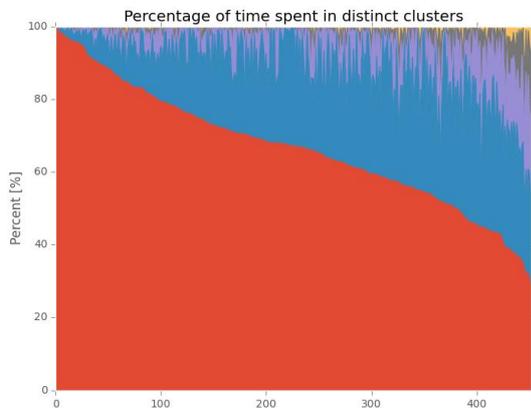


Figure 8. Percentage of time each feeder spends in distinct clusters.

In Fig. 8, it is shown how loads move between clusters and the duration of load to cluster occupancy. It displays that each load typically has two to three distinct load profiles. Fig. 8 was derived by counting the number of times a load spends in each cluster and then sorting each load so that the dominant cluster is represented by the red curve. This demonstrates how the loads change between clusters and the time spent in each cluster. It is confirmatory that the majority of the loads have the same load profile about 70% of the time, which relates to five days a week being 70% of the week. There is some variation from this on certain loads, but in those cases they can be explained by the location or type of load.

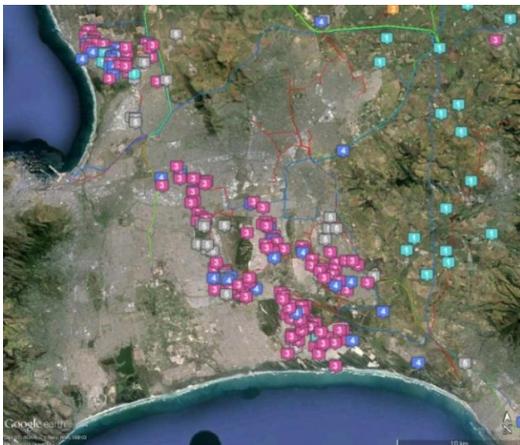


Figure 9. Cluster breakdown in an urban area.

As shown in Fig. 9 and Fig. 10 (satellite imagery using *Google Earth*), the types of load in an area are grouped into similar clusters. In urban areas, clusters three and four dominate as expected. As you move into rural or farming areas, cluster one becomes the dominant profile.

It should be noted that the cluster that the group is assigned into was selected based on the most common cluster for that particular load. This means that there would be different cluster assignments if weekdays or weekends were filtered out.

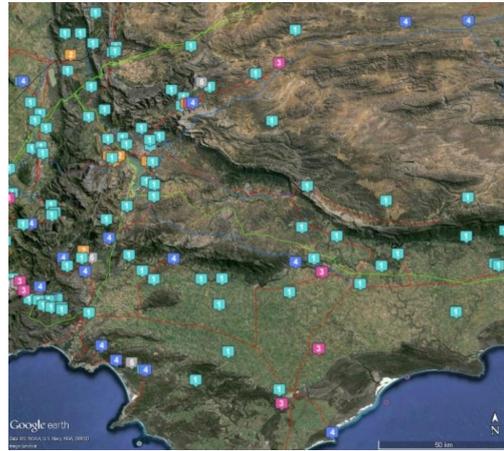


Figure 10. Cluster breakdown in a rural area.

#### IV. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

The Silhouette evaluation in Table II suggests that using correlation as a distance metric gives the best result. However, the Euclidean distance metric was considered as the slight outperformance was not enough to warrant the added computational complexity. Furthermore, using NUBS and PCA improved k-means convergence with no expense to the accuracy of the evaluations. This supports the use of these techniques in a production environment where algorithm run-time considerations are crucial.

It was shown that the calculated number of clusters for each algorithm provided good theoretical results. However, additional segmentation information was extracted by selecting two extra clusters, as suggested by domain expertise. The calculated clusters corresponded well with the expected customer or load groups. Depending on the location and type of customer in an area, the cluster assignments can be confirmed.

It should be noted that the results in this paper were obtained from a data set that represents only summer load behaviour. Therefore, the distance metric selection criteria may change for data sets that include seasonal load diversity. Future work should consider using these results to optimise network planning and to design adaptive tariff schemes in preparation of smart-meter roll out.

#### ACKNOWLEDGMENT

The authors wish to thank Prof. JA du Preez for his inputs regarding the NUBS algorithm, and Eskom for sponsoring this research.

#### REFERENCES

- [1] F. Iglesias and W. Kastner, "Analysis of similarity measures in times series clustering for the discovery of building energy patterns," *Energies*, vol. 6, pp. 579-597, 2013.

- [2] F. M. Zedan, A. M. Al-Shehri, S. Z. Zakhary, M. H. Al-Anazi, A. S. Al-Mozan, and Z. R. Al-Zaid, "A nonzero sum approach to interactive electricity consumption," *Power Delivery*, vol. 25, pp. 66-71, 2010.
- [3] J. Wenzhao, K. Chongqing, and C. Qixin, "Analysis on demand-side interactive response capability for power system dispatch in a smart grid framework," *Electric Power Systems Research*, vol. 90, pp. 11-17, 2012.
- [4] C. Alzate and M. Sinn, "Improved electricity load forecasting via kernel spectral clustering of smart meters," *Data Mining (ICDM) IEEE*, 2013.
- [5] C. Wang and M. D. Groot, "Managing end-user preferences in the smart grid," in *Proc. 1st International Conference on Energy-Efficient Computing and Networking*, 2010.
- [6] C. Kuo, M. Lee, C. Fu, Y. Ho, and L. Chen, "An in-depth study of forecasting household electricity demand using realistic datasets," in *Proc. 5th International Conference on Future Energy Systems*.
- [7] A. Albert and R. Rajagopal, "Smart meter driven segmentation: What your consumption says about you," *Power Systems*, vol. 28, no. 4, pp. 4019-4030, 2013.
- [8] ESKOM, Geo-Based Load Forecasting Standard, 2012.
- [9] T. Liao, "Clustering of time series data—a survey," *The Journal of the Pattern Recognition Society*, vol. 38, pp. 1857-1874, 2005.
- [10] G. Chicco, R. Napoli, and F. Piglione, "Comparisons among clustering techniques for electricity customer classification," *Power Systems*, vol. 21, pp. 933-940, 2006.
- [11] L. Jason and B. Anthony, "Ensembles of elastic distance measures for time series classification," in *Proc. SIAM*, 2014.
- [12] R. Pal, C. Chelmiss, S. Aman, M. Frincu, and V. Prasanna, "Online time series clustering for demand response a theory to break the Curse of Dimensionality," 2015.
- [13] D. De Silva, X. Yu, D. Alahakoon, and G. Holmes, "A data mining framework for electricity consumption analysis from meter data," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, 2011.
- [14] Y. I. Kim, S. J. Kang, J. M. Ko, and S. H. Choi, "A study for clustering method to generate typical load profiles for smart grid," *Power Electronics and ECCE Asia (ICPE \& ECCE)*, pp. 1102-1109, 2011.
- [15] M. Dong, P. C. Meira, W. Xu, and C. Chung, "Non-intrusive signature extraction for major residential loads," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, 2013.
- [16] S. Haben, C. Singleton, and P. Grindrod, "Analysis and clustering of residential customers," *IEEE*, 2015.
- [17] K. Y. Khumchoo and W. Kongprawechnon, "Cluster analysis for primary feeder identification using metering data," in *Proc. 2015 6th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, 2015.
- [18] E. Pan, H. Li, L. Song, and Z. Han, "Kernel-based non-parametric clustering for load profiling of big smart meter data," in *Proc. IEEE Wireless Communications and Networking Conference*, 2015.
- [19] Y. Li and P. P. Wen, "Clustering technique-based least square support vector machine for EEG signal classification," *Elsevier*, vol. 104, no. 3, pp. 358-372, 2011.
- [20] L. Schwardt and J. D. Preez, "PR414/PR813 lecture 5 clustering," Stellenbosch, 2003.
- [21] J. Lines and A. Bagnall, "Ensembles of elastic distance measures for time series classification," in *Proc. the 2014 SIAM International Conference on Data Mining*, 2014, pp. 524-532.
- [22] J. Baarsch and E. C. M., "Investigation of internal validity measures for K-means clustering," in *Proc. the International Multi Conference of Engineers and Computer Scientists*, 2012.

**J. du Toit** and **J.M. Nye** are affiliated to Stellenbosch University where they received a master's degree in engineering. Both are contributing to various academic and industry related conferences and events focusing on optimizing the energy sector by using machine learning.

**R. Davimes**, **A. Mohamed** and **K. Patel** are part-time postgraduate students at the University of Cape Town. They are undergoing various courses and researching applications of machine learning in the financial industry. All authors are currently employed by Eskom – South Africa's power utility.