Feature extraction and clustering of building energy profiles encoded as images

Stepan Ulyanin1, Jose R. Vazquez-Canteli2, Zoltan Nagy2

1 Georgia Institute of Technology, Atlanta, GA, USA
2 Intelligent Environments Laboratory, Department of Civil, Architectural and Environmental Engineering, The University of Texas at Austin, TX, USA

Abstract
Buildings consume roughly 40% of the global energy demand, and an increasing amount of sensor data allows enhanced monitoring and analysis of the building energy systems. One typical application of data-driven analytics is the clustering of daily energy profiles in order to identify buildings that have similar temporal consumption patterns. A popular algorithm for this purpose is K-means. Despite being simple to implement, K-means is a very rigid algorithm and provides no insight on hidden non-linear features of the data, which in many cases are the most valuable. We propose a novel approach to cluster energy profiles using some extracted non-linear features. Given a time series of energy consumption, we transform it into three matrices: (1) a Gramian Difference matrix, (2) a Gramian Summation matrix, and (3) a Markov Transition Field matrix. We then interpret each matrix as the color channel of an image with three color channels. Using a Convolutional Neural Network, we construct a mapping from the encoded time series to a feature space of reduced dimensionality. Further, we apply a Spectral clustering technique within the feature space of the time series, resulting in successful non-linear clustering.

Introduction
Buildings are responsible for 40% of the global energy consumption as well as 30% of the greenhouse gas emissions (UNEP, 2009). Understanding energy consumption patterns of buildings is, therefore, the first step in reducing the CO2 footprint of the building stock. Vast amounts of energy profile data of buildings have become available in recent years and require efficient analysis techniques. Unsupervised learning methods such as clustering are particularly interesting to reveal typical consumption patterns. However, most of these methods require explicit feature definitions which may be both time and resource consuming and may be biased by the preferences of the analyst (Miller et al., 2016).

Energy profile data are represented as time series with the use of time dimension. Typical features are the average energy consumption at a certain hour of the day. In this research, however, we explore how the transformation of the time series into an image, and subsequent analysis using deep learning techniques can reveal information about the profiles and resulting clusters. The role of deep learning has been playing an increasingly important role in many analytical applications, including implicit feature extraction. A typical linear approach to cluster time series data is using the K-means algorithm, which groups together time series minimizing the distance among them in the time domain. However, this type of clustering has its limitations when we want to cluster time-series based on more complex or non-linear patterns that may occur at different times for different time-series. Other, more sophisticated, yet not always robust methods, include recurrent neural networks. However, one of the major drawbacks in using recurrent neural networks for time series clustering or classification is the vanishing or exploding gradients problem (Pascanu et al., 2012). Computer vision methodology allows creating robust architectures for signal processing, avoiding common problems that are present in other deep architectures, such as vanishing gradients. Also, convolutional neural networks allow tractable feature extraction.

In this paper, we propose an end-to-end pipeline for clustering the time series data of energy profiles (see Figure 1). The energy profiles are encoded into three-channel images. Then, computer vision techniques extract features from the encoded time-series followed by clustering. The results, despite being more difficult to interpret, allow the extraction of non-linear patterns in the energy profiles that cannot be extracted using traditional linear clustering techniques such as K-means. This paper is organized as follows. We
introduce the proposed end-to-end architecture in the methodology section. In the results section, we show an analysis of the Pecan Street dataset, using the proposed algorithm. Finally, we address some challenges of the proposed architecture in the discussion section.

### Energy Profile Time-Series Clustering Algorithm

**Result:** Partition over the dataset, learned features of the time-series

1. **Time Series Encoding**
   
   Encode the time-series, using the Gramian Summation and Difference and Markov Transition Fields

2. **Feature Extraction**
   
   2a. Compress the image to a latent space of dimensionality $P$ using convolutional encoder
   
   2b. Obtain the latent space image encoding $Z$
   
   2c. Restore the image from the encoding $Z$ using convolutional decoder

3. **Clustering**
   
   Cluster the images in the latent space, using the encodings $Z$ obtained in 2b

### Methodology

#### Overview

We propose an end-to-end architecture, consisting of an encoding of the time series of the original data to colored images, or tensors of height, width and depth dimensions inspired by (Wang et al., 2015). Then, we use an autoencoder model to learn the latent space manifold of underlying data-generating distribution, simultaneously extracting features from the encoded images. Finally, clustering is performed in the feature space, assigning the samples to corresponding clusters. The proposed algorithm is shown in Table 1.

#### Step 1: Encoding time-series as images

Energy profiles are defined by the load and time of the day, month or year; such information is best represented in the form of classical time series. Wang et al. proposed a method of time-series encoding, using Markov Transition Field (MTF), Gramian Summation Angular Field (GASF) and Gramian Difference Angular Field (GADF). Time-series encoded as images retain most of the information, considering the time domain.

To preserve the temporal dimension, we construct a Markov transition matrix by breaking the values of the time series into quantile bins with the desired granularity. The elements of the resulting matrix are transition probabilities from any given bin to all the other possible bins. Readers are referred to the work of Wang et al. for more detail on the time-series encoding process.

Once we calculate the Gramian and Markov matrices, we stack them along channel dimension to obtain a three-channel image tensor that will serve as inputs and targets to an autoencoder as shown in Figure 1 and depicted in step 1 in Table 1.

Notice that we can reconstruct the original time series from the main diagonal of the Gramian Summation Field matrix. We use these reconstructions to analyze one of the resulting clusters.
Steps 2: Feature extraction
There are a variety of techniques for image processing in the field of deep learning. The most popular one is convolutional neural networks (LeCun et al., 1998) that map an image to the target output. Convolutional neural networks are also translation invariant. Such networks can map our encoded images to output, preserving the temporal dependencies which are encoded in the image by Markov Transition Fields. We propose to use a convolutional autoencoder (Masci et al., 2011) to learn features and underlying manifold of the resulting images.

The under-complete convolutional autoencoder is an unsupervised neural architecture that is most commonly used for dimensionality reduction by mapping an input to a latent space vector of lower dimensionality by applying a series of convolutions to the input image (steps 2a, 2b, and 2c in Table 1).

It is common to use shallow autoencoders in practice; however, we use a deep autoencoder architecture to allow for extensive nonlinear feature learning and mapping. The encoder part of our architecture consists of 5 convolutional layers. The resulting feature maps are then normalized using batch normalization (Ioffe et al., 2015) for faster convergence and training stability.

We use ReLU activations in every layer of both the encoder and the decoder. Also, we initialize filters using the method proposed by Kaiming He (He et al., 2015) to help the network converge.

The loss function is defined as a binary cross-entropy between the real pixels of the image and the pixels of the reconstructed image. However, one might find mean square error to be more effective in some other cases. The autoencoder is trained with the Adam optimizer (Kingma et al., 2015) with a learning rate of 0.001 and no learning rate decay.

Such autoencoder architecture automatically extracts features from an image, eliminating the need for explicit feature engineering, which can be very expensive or even intractable. Also, by reducing the dimensionality of the image, we extract valuable signal by filtering out the noise through a series of convolutions and introducing a latent vector bottleneck, which is an essential element of an under-complete autoencoder.

Additionally, we use the class activation heatmap method known as Grad-CAM proposed by Zhou et al. (Zhou et al., 2016) to map the extracted features to the features in the time domain. Typically, this approach is used to identify how each of the feature maps contributed to the calculation of the outputs in supervised learning algorithms, but we have found it useful in the proposed methodology.

Step 3: Non-linear clustering
The resulting encoded latent vectors are then clustered with the use of the Spectral clustering algorithm (Yu et al. 2003) as depicted in step 3 in Table 1 and shown in Figure 1.

Validation with the MNIST dataset
We performed validation with the MNIST dataset because it is labeled and allows us to evaluate the results of the clustering. Also, any neural architecture is easy and fast to train, using the MNIST dataset.

Note, that we are using the MNIST dataset to validate only a part of our method. The sole goal of the validation with this dataset is to demonstrate the possibility of simultaneous clustering and extensive feature extraction employing an under-complete autoencoder and Spectral clustering algorithm.

Availability of labels allows us to obtain metrics such as completeness and homogeneity scores (Rosenberg et al., 2007) to quantify the results of the clustering.

We start by building an autoencoder to extract the features from the images and to learn the underlying manifold of the data-generating distribution.

The autoencoder requires careful empirical calibration to find the best set of hyperparameters.

We obtain a homogeneity score of 0.9188 and a completeness score of 0.9187 as the result of the Spectral clustering in the latent feature space.

For convenience, we use T-SNE algorithm (Van der Maaten et al., 2008) to be able to visualize the clusters and the data points in 2-dimensional space. Figure 2 illustrates the different clusters obtained after application of our method to the MNIST dataset.

Results
We apply the algorithm to the Pecan Street dataset. Pecan Street dataset is 358 measurements of energy consumption, spanning a whole year. One data point in the dataset is 24 hours of measurements across 15-minute intervals. Therefore, every data point in the Pecan Street dataset consists of 96 measurements.
Step 1: Encoding the time-series into images
First, we transform every time series into three-dimensional encodings, consisting of Gramian Angular Summation (GASF), Gramian Angular Difference Fields (GADF) and Markov Transition Fields (MTF). We encode the data with a granularity of 96 pixels in height and width as we use 24 hours of data per sample with 15-minute snapshots; there are four 15-minute periods in an hour and 24 hours in a day of measurement. An example of a time-series and its Gramian and Markov fields are shown in Figure 3. When we stack these three fields up, we obtain a colored image that we can process with convolutional neural networks. The image can be seen on the left in Figure 4.

GAFs transform time-series to polar coordinates, while preserving temporal relations. The main diagonal of the GAF matrix can be used to reconstruct the original time series. The other components of the matrix provide information of the temporal correlations among the values of the time series. The MTF is constructed after finding the probability of transitioning from one bin to another at different time-intervals. The values on main diagonal are the probabilities that a value of the time-series transitions to itself (that its value does not change in the next time step).

Step 2: Autoencoder
In order to learn the underlying manifold of the data-generating distribution of the Pecan Street dataset, we construct an autoencoder. We use a 5-layer convolutional encoder and a 5-layer convolutional decoder. Figure 4 illustrates one of the images before being encoded and after being decoded. Note that we lose resolution as the result of the non-linear mapping and the bottleneck.

Step 3: Spectral Clustering
The encodings of the images can now be clustered in P-dimensional latent space, using the Spectral nonlinear clustering algorithm. Figure 5 shows one of the clusters of the original time series profiles according to our algorithm.

Step 4 (Optional): T-SNE
The latent space encodings can now be further reduced in dimensionality to a two-dimensional space for visualization purposes. As shown in Figure 6, the scarcity and complexity of data do not yield obvious clusters in the Pecan Street dataset, unlike the MNIST dataset. It is likely that given more data samples, the underlying manifold can be projected in two-dimensional space with more packed looking clusters.
Feature extraction

Our algorithm allows us to cluster complex time-series data points with simultaneous feature extraction.

Note the difference between clusters depicted in Figure 6, where we use the proposed methodology, and Figure 7, where K-Means is applied to the raw time-series data points of the Pecan Street dataset.

K-Means results in slightly better-looking clusters with fewer errors than the clusters obtained by our algorithm. However, it is important that we have not only partitioned the dataset but also extracted all the essential features. The features are shown in Figure 8. Different layers of the autoencoder have learned increasingly abstract features of the original data to make a choice in the clustering routine. In order to visualize the learned features, the extracted features from the image domain are transformed back to the time domain.

Our proposed architecture is based on an unsupervised mapping technique in the instance of an autoencoder. Even though we use elements of models commonly used in unsupervised learning, we can use the resulting encoder and a latent vector to see how each feature map contributed to the calculation of the sample-specific encodings, which directly resulted in the choice of the cluster, using Grad-CAM algorithm.
Figure 7. Clusters of the Pecan Street dataset obtained by applying K-Means clustering to the raw time-series.

Figure 8 shows the features that are extracted from one of the time-series data points of the Pecan Street dataset as they reach the third convolutional layer and become increasingly abstract. The Grad-CAM algorithm makes it possible to transfer some of these features from abstract spaces to the time-domain so we can directly visualize the parts of the time-series that resulted in the act of assigning the data point to a certain cluster.

In Figure 9, we show the parts of the time series that contribute the most to the cluster choices (orange segments). These raw features contribute the most, according to the last (fifth) convolutional layer, to the resulting latent space encoded vector. The vector is then clustered and projected into a two-dimensional space for visualization purposes. Hence, these features implicitly contribute to the choice of the clusters when the autoencoder extracts them.

Note that the extracted features may not be extremely similar when projected back to the time-domain as the direct consequence of the non-linear mapping from the input image space to the latent encoding space.

The extracted features that we observe empirically, however, seem to look quite like each other as shown by orange lines in Figure 9. The peaks are identified in the same time region, so as some changes in the energy consumption around steps 20, 30, 40 and 50.

**Discussion**

By encoding the time-series as images, we can use non-linear feature extraction and clustering techniques to find latent common features that can be used to group various energy profiles. We can form clusters using information from the GAFs and MTF matrices, which provide some insight on the correlations between various bins of the time series and their transition probabilities.

Most of the methods used for time-series clustering are linear, such as K-means. However, it is not always possible to fit a hyperplane through the training set due to the non-linear nature of the data. Using a convolutional autoencoder allows us to extract non-linear features from the data.

In the process of fitting the model, convolutional kernels learn features that are crucial to the clustering choice and can be analyzed further.

The results show how this clustering approach identified similarities in the energy profiles such as frequent oscillations in the afternoon or the peaks at a specific time-step such as 70 as shown in Figure 9. The proposed clustering method is different from the traditional K-means as it is augmented by feature extraction, but its suitability will depend on the specific application.

The limitations of this method rely on the fact that the GAFs and MTFs matrices maintain the temporal relations of the data. Therefore, the resulting clusters are formed considering these temporal relations, and not just the shapes of the time series as other methods such as SAX (Lin et al., 2007). This makes this method advantageous for practical applications that need to account for time-dependencies.

**Potential applications**

This clustering technique could be used for grouping buildings by the usage of different kinds of appliances that have the same patterns of electricity consumption, or by the usage of energy. This kind of clustering technique could also help identify users that follow similar energy consumption patterns that do not necessarily have the same magnitude. However, there are still some open challenges:
Random initialization

The autoencoder algorithm uses random initialization and is an iterative algorithm. The task is to learn the underlying latent manifold of the data-generating distribution.

Every time we fit the autoencoder to our training data, we approximate the posterior distribution of the latent vector conditioned on the input data. The approximation is going to vary from one training session to another, depending on the random initialization of the kernel weights in the convolutional encoder and deconvolutional decoder, due to non-convex nature of the autoencoder loss.

One can reduce the variation in the latent space by using more data; however, in many cases, it is very expensive or not tractable. In this case, it is probable that an ensemble of autoencoders can yield a better approximation of the underlying manifold. This part is left for further research.

Extracted features tractability

The autoencoder possesses an automatic feature selection quality of neural networks architecture. This family of machine learning algorithms can select features that may be intractable.

Convolutional neural networks do not have this problem to the same extent as fully connected networks, although, the number of used filters in a network may affect tractability of features, and some extracted features may overlap.

The features extracted with higher layers of convolutional autoencoder (or network) are less tractable than the features extracted with lower layers as these features become increasingly abstract in deeper layers of the network. Hence the need for feature inspection algorithms such as Grad-CAM.

Conclusion

Understanding energy consumption patterns of buildings is a fast-growing priority for many research fields as it is the first step in reducing the CO₂ footprint of the building stock. Fortunately, data are becoming vastly available in recent years due to the improvement and application of microsensors and other methods of surveying.

Time-series data of energy consumption are widely used for multiple applications. Depending on the application, common linear data clustering techniques cannot efficiently fit and cluster the data. Moreover, some non-linear features of the data may play a more prominent role in the final clustering decision; hence it is essential to learn these features from the data.

Computer vision techniques allow efficient feature extraction through convolution; architectures such as autoencoders are useful in the clustering routine due to the dimensionality reduction. Encoding the time-series data, using Gramian and Markov fields we can transform the data from the time domain into image domain.

We propose an end-to-end architecture that utilizes convolutional autoencoders to extract features and reduce the dimensionality of the data that clusters the data in a non-linear way, which is a superior method to analyze the time-series of energy consumption profiles.

We validate parts of the proposed architecture with the MNIST dataset clustering routine and show results on the Pecan Street dataset.
References


UNEP. Buildings and climate change, summary for decision-makers; 2009.


