Time Series Representation Learning Applications for Power Analytics

Time Series Representation Learning Applications for Power Analytics

Anish Mathew[1] Deepak P[2] Sahely Bhadra[1] Naran Pindoriya[3] Aristides Kiprakis[4] S N Singh[5]
[1]*Indian Institute of Technology Palakkad, India*
[2]*Queen's University Belfast, UK*
[3]*Indian Institute of Technology Gandhinagar, India*
[4]*University of Edinburgh, UK*
[5]*Madan Mohan Malaviya University Of Technology, India / Indian Institute of Technology Kanpur, India*
anishmm1997@gmail.com deepaksp@acm.org sahely@iitpkd.ac.in
naran@iitgn.ac.in Aristides.Kiprakis@ed.ac.uk snsingh@iitk.ac.in

***Abstract*—The uptake of solar power generation is on the rise. This necessitates more research into developing data-driven intelligent methods that can perform effective analytics over power generation data to inform strategies to improve solar power generation systems. In this paper, we consider the utility of time series representation learning for analytics over power generation data. WaRTEm, a representation learning method that focuses on learning time series representations that are invariant to local phase shifts, is the focus of our investigations in this paper. We identify two metadata attributes for power generation sequences, month and CellID, as attributes that embed useful notions of semantic similarity between time series sequences. We evaluate the effectiveness of WaRTEm representations, as against using the raw time series sequences, in alignment to the month and CellID labellings, using accuracy over 1NN retrieval as an evaluation framework. Through empirical evaluations, we identify that WaRTEm embeddings are consistently able to achieve better representations when evaluated on 1NN accuracy. We also identify some features of WaRTEm that are more suited for time series representation learning, which provides promising directions for future work.**

I. Introduction

Recent times have seen large uptake of data analytics methods in the power sector. This includes learning methods for load forecasting [1], security assesment [2], solar radiation forecasting [3] and stability assesments [4]. Most data from power systems, whether it be of power generation from traditional or renewable sources or power consumption from residential or industrial consumers, involve a significant temporal component. These may be best regarded as univariate or multivariate time series data. Time series data has properties that stem from the sequencing within them, that make a number of state-of-the-art machine learning methods inapplicable for them, as we describe below.

A vast majority of machine learning algorithms expect data objects to be represented as feature-vectors. The data vectors or embeddings are expected to be meaningful within the vector spaces that they reside in; in other words, similarities between vectors using reasonable vector similarities/distances (e.g., euclidean) are expected to reflect semantic relationships between data points. The trend of expecting that data objects be meaningful within their vector spaces has only increased with the emergence of deep learning methods where vector spaces are increasingly relied upon.

Consider a simple time series dataset comprising power generation information from a Solar Farm. The quantum of power generated is recorded at regular intervals (e.g., three times each hour), and the natural representation for a day-level power generation from a single solar power generation device would be a univariate vector with the X-axis representing time, and the Y-axis representing the quantum of power generated at each recording interval. We may want to use these power generation vectors to identify solar devices that perform similarly; a significant deviation from the norm could indicate a fault with the generation system, or the recording system. These analytics, like the case of any data-driven analytics, would be best performed over large datasets. Large datasets allow the learning methods to generalize better, and offer more reliable insights. Thus, these time series vectors may be harvested across a number of farms, potentially even nation-wide, and then subjected to centralized analysis. Any such analysis has to be cognizant of a variety of real-world factors that manifest as variations across such data. First, consider a large solar farm stretching over multiple kilometers, such as solar farms over canals. It is well understood that cloud movements play a role in power generation, with cloud cover hampering efficient power generation. The presence of clouds would manifest as a dip in power generation, and this dip would appear at different regions of the time axis in different devices based on when the cloud affected the device's power generation. These *phase shifts* across devices caused by cloud movements and related phenomena would cause very similar devices be judged to be different if lock-step measures such as Euclidean distances are used to compare time series power generation data from them. Second, there is a different source of phase variation when comparing power generation time series information from across different states and regions. For example, despite being in the same time zone, the power generation peak for a solar farm in Gujarat may be expected to be much later

978-1-7281-3192-4/19/$31.00

in the day as compared to those for a farm in West Bengal, due to widely varying longitudes, and thus the *'real'* time at those locations. Any method to assess time series similarities should be conscious of such possible phase variations. This has been recognized widely in time series analytics from other sectors; similarity measures that are robust to phase shifts such as dynamic time warping (DTW) have been regarded as among the better methods to compare time series sequences [5].

The need for bespoke similarity measures such as DTW for usage within time series analytics due to aforementioned reasons has posed practical difficulties in applying deep learning methods over them. A workaround or another line of attack for this issue is that of using a representation learning method to generate vector representations for time series such that conventional distance measures over such vectors (such as euclidean distance) would mirror meaningful distances over the original time series representations. In this paper, we consider one such recent method, Warping Resilient Time Series Embeddings (WaRTEm [6]), and its applications to analytics applications over time series data of power generation from a large solar farm located in IIT Gandhinagar.

Roadmap: We star by outlining related recent research in Section II. WaRTEm, the representation learning method that we use in this study, is described in Section III. In Section IV, we describe the scenario of power generation data, and describe how representation learning could be evaluated within the setting. Section V describes our empirical evaluation, which is then followed by conclusions and future work.

II. Related Work

We cover related work to this research from two different directions: (i) machine learning approaches for representation learning, and (ii) analytics applications over power data.

A. Representation Learning

The performance of machine learning models is very much dependent on the data that is fed to them. This makes the effort of ensuring effective representations for the downstream machine learning algorithms a critical phase in any machine learning application. With the emergence of deep learning frameworks for machine learning, the scholarly community engaged in representation learning has focused on learning vector representations, often called embeddings, vector embeddings being compatible with deep learning pipelines. Interest in representation learning has increased over the last many years, reflecting in various ways, one of which is through the establishment of a new top-tier computing conference called the international conference on learning representations (ICLR)[1]. Bengio et al. [7] provide an excellent survey of representation learning techniques in the literature over the last many years. Given that different types of data (e.g., images, text, time series) involve different kinds of information, bespoke type-specific representation learning methods are necessary. Examples of such bespoke representation learning methods include word2vec for text data [8] and representation learning for player trajectories [9] and question-answer pairs [10]. Representation learning for time series has also been explored in recent times. Lei et al. [11] propose a time series representation learning method that uses matrix operations guided by dynamic time warping [12]. Two other recent techniques, [13] and [14], use convolutional neural network architectures in order to learn representations for time series. In this paper, we make use of a recent time series representation learning method, WaRTEm [6] that is specifically targeted to obtain embeddings that are resilient to *warpings* i.e., local stretching and compressions of time series, in this work. We believe that such warping resilience would be critical for time series representations for power analytics.

[1]https://iclr.cc/

B. Analytics in Power Generation

Analytics over power generation data has largely focused on forecasting; within forecasting, the focus has rightly been on short-term forecasting [15], with long-term forecasting more dependent on weather patterns than historical power generation behavior. Outside the realm of power generation, there has been a large body of literature on applying data analytics techniques to power consumption data. With the adoption of smart meters that makes fine-grained power consumption data pervasive, big data analytics has seen much uptake in this area; we briefly outline two representative directions that employ techniques from big data analytics. First, using smart meter data in order to do load estimation and profiling has been an area of much interest. Techniques for this task include clustering of households [16], missing data estimation [17] and clustering at the feeder level [18]. A second busy stream of research has been on non-technical loss estimation, a task that is critical to developing countries where such losses pose big concerns. Techniques for this task include anomalous behavior detection boosted by supervised learning [19] and extreme learning machines; a survey of such methods appears at [20]. To our best knowledge, there has not been any research into using time series representation learning methods for power analytics tasks.

III. WaRTEm: Warping Resilient Time Series Embeddings

We now briefly outline WaRTEm [6] emphasizing on aspects which we believe are suitable for analytics applications over power generation data. WaRTEm is a representation learning method that uses a deep learning architecture for unsupervised representation learning. The core learning task is to take a dataset of time series sequences, $\mathcal{T} = \{\ldots, t, \ldots\}$ and convert it into a set of vectors in a $\mathbb{R}^d$ dimensional space where d is a parameter specifying the desired dimensionality of the representation space.

$$\mathcal{T} = \{\ldots, t, \ldots\} \xrightarrow{\text{WaRTEm}} \mathcal{U} = \{\ldots, u, \ldots\} \quad (1)$$

The above transformation uses information across time series sequences in $\mathcal{T}$ in order to learn a d-dimensional vector for each time series. We describe the main idea behind WaRTEm and then outline the deep learning architcture that is employed by it.

A. Local Variations

The core idea behind WaRTEm is that the eventual representations in $\mathcal{U}$ should be robust to small phase changes in the time series sequences. In other words, consider taking a time series sequence $t \in \mathcal{T}$ and making a slight variation such as shifting a power generation peak slightly rightward. The original and the transformed time series should map to very nearby, if not identical, points in the d-dimensional space under the WaRTEm transformation. This notion forms the backbone upon which the representation learning process in WaRTEm is structured. Towards realizing the notion of *slight shifts*, WaRTEm uses two kinds of *transformation operators*, each of which can be instantiated on either *left* or *right* directions.

1) Copy Warping:: Consider a window of four points from a time series t denoted as $[p_1, p_2, p_3, p_4]$; this is the warping focus window and the transformation will be localized within this window. The *left* and *right* copy warping operators are illustrated as below:

$$Left : [p_1, p_2, p_3, p_4] \rightarrow [p_1, p_3, p_4] \rightarrow [p_1, p_3, p_4, p_4]$$

$$Right : [p_1, p_2, p_3, p_4] \rightarrow [p_1, p_2, p_4] \rightarrow [p_1, p_1, p_2, p_4]$$

The *LeftCopyWarp* (LCW) operator deletes off p_2 from the window and duplicates p_4 to restore the length, whereas RCW deletes off p_3 and duplicates p_1. In other words, LCW shrinks the left side of the window and extends the right endpoint to a plateau, whereas the vice versa is the case for RCW. It may be noted that a time series t and it's warped variant $LCW(t)$ or $RCW(t)$ differ only in the values that they take within the warping focus window.

2) Interpolation Warping:: Consider a warping focus window $[p_1, p_2, p_3, p_4]$ as earlier. The variants of the interpolation warping are as follows:

$$Left : [p_1, p_2, p_3, p_4] \rightarrow [p_1, p_3, p_4] \rightarrow [p_1, p_3, \frac{p_3 + p_4}{2}, p_4]$$

$$Right : [p_1, p_2, p_3, p_4] \rightarrow [p_1, p_2, p_4] \rightarrow [p_1, \frac{p_1 + p_2}{2}, p_2, p_4]$$

The LIW operator, as in the case of LCW, shrinks the left side of the window, but then extends the right side by a slope (as against a plateau in LCW). This slope is formed by adding a point that is midway between p_3 and p_4 both in terms of its value and placement. RIW is simply the mirror image of LIW.

3) Illustration of the Warping Operators: Figure 1 illustrates examples of series formed by two of our warping operators to help visualize the changes effected by them.

B. WaRTEm Approach

The WaRTEm neural network architecture comprising twin auto-encoders (AEs) is illustrated in Figure 2. It is modelled to take a pair of time-series sequences as input. Each time series sequence in the input pair is passed through a separate convolutional auto-encoder (shown side by side in Fig. 2). The encoder part of the AE comprises a sequence of pairs of 1-d convolution and maxpooling layers followed by a final fully connected layer, whereas the decoder analogously uses upsampling and pairs of 1-d convolution layers. As is typical of auto-encoders (AEs), the respective time series get

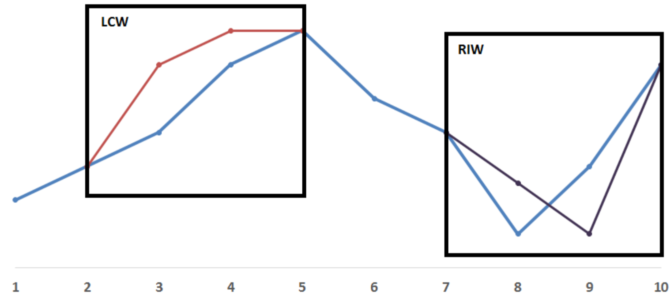

Fig. 1. Warping Operators Example: The warping operators LCW and RIW are illustrated by the changes they effect within the warping windows. Blue indicates the original time series, with others the respective warped versions.

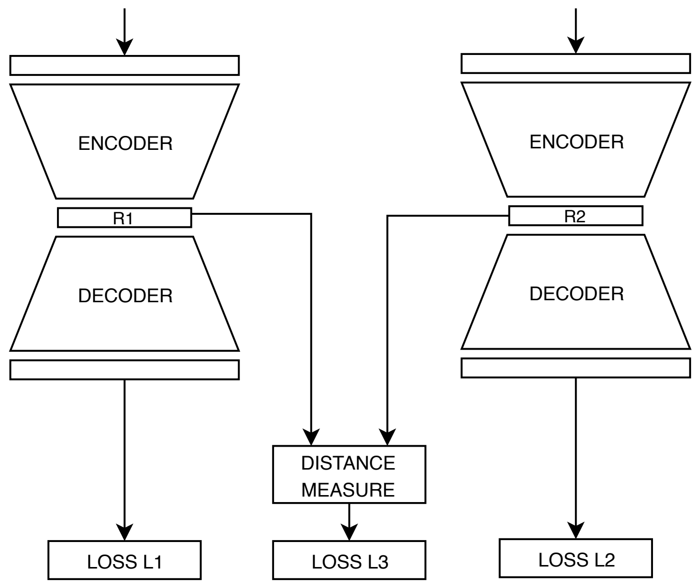

Fig. 2. Twin Auto-encoder Architecture used in WaRTEm

converted into an internal representation (aka code) through the encoder, with the decoder expected to re-construct the original input to high accuracy from the code. $L1$ and $L2$ indicate the conventional reconstruction losses for the separate AEs. The linkage between the AEs is achieved through the introduction of a new loss term, $L3$ which is designed as the squared euclidean distance $\|R_1 - R_2\|_2^2$ between the codes (R_1 and R_2) corresponding to the pair of input time series. To learn embeddings that cater to a different similarity measure than euclidean, WaRTEm can be adapted by designing a corresponding loss term between R_1 and R_2. As indicated, $L3$ is propagated back through the encoder parts of the AEs, and does not affect the decoder weights. In other words, in addition to training the separate auto-encoders to reconstruct their respective inputs, we also try to ensure that the codes corresponding to the time series pairs are close to each other. The way this maps to our intent of learning time series embeddings robust to local variations will be outlined in our warping-based training strategy below.

The training strategy outlines the manner in which we use the time series dataset $\mathcal{T}$ in training our twin AE architecture. The twin AE architecture is motivated by our observation based on empirical studies that *warping resilience* is quite complex for a single AE to learn (using, for example, a variant of denoising AE). Thus, we specialize the task to two, viz., *leftward* and *rightward* warping resilience, so separate AEs can learn them separately.

For each time series $t \in \mathcal{T}$, for each choice of warping focus window and for each directionality of warping (*left* or *right*), we generate a warped variant. Consider the choice of *left* direction; we first sample a random integer r between 0 and $(0.5 \times length(T))$. We then progressively perform r leftward warpings over randomly chosen warping focus windows, choosing LCW or LIW, to generate a warped variant of t, denoted as $L(t)$. The pair $[L(t), t]$ thus generated forms an element of the training dataset for our twin AE. Analogously, the choice of *right* direction yields a warped variant $R(t)$, forming a training pair $[t, R(t)]$. It may be noted that each t thus generates two training pairs for each choice of warping operator, one for *left* and another for *right*, and many pairs can be generated based on the number of warping focus windows chosen. The construction of the ordering in the pairs is pertinent to the separation of warping resilience learning; the left entry in the pair is either a left-warped variant or the original series, but never a right-warped variant (and similarly for the right entry). This creates a warping directionality co-ordination between the AEs which helps separate the nature of learnings within the respective AEs.

The training process is continued for as many epochs as needed; WaRTEm uses a held-out dataset to compute loss trends across epochs to effect early stopping. Then, each time series t is passed through each of the AEs separately to generate their left and right AE codes, which are then averaged to be used as the corresponding embedding u. This completes the description of WaRTEm transformation from time series to embeddings.

IV. Analytics Applications over Power Generation Time Series Data

Towards illustrating the value that WaRTEm embeddings brings to power generation time sequences from a solar power plant, we outline two tasks that will form the framework of our empirical evaluation. Consider that $\mathcal{T}$ is a large dataset of time series sequences. Without loss of generality, let each sequence in $\mathcal{T}$ be a sequence of 72 numbers, each number recording the quantum of power generated from a specific solar cell unit for a duration of 20 minutes in the day. In other words, the first number in the sequence records the amount of power generated between midnight and 0020 hours, whereas the 20^{th} number in the sequence maps to the quantum of power generated between 0620 and 0640 hours. Thus, each time series sequence is associated with two kinds of metadata:

- *Solar CellID:* The ID of the solar cell unit that generated the power generation time series sequence.
- *Day:* The date during which the sequence was recorded. This uniquely identifies a day using the date, a combination of day, month and year.

From the point of view of emergy analytics, we would consider a representation for power generation time series sequences to be good if it broadly agrees to the following principle: *time series sequences that are semantically similar should appear close to each other in the eventual representation*.

The notion of *semantic similarity* often depends on the concrete task that the representations are used for. Ideally, this would involve harvesting a dataset where time series sequences are labelled as either *similar* or *dissimilar*, and verifying whether the representations learnt (as may be obvious, the representation learner would not have access to the labellings) agree to the labellings, i.e., pairs labelled to be similar have their representations much closer to each other than pairs labelled to be dissimilar. Given the practical difficulties in organizing such a labelling effort as well as the expertise needed to perform the labelling (the expertise threshold can be reduced by using a more fine-grained labelling scheme than a binary one, but the expertise requirement remains very high), we make use of the time series metadata in order to outline empirical evaluation mechanisms for [WaRTEm representation + euclidean distance] combination over the [time series sequences + euclidean distance] combination. These are outlined in the subsections below.

A. CellID as Semantic Similarity

The power generation profile of a solar cell unit, identified using the CellID metadata attribute, says a lot about the power generation profile that would be associated with the unit. For example, an old or dusty power generation unit would have lower power generation activity as compared to a newer one. Thus, power generation time sequences that are generated by the same unit, everything else remaining the same, can be regarded as being more semantically similar than sequences that are generated by different units. Thus, two time series sequences having the same CellID can be regarded as being labelled as *similar*.

B. Month of the Year as Semantic Similarity

The power generation profile changes with weather conditions. The month of the year is a fairly decent predictor of the average weather condition during the days in the month. For example, the dataset that we obtained was from Gujarat (India) which sees good sunshine during the summer months yielding efficient power generation for Solar cells, whereas the rainy season (june-july) and the winter affects the power generation in different ways. Considering the month of the year as a signal of semantic similarity, we can regard two time series sequences associated with the same month as being labelled as *similar*.

C. Empirical Evaluation using 1NN Retrieval

We would like to evaluate the WaRTEm representations for agreement to the above two notions of semantic similarity. We adopt the following framework to arrive at a quantitative way of estimating agreement to a chosen measure of semantic similarity.

- For each representation vector $u \in \mathcal{U}$
  - Identify the most similar vector to u from $\mathcal{U}$ based on euclidean distance; this is the first nearest neighbor (aka 1NN) to u
  - Check whether the nearest neighbor shares the same semantic similarity label (e.g., CellID or month, whichever is the choice) and record it as *accuracy(u)* (1 if they share the same label, 0 if not)
- Estimate the average of accuracy(u) across vectors in $\mathcal{U}$ as accuracy($\mathcal{U}$)

The above quantification scheme allows us to compare WaRTEm with other representations and similarity measures such as euclidean distance over the raw time series sequences. For achieving the accuracy number for the raw time series sequences, we use the same framework as above, but over the original time series sequences. A comparison between the accuracies helps indicate the relative merit of the two representations in agreeing to notion of semantic similarity defined using months or devices.

CellID as Similarity Label	
Method	*Accuracy*
WaRTEm + Euclidean	**46.00**
Time Series Sequences + Euclidean	44.74
Month as Similarity Label	
Method	*Accuracy*
WaRTEm + Euclidean	**80.00**
Time Series Sequences + Euclidean	74.85

TABLE I
1NN RETRIEVAL RESULTS

CellID as Similarity Label	
Method	*Accuracy*
WaRTEm Full	46.00 (1.16)
WaRTEm Copy	**46.00 (0.96)**
WaRTEm Interpolation	46.00 (1.3)
Month as Similarity Label	
Method	*Accuracy*
WaRTEm Full	80.00 (2.88)
WaRTEm Copy	**81.00 (2.32)**
WaRTEm Interpolation	80.00 (2.56)

TABLE II
WARTEM OPERATOR ANALYSIS

V. EXPERIMENTS

We first outline our dataset followed by the results from the 1NN evaluation outlined in Section IV-C.

A. Dataset

For the preliminary analysis that will form this paper, we sourced a dataset from the IIT Gandhinagar solar power plant. Gujarat experiences warm summers when the power generation peaks consistently, with reasonably cold winters. In order to pose a challenging task for the retrieval evaluation, we considered the subset of the data from the summer months, viz., June, July, August and September. We expect this to be challenging since differentiating months that have widely different weather patterns (e.g., telling apart *July* from *December*) would obviously be easy. We collected the power generation information from across 5 solar cell units, across these 4 months, yielding a total of 485 day-level power generation sequences. Four of the CellIDs had 119 time series sequences each, whereas the fifth had only 9 sequences. Among the months, we had 121 time series sequences from *June*, 116 from *July*, 127 from *August* and 121 from *September*. The variations between devices and months are due to missing data that could be due to one of many causes including technical faults. Each power generation sequence has 72 data points, each point corresponding to the quantum of power generated during a 20 minute interval in the day. WaRTEm allows to specify the desired dimensionality of the output representations. We choose 14 as the output dimensionality, which approximately correlates with 20% of the size of the original representation (recall that we start with 72 length vectors).

B. Experiment Results

The accuracy comparisons across the 1NN retrieval tasks are outlined in Table I. As may be seen from the results therein, the WaRTEm + euclidean combination is seen to be more effective than doing euclidean distance based dissimilarities over the raw time series themselves. There were 5 different CellIDs, and four different months. Given the slightly smaller domain size of months, it is intuitively easier to obtain higher accuracies on the month labellings. However, the wide difference between the accuracies in month and CellID evaluations indicates that both representations find it significantly easier to separate out the months as against the different CellIDs. This may be regarded as intuitive given that the CellIDs do not differ significantly in their inherent make-up, making different CellIDs' power generation sequences expected to be similar. It is also notable here that the different CellIDs were in different parts of the campus, but in reasonably proximal locations, thus climatic conditions such as the cloud cover etc. are expected to affect them in similar ways. From Table I, it may be seen that the accuracy improvements achieved by WaRTEm are much more significant for months (5+ percentage points) as compared to CellID (1+ percentage points).

C. Different Interpolation Operators in WaRTEm

In exploring the utility of WaRTEm for tasks in power generation analytics, we now analyze the impact of the two different kinds of warping operators that WaRTEm uses in order to create *local variations* of time series sequences. Apart from the full WaRTEm model (used in Section V-B), we also additionally train it separately using just the *copy* and just the *interpolation* warping operators. The results of this analysis are outlined in Table II. From this analysis where the accuracy numbers are indicated along with the standard deviations, we draw two observations. First, it is clearly visible that copy operators are more effective in learning representations for power generation sequences, given that it produces either higher accuracies (as in the case of the *month* experiment) or smaller standard deviations or both. Second, the interpolation operator by itself achieves similar accuracies as in the case of copy, but may be considered inferior due to the higher standard deviations. When used in combination with interpolation operator, it is seen to dampen the accuracy as against just using interpolation operators. This differential impact of copy and interpolation operators in representation learning for power generation sequences needs to be subject to further empirical evaluation over perhaps a larger dataset.

D. Discussion

Our empirical analyses establish that WaRTEm representations are more suited for power analytics applications than the raw time series sequences. This also needs to be seen in the context of the relative sizes of the representations; WaRTEm is able to achieve the gains in $1NN$ accuracy using a representation size that is only 20% of the size of the original time series sequences. Our analysis also underlines that copy warping operators are better suited for representation learning over the power generation sequences.

VI. Conclusions and Future Work

In this paper, we investigated the utility of time series representation learning, specifically a recent representation learner targeted towards generating warping resilient embeddings (WaRTEm), for applications in power analytics. In particular, we considered daily power generation sequences from a solar power plant as time series sequences and generated WaRTEm vector representations that are 20% of the size of the original data representation. We outlined two concrete evaluation tasks motivated by analytics applications in the power domain, both quantifying the agreement of the representations to semantic labels that are external to the time series themselves. Our choice of *month* and *CellID* as semantic labels helps derive quantiative metrics measuring the alignment of representations against them. Based on our empirical analysis, we find that WaRTEm representations are significantly more semantically aligned when measured against *month* and *CellID* labellings. In particular, WaRTEm representations record up to 5 percentage points in $1NN$ accuracy improvements over using the raw time series sequence representations. We also illustrated, through an empirical analysis, that the WaRTEm *copy* warping operators are more suited to representation learning over our power generation sequences than the *interpolation* warping operators.

A. Future Work

Our empirical insight that copy warping operators are better suited for representation learning over power generation sequences indicates that it would be useful to explore designing more operators with a similar flavor, to further improve the quality of the representations. The full utility of robustness to local variations can only be tested with more exhaustive empirical evaluation over large datasets spanning a larger geographical region. We will look to perform such analyses as future work. At present, WaRTEm, being a general purpose representation learning method for time series sequences, does not make use of the day-night cycle which reflects in power generation sequences as a peak around mid-day. Designing bespoke operators that embed such domain information within them would be a very promising research direction.

VII. Acknowledgements

Part of this work was supported by UKIERI-DST through the D-DIEM project[2].

References

[1] S. Li, L. Goel, and P. Wang, "An ensemble approach for short-term load forecasting by extreme learning machine," *Applied Energy*, vol. 170, pp. 22–29, 2016.

[2] N. V. Tomin, V. G. Kurbatsky, D. N. Sidorov, and A. V. Zhukov, "Machine learning techniques for power system security assessment," *IFAC-PapersOnLine*, vol. 49, no. 27, pp. 445–450, 2016.

[3] C. Voyant, G. Notton, S. Kalogirou, M.-L. Nivet, C. Paoli, F. Motte, and A. Fouilloy, "Machine learning methods for solar radiation forecasting: A review," *Renewable Energy*, vol. 105, pp. 569–582, 2017.

[4] B. Wang, B. Fang, Y. Wang, H. Liu, and Y. Liu, "Power system transient stability assessment based on big data and the core vector machine," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2561–2570, 2016.

[5] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, 2017.

[6] A. Mathew, S. Bhadra *et al.*, "Warping resilient time series embeddings," *arXiv preprint arXiv:1906.05205*, 2019.

[7] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[9] Z. Wang, C. Long, G. Cong, and C. Ju, "Effective and efficient sports play retrieval with deep representation learning," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019.*, 2019, pp. 499–509. [Online]. Available: https://doi.org/10.1145/3292500.3330927

[10] D. P, D. Garg, and S. K. Shevade, "Latent space embedding for retrieval in question-answer archives," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 2017, pp. 855–865. [Online]. Available: https://www.aclweb.org/anthology/D17-1089/

[11] Q. Lei, J. Yi, R. Vaculin, L. Wu, and I. S. Dhillon, "Similarity preserving representation learning for time series clustering," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 2845–2851.

[12] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and information systems*, vol. 7, no. 3, pp. 358–386, 2005.

[13] D. Shanmugam, D. Blalock, and J. Guttag, "Jiffy: A convolutional approach to learning time series similarity," 2018.

[14] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi, "Unsupervised scalable representation learning for multivariate time series," *arXiv preprint arXiv:1901.10738*, 2019.

[15] G. K. Venayagamoorthy, K. Rohrig, and I. Erlich, "One step ahead: short-term wind power forecasting and intelligent predictive control based on data analytics," *IEEE Power and Energy Magazine*, vol. 10, no. 5, pp. 70–78, 2012.

[16] F. McLoughlin, A. Duffy, and M. Conlon, "A clustering approach to domestic electricity load profile characterisation using smart metering data," *Applied energy*, vol. 141, pp. 190–199, 2015.

[17] A. Al-Wakeel, J. Wu, and N. Jenkins, "K-means based load estimation of domestic smart meter measurements," *Applied energy*, vol. 194, pp. 333–342, 2017.

[18] A. K. Dashtaki and M. R. Haghifam, "A new loss estimation method in limited data electric distribution networks," *IEEE Transactions on Power Delivery*, vol. 28, no. 4, pp. 2194–2200, 2013.

[19] M. M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, and A. Gómez-Expósito, "Detection of non-technical losses using smart meter data and supervised learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2661–2670, 2018.

[20] G. M. Messinis and N. D. Hatziargyriou, "Review of non-technical loss detection methods," *Electric Power Systems Research*, vol. 158, pp. 250–266, 2018.

[2] https://d-diem.eng.ed.ac.uk/