Multi-Label Auto-Encoder based Electrical Load Disaggregation

Spoorthy Paresh
TCS Research and Innovation
Bangalore, India
s.spoorthy@tcs.com

Naveen Kumar Thokala
TCS Research and Innovation
Bangalore, India
naveen.thokala@tcs.com

Angshul Majumdar
IIIT Delhi
New Delhi, India
angshul@iiitd.ac.in

M Girish Chandra
TCS Research and Innovation
Bangalore, India
m.gchandra@tcs.com

***Abstract*—Load Disaggregation has gained much popularity in the recent times, owing to the advantages it brings to energy utility companies. Many modeling techniques ranging from Dictionary Learning to HMM-based techniques to Neural Network based modeling have been proposed in the literature to solve this problem. However, scalability and computational lightness, have been two main areas of concern associated with the problem modeling. In this work, the authors propose to use Multi-Label Auto-Encoder architecture to solve this problem. The proposed architecture incurs minimum instrumentation cost, which makes it truly non-intrusive. The use of superposed appliance class labels of interest in the discriminative penalizing term of the architecture, ensures that disaggregation is achieved without the need to train a separate model for each appliance class of interest.**

***Index Terms*—Load Disaggregation, Smart Meter, Multi-Label Auto-Encoder.**

I. Introduction

Smart meters roll-out in the recent years has enabled energy sector to move towards digitization. Smart meters are designed to collect total power consumption data at a pre-defined sampling rate, that typically ranges from one second to one hour. Smart meter data is used to obtain appliance-level energy consumption at installed locations, using source-separations methods to disaggregate the smart meter data into the corresponding appliances loads. This process is termed as Load Disaggregation. Disaggregated information such as operation of a) schedul-able loads (Washing Machines, Dish Washers, Dryers etc.), b) Electric Vehicles (EV) that act as a source (negative load) and, c) elastic loads (like air conditioning) etc. is vital to energy-utility companies as this would enable them to (i) design optimal demand response for targeted customers and (ii) to provide the customers with energy saving recommendations that may also include replacing inefficient appliances, if any. Load disaggregation further helps customers to have the know-how to optimize the usage of various appliances..

Load Disaggregation (LD) has been a subject of research for over two decades now and has started with Hart's [1] work where he used events in the total consumption to detect the individual appliances. Since then, Load Disaggregation has gained significant momentum. Load Disaggregation can broadly classified into (i) Event-based load disaggregation and (ii) Non-event based load disaggregation. Event based LD is carried out using ON/OFF events of the appliances [10] [12] and some times can include other discriminating features like *time of operation*. Event-based LD is applicable for smart meter data with sampling frequencies up to 1 minute where events could be observed whenever the appliance switched ON/OFF. For sampling frequencies greater than 1 minute, the problem is to map the aggregate power data onto the combination of loads operational in the window of observation using statistical optimization methods. This usually is termed as *Non-Event* Load Disaggregation problem.

Load Disaggregation problem, in this category, received great attention, owing to the fact that while the total power consumption could itself be a linear superposition of appliances, the aggregate power consumption at any point in time, cannot be mapped to an unambiguous set of appliance-combinations. To this effect, several optimisation methods and neural network based methods were studied and proposed, ranging from Sparse Coding methods [6], [8] to HMM -based modeling techniques [9], to Deep Learning based architectures [11].

Kelly et al. [11] proposed deep architectures using Denoising Auto-Encoders (DAEs) and Long Short-term Memory (LSTM) neural network for load disaggregation where there is seperate network trained for each appliance of interest considering aggregate load as input and appliance consumption pattern as output. Sparse Coding based source separation algorithm was proposed by Kolter [8], in which he considered hourly consumption data to develop sparse models for each appliance class so as to solve the problem using non-negative sparse coding technique.

However, the authors' have in their previous works formulated Dictionary-Learning based disaggregation framework [6] and Neural Network based method [10] and observed that these methods often incur huge computational costs, due to the large data requirement during the training phase. Therefore these methods are not practically feasible as a viable business solution. It is evident that for a real life solution, the Non-Event Load Disaggregation must, most importantly, possess the ability to extract loads from the aggregate data, in a computationally simpler manner. This implies that the formulation for training of the appliance class models requires simplification, which would in turn, reduce the amount of data required to solve the problem. This is important in the

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

area of Load Disaggregation, as labeled or annotated smart meter data is often difficult to acquire, while unlabeled data is easily acquired through a smart meter installation. This further implies that the solution should be computationally light and scalable.

To this effect, a recent emerging line of research using semi-supervised learning approaches has come to the authors' attention. Semi-supervised learning methods offer the flexibility to learn patterns from the aggregate data in an unsupervised manner in the first step, which are then used by a discriminative model to classify or identify the patterns in the window of observation. Further, the discriminative part of the model does not require extensive formulation, and therefore is easier to handle.

A semi-supervised approach that instead combines the discriminative ability with the representational stage of an Auto-Encoder [5] had come to authors' attention. In this work, the authors [5], inspired by the results of training neural-network based architectures using Alternating Direction Method of Multipliers (ADMM) in [18], have used this method to obtain the discriminant features of the data at the decoding layer of an Auto-Encoder framework. It may be noted that ADMM has been demonstrated to produce more efficent and stable results as opposed to those produced by stochastic gradient descent method on large data-sets [18].

While auto-encoders have been used earlier for feature extraction and for single-label classification in the literature, using Auto-Encoders for multi-label classification has been first reported in [5]. Termed as *Discriminative Auto-Encoders (DiAE)* in their work, this framework has been evaluated for character recognition and inspired by the results in this paper we propose to solve the problem of Load Disaggregation using this framework.

To the best of the author's knowledge, this has been the first attempt to use Discriminative Auto-Encoder (DiAE) network for the task of load disaggregation. Load Disaggregation is a more complex task, as compared to character recognition, in that it belongs to the class of source-separation problems which makes the mapping of the aggregate data onto the corresponding loads of interest challenging as mentioned earlier in this section. Therefore, the performance of DiAE in this case is worth examining. In this work, the DiAE framework is used as follows: obtain the encoded representations of the aggregate data, map the representations to the decoder layer and to the appliance class labels matrix in an alternating manner as detailed in [5], [18]. This enables the decoding layer of the Auto-Encoder to act as a discriminator to obtain the disaggregated results.

We evaluate the performance of this framework on REDD data-set, and further compare these results with those obtained by applying Multi-Label KNN (ML-KNN) and RAKEL algorithms on the same data-set.

II. Multi-Label Auto-Encoder

The Multi-Label Auto-Encoder Model involves a discriminative penalising term in the representation-learning formulation. The discriminative ability is obtained at the decoding layer by learning a linear mapping between the representations of the input data and the class labels of interest. This formulation results in a simplified architecture that ensures the discriminative ability is learnt via a single formulation. Authors' in [5] propose the DiAE design as shown in the Figure 1.

The figure also demonstrates how the architecture would be used in the current work. The class labels in our work (*D* as mentioned in the Figure 1), is appliance label matrix, consisting of the ON-OFF state information for all appliances of interest for a window of observation. As we can observe, the class label matrix *D* consists of superposed information at any point in time unlike a one-hot encoding vector used in [5].

To the best of the authors' knowledge, such an approach was not attempted earlier to the problem of load disaggregation. By using DiAE, we intend to map the aggregate power consumption data, via the encoded representations, to the appliance class matrix. Using a series of updations as demonstration in section II-A, we finally achieve our goal of drawing the requisite mapping between the input data representations and the appliance labels. This way, given an aggregate data input, we will be able to identify the appliances present in it. The formulation for DiAE is detailed in section II-A.

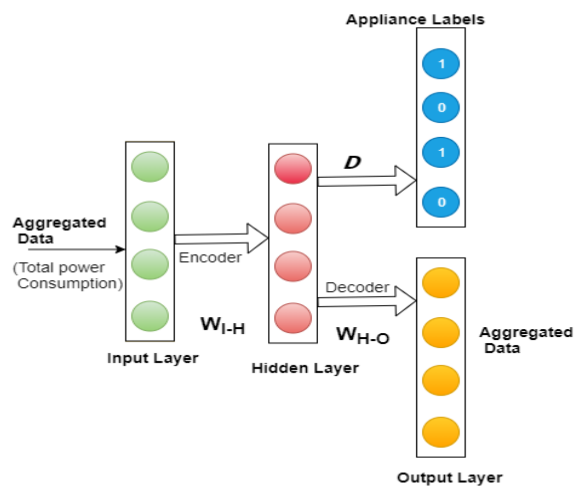

Fig. 1. Block Diagram for DiAE

A. Training

A basic autoencoder encodes the aggregate data and yields the latent representations of the same as:

$$h = \phi(W_1 x) \quad (1)$$

Here, W_1 refers to the weights of the nodes of the encoding layer and h refers to the representations learnt from x and ϕ is the activation function.

The decoding layer of this autoencoder would reconstruct the input layer back from the representations h as:

$$x = W_2 \phi(W_1 x) \quad (2)$$

During training, encoding and decoding weights are learnt by minimizing the distance between the input data, x and reconstructed output $(W_2\phi(W_1x))$:

$$\|X - W_2\phi(W_1x)\|_F^2 \tag{3}$$

In DiAE, the linear mapping D, between the representations and the class labels of interest is also learnt. This enables discriminative ability at the decoder layer. This can be formulated as follows:

$$\underset{W_2,W_1,Z,D}{\operatorname{argmin}} \|X - W_2\phi(W_1X)\|_F^2 \ + \ \lambda\|L - DZ\|_F^2 \tag{4}$$

However, in order to avoid the learning of encoder weights, W_1 and encoder representations, Z simultaneously, a third term $Z - \phi(W_1X)$ is introduced. This brings us to the final formulation for DiAE as:

$$\underset{W_2,W_1,Z,D}{\operatorname{argmin}} \|X - W_2\phi(W_1X)\|_F^2 + \lambda\|L - DZ\|_F^2 + \mu\|Z - \phi(W_1X)\|_F^2 \tag{5}$$

where L indicates the class labels and Z indicates the encoder representations.

Clearly, this formulation is the Augmented Lagrangian or ADMM formulation, discussed in [18]. We break this formulation into four steps, in order to obtain the updations for W_2, D, W_1 and Z as explained in Algorithm 1. We implement each step as a least square minimization problem, akin to the way it is implemented in [5].

Algorithm 1: The DiAE least square minimization steps to compute W_2, D, W_1

$Step1 : W_2 \leftarrow \operatorname{argmin}_{W_2} \|X - W_2\phi(W_1x)\|_F^2$
$Step2 : D \leftarrow \operatorname{argmin}_D \|L - DZ\|_F^2$
$Step3 : W1 \leftarrow \operatorname{argmin}_{W_1} \|Z - \phi(W_1x)\|_F^2$
$Step4 : T \leftarrow \operatorname{argmin}_Z \|X - W_2\phi(W_1X)\|_F^2 + \|L - DZ\|_F^2 + \|Z - \phi(W_1X)\|_F^2$
$Step5 : H \leftarrow T = |T|$

III. Implementation

A. Data Sets

In our work, we have used REDD dataset, to train and test the formulation [7]. The data set consists of data for six houses. The aggregate and the appliance level readings were measured at the sampling frequency of every 3 seconds and 1 second respectively. The data from houses from 2 to 4 were used for training and is tested on house 5.

Choice of Appliances: A combination of high power and low power consuming appliances were used in our work, to validate the proposed approach. We chose washer-dryer, refrigerator, dishwasher and an oven as appliances of interest, as some of these appliances (washer-dryer, oven) are high-power consuming appliances while some others (refrigerator, dishwasher, oven) are highly used appliances in most of the houses.

B. Data Pre-processing

In our work, we down-sample the data to 1 minute sampling rate. This achieved two purposes:

- The long intervals for which the data was zero or not available was significantly reduced.
- This also allows us to use lesser number of samples, so as to reduce the complexity burden on the algorithm.

We have considered close to two months of data for validating the proposed architecture.

Windowing: Raw aggregated data was fed as input to the encoder. The discriminant layer will have the information about the appliances which are operating during that particular window. The appliance information is given in the form of binary values i.e L_i is 1 if the i^{th} appliance (L_i) is present in that particular window or otherwise L_i is 0. Appliances' consumption data will have non-zero values whenever they are ON and zero or very small positive values whenever the appliance is OFF. To binarise these appliances, we used a small positive threshold ϵ to obtain just the ON or OFF states, thereby modeling them as binary state machines. This is done as we are using the formulation to identify the operational appliances in the aggregate data.

$$\begin{aligned} L_i \quad &= 1, \quad P_i > \epsilon \\ &= 0 \quad P_i < \epsilon \end{aligned} \tag{6}$$

where P_i is the power consumption of the i^{th} appliance.

IV. Testing

After training the architecture with 40 days of data, we tested the same with 25 days of data. The product of learnt linear mappings and the learnt encoded representations is used to estimate the labels, as given by the following equation.

$$l = DW_1X_{test} \tag{7}$$

The estimated labels will have a certain bias as we observe in the Figure 2 to Figure 4. Therefore, the estimated results are thresholded appropriately, to detect the presence of the appliance. If the estimated values fall below the thresholded value, we consider them to be absent at that time instant; else, they would be considered as operational. The thresholds are usually unique for each appliance, and therefore, careful understanding of the appliance ranges, during the training phase is necessary before deciding the same.

A. Evaluation procedure

Window sizes of one hour, three hours and one day were used to assess the performance of the algorithm. We found that windowing size did not significantly affect the reconstruction, however, in order to obtain full cycles of appliance operation in one window, we preferred three hours window size as the optimal window size. We observe that pre-processing of the data to remove all the windows where no loads were operating , both in the labeled data and in the aggregate data, was crucial to improving the reconstruction accuracy. The results provided in Figures (2-4) are obtained by concatenating windows to demonstrate the reconstruction for one day of power consumption. We observe that Washer-Dryer appliance has the best reconstruction; this can be attributed to the high-power consumption of the appliance. Figures (2 - 4) contain the reconstruction instances of the Washer-Dryer against the ground truth. Refrigerator and dishwasher reconstructions had false negatives (refer Figures 2,4), i.e, the appliances were not detected in some windows. This could be attributed to the low-power consuming nature of these appliances, compared to the rest of the loads. Oven reconstruction was the least accurate in comparison - this is chiefly owing to the fact that oven's operation time is less than a minute. While sufficient labeled data does exist, latching onto the appliance especially in the presence of other appliances operating for a longer time seems to be a problem. Figure 2 contains the reconstruction error for oven in the presence of dishwasher. There were no false positives reported in any of the appliances' reconstruction.

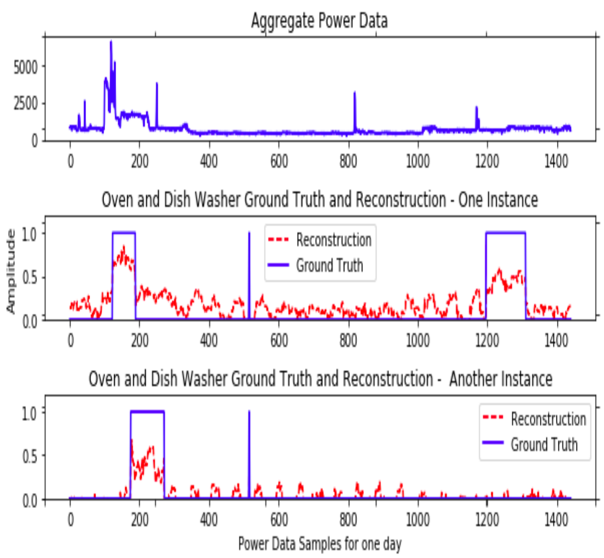

Fig. 2. Oven and DishWasher reconstruction

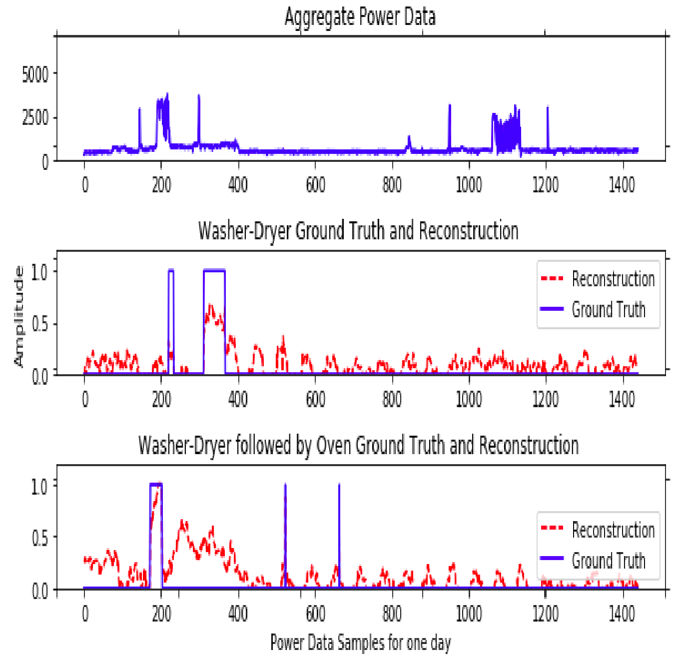

Fig. 3. Oven and Washer-Dryer reconstruction

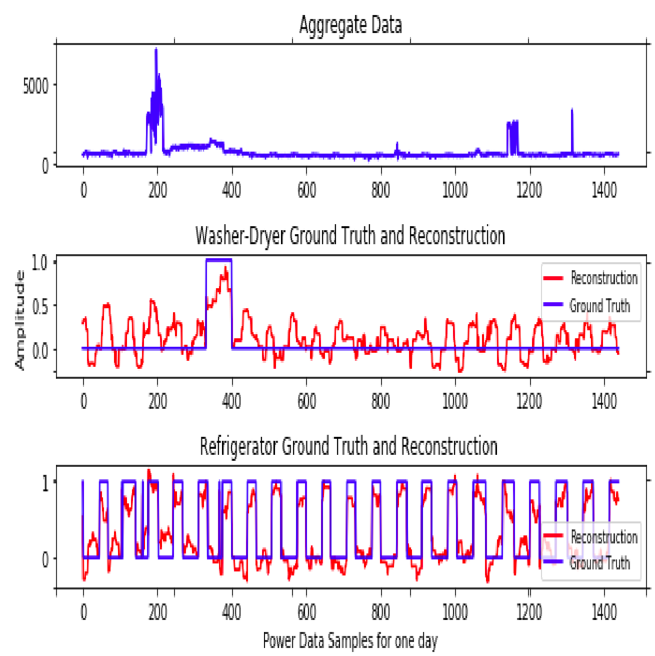

Fig. 4. Refrigerator and Washer-Dryer reconstruction

V. Comparison and Discussions

Comparision with MLKNN and RAKEL algorithms: We compare the results of the DiAE framework with those of popular multi-label classification algorithms - MLKNN [19] and RAKEL [20]. These algorithms were implemented and tested with REDD data sets, for the same period of training and testing as for DiAE framework.

We observe that while there is not much difference in the results using DiAE and the rest of the two algorithms for refrigerator instances - all the three are observed to accurately predict the appliance; however, the same does not hold for high power appliances such as Microwave oven and Washer-Dryer. It is observed that DiAE outperforms both MLKNN and RAKEL when tested for high power appliances. We present the results using MLKNN and RAKEL algorithms for two-hour window durations, to highlight the large number of false positives in the reconstructions.

Improvement against proposed framework by Kelly et al.: The results from our work, were also compared against those demonstrated by Kelly et al. [11]. This is shown in table V. Kelly et al. demonstrates the performance of three different neural network models for energy disaggregation. In this work, De-noising Auto-Encoders (AE) were trained, one for each appliance, with the assumption that all the remaining appliances contributing to the aggregate power is noise. The output of each denoising AE is the identified appliance. However, by using the DiAE formulation, we have simplified the discriminative process greatly as opposed to the training process involved in the former [11], without compromising on the disaggregation performance. We observe that our results using this approach score better in terms of accuracy and F1-score, with the only exception of Oven reconstruction.

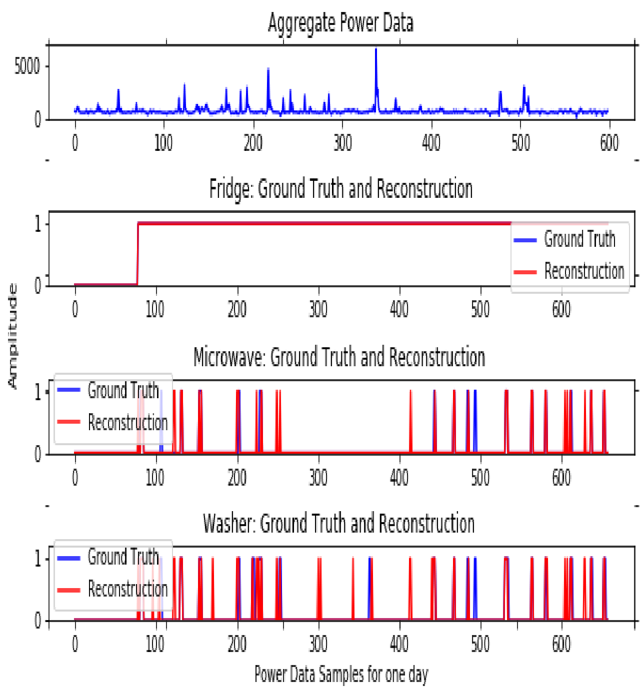

Fig. 5. Refrigerator, Microwave and Washer-Dryer reconstructions using MLKNN algorithm

LD Results-DiAE framework Vs Denoising AE				
	Denoising AE		DiAE Approach	
Appliance Name	Accuracy	F1-score	Accuracy	F1-score
Dish Washer	0.95	0.6	0.975	0.95
Refrigerator	0.85	0.81	0.9	0.85
Oven	0.85	0.6	0.75	0.62
Dryer-Washer	0.75	0.49	0.9	0.9

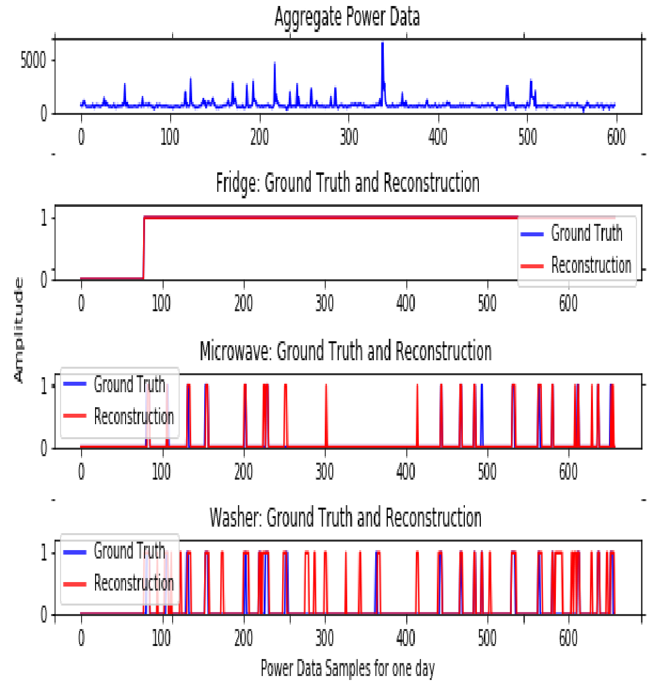

Fig. 6. Refrigerator, Microwave and Washer-Dryer reconstructions using RAKEL algorithm

LD Results due to MLKNN and RAKEL				
	MLKNN		RAKEL	
Appliance Name	Accuracy	F1-score	Accuracy	F1-score
Dish Washer	0.8	0.75	0.85	0.77
Refrigerator	0.95	0.9	0.95	0.9
Oven	0.65	0.6	0.65	0.6
Dryer-Washer	0.75	0.7	0.7	0.65

As we obtain appliance identification as the output in this formulation, it is necessary to clamp the reconstruction to a '1' or '0', depending on a threshold value. A threshold value of 0.5 is suitable with the data used in this work. Later a simple post-processing scheme, involving multiplication of the identified appliances by the aggregate power consumption data, will help to obtain the appliance shapes too. Along with the timing information that is already in place, the reconstructed shape thus obtained through post-processing would provide for complete picture of the disaggregation.

VI. Conclusions

We present an approach to load disaggregation by using Discriminative Auto-Encoder framework. The formulation is based on simple sequential updations using least square minimization, to establish discriminative ability accurately at the decoding layer. A hallmark of this work is the ability to use

DiAE framework for superposed loads in the labeled data , and this work has been the first attempt to report results with such a modification to the class label matrix. We have been able to use just 40 days of training data to achieve the accuracy, which demonstrates the use of this algorithm as intended for a real life solution.

We also compared the proposed methodology against MLKNN and RAKEL algorithms, and observed that the proposed method outperforms both these algorithms, when tested for high-power consuming appliances such as washer and Microwave oven, which is usually required for real life situations.

Extending this work further, the authors are implementing deeper version of the proposed architecture and the performance of the same for Load Disaggregation problem, is currently being studied.

REFERENCES

[1] G. Hart, *Nonintrusive Appliance Load Monitoring*. Proceedings of the IEEE, vol. 80, no. 12, 1992.

[2] Makonin, Stephen, et al.*AMPds: A public dataset for load disaggregation and eco-feedback research.* 2013 IEEE Electrical Power and Energy Conference.

[3] Barsim, Karim Said, and Bin Yang. *On the feasibility of generic deep disaggregation for single-load extraction.* arXiv preprint arXiv:1802.02139 (2018)

[4] Faustine A, Mvungi N.H, Kaijage S, & Michael K. *A survey on non-intrusive load monitoring methodies and techniques for energy disaggregation problem*arXiv preprint arXiv:1703.00785.

[5] Gogna, Anupriya, and Angshul Majumdar. *Discriminative Autoencoder for Feature Extraction: Application to Character Recognition.* Neural Processing Letters 49.3 (2019): 1723-1735.

[6] Rahul Sinha, S Spoorthy, Prerna Khurana, M Girish Chandra.*Power system load data models and disaggregation based on sparse approximations.* IEEE 14th International Conference on Industrial Informatics (INDIN),2016: 292-299.

[7] Kolter, J. Zico, and Matthew J. Johnson. *REDD: A public data set for energy disaggregation research.* Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA. Vol. 25. No. Citeseer. 2011.

[8] J. Zico Kolter, Siddarth Batra, and Andrew Y. Ng. 2010. *Energy disaggregation via discriminative sparse coding.* In Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1 (NIPS'10), J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta (Eds.), Vol. 1. Curran Associates Inc., USA, 1153-1161.

[9] Parson, Oliver, et al. *Non-intrusive load monitoring using prior models of general appliance types.* Twenty-Sixth AAAI Conference on Artificial Intelligence. 2012.

[10] N. K. Thokala, M. G. Chandra and K. Nagasubramanian. *On load disaggregation using discrete events.* 2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia), Melbourne, VIC, 2016, pp. 324-329.

[11] Jack Kelly and William J. Knottenbelt. *Neural nilm: Deep neural networks applied to energy disaggregation.* Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments. ACM, 2015.

[12] J. Alcal, J. Urea.A, Hernndez and D. Gualda. *Event-Based Energy Disaggregation Algorithm for Activity Monitoring From a Single-Point Sensor.* in IEEE Transactions on Instrumentation and Measurement, vol. 66, no. 10, pp. 2615-2626, Oct. 2017.

[13] Armel.K.C, A.Gupta, Shrimali.G, Albert, A. *Is disaggregation the holy grail of energy efficiency? The case of electricity.* Energy Policy, Vol.52, January 2013, United States, issn:0301-4215.

[14] Nipun Batra and Oliver Parson and Mario Berges and Amarjeet Singh and Alex Rogers. *A comparison of non-intrusive load monitoring methods for commercial and residential buildings.* arXiv preprint arXiv:1408.6595.

[15] Nipun Batra, Jack Kelly, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, Mani Srivastava. *NILMTK: an open source toolkit for non-intrusive load monitoring.* Proceedings of the 5th international conference on Future energy systems, 2014/6/11, 265-276.

[16] C. Laughman, S. Leeb, and Lee. *Advanced non-intrusive monitoring of electric loads.* IEEE Power and Energy, 2003.

[17] Candes, Emmanuel J, Michael B. Wakin, and Stephen P. Boyd. *Enhancing sparsity by reweighted l1 minimization.* Journal of Fourier analysis and applications 14.5-6 (2008).

[18] Taylor, G., Burmeister, R., Xu, Z., Singh, B., Patel, A. and Goldstein, T., 2016, June. Training neural networks without gradients: A scalable admm approach. In International conference on machine learning (pp. 2722-2731).

[19] K. Basu, V. Debusschere, S. Bacha, U. Maulik and S. Bondyopadhyay, "Nonintrusive Load Monitoring: A Temporal Multilabel Classification Approach," IEEE Transactions on Industrial Informatics, Vol. 11 (1), pp. 262-270, 2015.

[20] S. M. Tabatabaei, S. Dick and W. Xu, "Toward Non-Intrusive Load Monitoring via Multi-Label Classification," IEEE Transactions on Smart Grid, Vol. 8 (1), pp. 26-40, 2017.