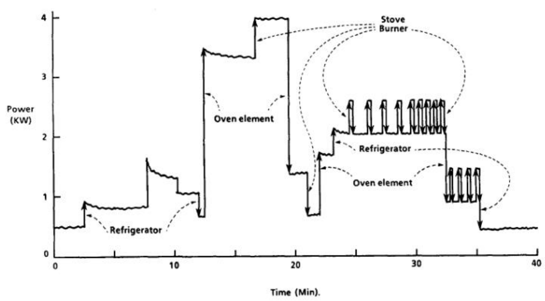

REAL-TIME ENERGY DISAGGREGATION

Author:
1706058- Abdullah Jubair Bin Iqbal
1706059- Tasnim Farhan Fatin
1706060- Md Al-Amin
1706061- Swadesh Vakta

EEE 312

Table of Contents

Forwarding Letter

Abstract

In this project, we've tried applying various DNNs to the problem of non-intrusive load monitoring (NILM) and compared their results for various appliances using the REDD dataset. We took a sliding window approach in hopes that we'll be able to achieve real time disaggregation with further tuning and testing. We compare the disaggregated energy consumption results based on MSE, MAE, Relative Error and F1 Score.

Keywords

Energy disaggregation; neural networks; feature learning; NILM; energy conservation; deep learning.

1 Introduction

The Non-Intrusive Load Monitoring, also referred to as Energy disaggregation, was invented by George W. Hart, Ed Kern, and Fred Schweppe of MIT in the early 1980s.

The household's energy consumption signals are analyzed and decomposed in various other signals corresponding to individual appliance energy consumption. If we see the following figure, we'll get a clear vision figure 1.

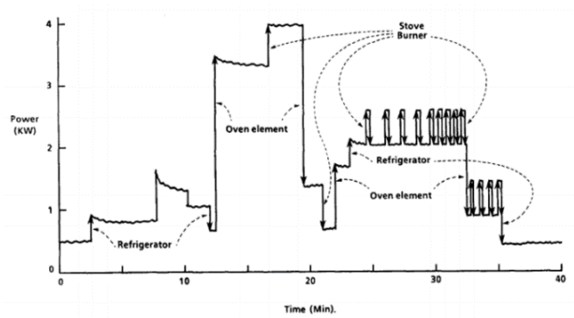

Figure 1: An example of NILM principle *[1]*

Figure 1 shows a power consumption versus time for a single-family household over 40 min. It contains four different-sized step changes, providing characteristic signatures for four appliances. The Energy disaggregation problem can be formulated with the next equation at any point in time t: [2]

$$X_t = \sum_{i=1}^{N} y_t^i + \sigma(t)$$

With σ(t) any contribution of the devices not taken into account and measurement noise. A given sequence of main power consumption $X = \{X_1, X_2, \dots, X_T\}$ corresponding to N active appliances at $t = \{1, 2, \dots, T\}$. Thus, the objective is to infer the power contribution y_t^i from the device $i \in \{1, 2, \dots, T\}$ at time t.

The key problem is how to design an efficient and generalized energy disaggregation algorithm easily applied across several buildings that can run as close to real-time as possible. [2]

NILM is taking more and more space in current research due to the development of smart grids and the massive deployment of smart meters. This is also due to the unquestionable advantages of NILM which allows us to take up the expected challenges. Among them can be distinguished [3] [4]

- Detailed consumption information: the main advantage for the customer, the provision of more information to the user, with a higher frequency to obtain energy savings and then bill's reductions. In addition, the current time or real-time information about appliances switched on could become the norm. It could even provide reminders to switch off some appliances before leaving home.
- Appliance-by-appliance energy consumption: This allows the consumer to find out which appliances consume the most energy in their home, and more generally the share of each appliance in their total energy consumption. This is one of the most efficient ways for the consumer to reduce its energy bill. In fact, in [19], they estimate savings from 9% to 20% by putting in place a consumption strategy based on this itemized energy consumption data.
- Detecting malfunctioning devices: an accurate record of device use is useful for checking the status of the devices and the detection of faulty devices (MFD).
- Demand response application: the detection of deferrable loads or inactivity periods in the energy consumption of consumers can be used to offer them a demand response program on the devices concerned.
- Ambient intelligence: it enables other sensing approaches without the need to implement new sensors.
- Occupancy detection: it would be possible to deduce the presence or absence in the household by the power consumption. However, it may involve an intrusion into the privacy of thousands of users of the electric network. On the other hand, it could provide security detecting illegal occupations when customers are away from home on weekends or holidays.
- Illegal load detections: the detection of anomalous loads in households are made more precise and can be used to report possible energy thefts in public and private buildings.

2 Non-Intrusive Load Monitoring (NILM)

Unlike Intrusive Load Monitoring, the NILM does not require any prior installation on appliances (like sub meters). Only a smart meter for the whole house is necessary that implies low cost and simplicity of deployment, that is why this method is favored, despite its lesser effectiveness. Therefore, the NILM relies mostly on the capacities of the software, which involves important previous steps.

2.1 Experimental Framework

All the parameters involved in the project will be discussed here:

2.1.1 Datasets

Two datasets are used, one from Europe and one from the USA, to make a comparison on two different domains. These datasets record the aggregated demand of the entire household as well as the ground truth demand of individual appliances.

REDD

This REDD dataset is one of the first used in a large scale by the scientific group. It contains the main power and sub meter power data of 6 houses with a 1s sample rate, and even high frequency for 2 houses.

Table 1: Dataset

Dataset	**REDD**
Countries	United States
Duration	3-19 days
Households	6
Frequency	3 seconds (1 seconds)
Terms of Use	Registration
Citations	759

2.2 Denoising Autoencoder (DAE)

Denoising Autoencoder take partially corrupted input and is trained to recover the original undistorted input. In practice, the objective of denoising autoencoders is that of cleaning the corrupted input, or *denoising.* A DAE architecture is represented in figure 2:

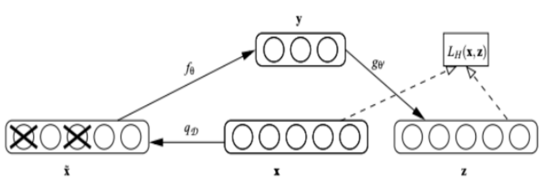

Figure 2: General Denoising Autoencoder architecture [11]

This example of training process works as follow:

- The initial input x is corrupted into $\tilde{x}$ through stochastic mapping $\tilde{x} \sim q_D(\tilde{x}|\ x)$
- The corrupted input $\tilde{x}$ is then mapped to a hidden representation (y) with the encoder $h = f_\theta(\tilde{x}) = s(W\tilde{x} + b)$.
- From the hidden representation the model reconstructs, $z = g_{\theta'}(h)$.
- The reconstruction error is measured by loss $L_H(x, y)$.

In our project, we used the architecture created by J. Kelly. [5] Layers of the architecture is described below:

1. Input (length determined by appliance duration)
2. 1D conv (filter size=4, stride=1, number of filters=8, activation function=linear, border mode=valid)
3. Fully connected (N=(sequence length) ×8, activation function=ReLU)
4. Fully connected (N=128; activation function=ReLU)
5. Fully connected (N= (sequence length) ×8, activation function=ReLU)
6. 1D conv (filter size=4, stride=1, number of filters=1, activation function=linear, border mode=valid)

2.3 Recurrent Neural Network with Long Short-Term Memory (LSTM)

The RNN is a Deep Neural Network that processes sequential data. It is a generalization of feedforward neural network with an internal memory. RNN uses its internal memory to process variable-length sequences of input. As a result, unlike other neural networks, all the inputs are related to each other as shown in figure 3.

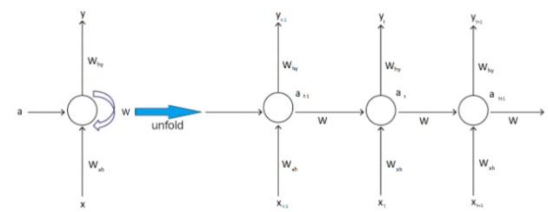

Figure 3: Architecture of RNN

In figure 3, W_{xh} corresponds to the connection between input and hidden layers, W is the weights from a hidden layer to another. This particular algorithm allows a prediction based on the neuron's previous prediction at t-1. Thus, it generates a map from the entire history of the inputs to an output vector [38].

Back Propagation Through Time (BPTT) process to back propagate the error, at each time step, from the last time step to the first-time step updating the weights along the way [62]. The next schema represents well this idea:

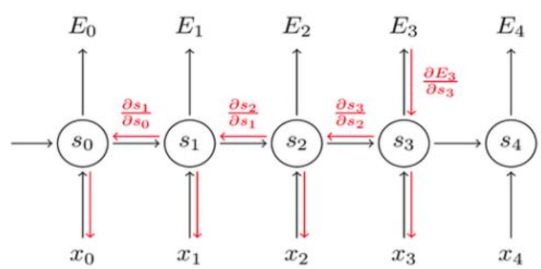

Figure 4: Back Propagation Through Time schema

One of the problems of backpropagation is the possible loss of the gradient information on the way, sometimes called the "vanished gradient" problem, which leads to a limitation of the "memory capacity" of the DNN.

Long Short-Term Memory (LSTM) networks can solve this vanishing gradient problem of RNN. It trains the model by using back-propagation. In an LSTM network, three gates are present. **Input gate** discovers which value from input should be used to modify the memory. **Forget gate** discovers what details to be discarded from the block. **Output gate** — the input and the memory of the block is used to decide the output. This multiplicative input and output gate protect the memory content from perturbation. An example of the operation of an LSTM RNN is represented below.

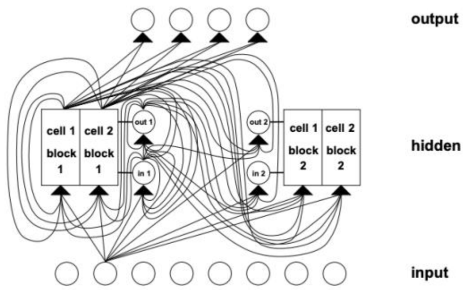

Figure 5: Example of architecture of RNN with LSTM [12]

There is a net with 8 input units, 4 output units, and 2 memory cell blocks of size 2. in1 and out1 are respectively the input and output gates, cell1/block1 tags the first memory cell of block 1.

The precise architecture applied is the following :

1. Input (length determined by appliance duration)
2. 1D conv (filter size=4, stride=1, number of filters=16, activation function=linear, border mode=same)
3. Bidirectional LSTM (N=128, with peepholes)
4. Bidirectional LSTM (N=256, with peepholes)
5. Fully connected (N=128, activation function=TanH)

2.4 Sequence-to-Point (Seq2point)

Seq2Point predicts the midpoint of an appliance given a window of the mains (the input). The mains readings are defined as $Y[t:t+W-1]$ and the output is defined by the midpoint element x_τ of the appliance's power window $X[t:t+W-1]$, with $\tau = t + [W/2]$.

The model used is $x_\tau = F_p(Y[t:t+W-1]) + \epsilon$ and the Loss function:

$$L_p = \sum_{t=1}^{T-W+1} \log p\big(x_{tau} | Y[t:t+W-1], \theta_p\big)$$

Its main advantage is to make a single prediction for every midpoint element, instead of the average of predictions for each window.

The architecture involved here is detailed in the figure below:

1. Input sequence with length Y[t:t+W−1]
2. 1D Convolution: filters: 30; filter size: 10, activation function=ReLU
3. 1D Convolution: filters: 30; filter size: 8, activation function=ReLU
4. 1D Convolution: filters: 40; filter size: 6, activation function=ReLU
5. 1D Convolution: filters: 50; filter size: 5, activation function=ReLU
6. 1D Convolution: filters: 50; filter size: 5, activation function=ReLU
7. Fully connected: units: 1024, activation function=ReLU
8. Output: Output: Number of units:1, activation: linear

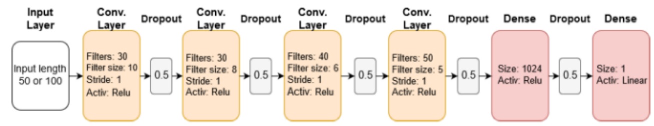

Figure 6: Precise Architecture of RNN with LSTM involved in this paper

2.5 Window GRU - DNN

This algorithm was inspired by the limitations of the RNN with LSTM described above, with the willingness to obtain the same quality of result with an optimized structure, requiring less computational demand. They chose to replace the LSTM with simpler architecture elements with no internal memory: light-weight Gated Recurrent Units (GRU).

The second optimization is the considerable reduction of nodes per layer, about half size, as empirically no significant changes in the accuracy has been noted. They reached a decrease of the trainable parameters by 60%.

The last enhancement is the additional dropout between the different layers to prevent overfitting and manage more easily missing values. The precise architecture is represented below

1. Input (length determined by appliance duration)
2. 1D conv (filter size=4, stride=1, number of filters=16, activation function=ReLU, border mode=same)
3. Bidirectional GRU (N=64, activation function=ReLU)
4. Bidirectional GRU (N=128, activation function=ReLU)
5. Fully connected (N=128, activation function=ReLU)
6. Output: Output: Number of units:1, activation: linear

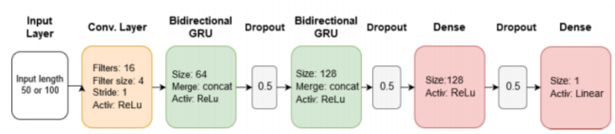

Figure 7: Architecture of the GRU network involved in this paper

The DNN observes the precedent mains power readings $Y_{t:t+W-1}$ and calculate, for the last time point, the power consumption $x_{j(t+W-1)}$ of a single device.

3 EXPERIMENT DATA

3.1 Data Preprocessing

To produce accurate prediction, raw data is normalized which scales data in range of [0 1], according to the equation:

$$X_{new} = \frac{X}{X_{max}}, Y_{new} = \frac{Y}{Y_{max}}$$

Where X is the aggregated data and Y is the ground truth.

3.2 Framing by Window

Sliding window method segments data according to the window length. Window size is gradually increased to decrease error. If we have N number of observations and a window length of W, then we will have N-W+1 windowed observations to evaluate.

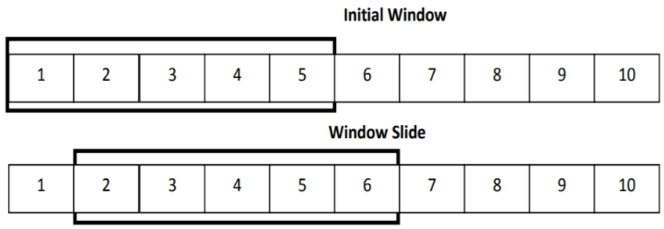

Figure 8: Sliding Window of length 5

Figure 2 shows process of sliding window with window size=5.

3.3 Training and Testing Data

We trained all of the models using the REDD dataset with a sampling period of 3 seconds. The buildings were split for training and evaluation as mentioned in Table 1.

Table 2: Dataset splitting for Training and Testing

Appliance	Training	Testing
Fridge	1,3,5,6	2
Microwave	2,3,5	1
Light	1,3,4,5	2

At first, window lengths were kept default to 50 samples (2.5 minutes). But gradually different models showed different output with respect to window length. The final window sizes are mentioned in Table 3.

Table 3: Window length for different models

Model	Refrigerator	Microwave	Lighting
LSTM	50	100	100
DAE	50	200	50
GRU	50	100	100
S2P	50	50	50

4 Metrics

The metrics used here to measure the performance of the different algorithms.

4.1 Regression Metrics

They measure how well the NILM algorithm can estimate and assign the power consumed by each appliance. This is the ability of the algorithm to infer the correct power consumption value.

4.1.1 MAE (Mean Average Error)

MAE is one of the most used metrics, it is, therefore, indispensable if one intends to compare its results with another research.

$$MAE = \frac{1}{T}\sum_{t=1}^{T} |\hat{x}_t - x_t|$$

It measures the absolute difference between output prediction of the disaggregation algorithm and the ground truth, then makes an average.

4.1.2 Relative Error

This metric is the quotient of the absolute difference between the real power $y_t^{(i)}$ and the estimated output power $\hat{y}_t^{(i)}$ by the sum of 1 and the predicted power. The relative error is defined as:

$$Relative\ Error = \frac{|y_t^{(i)} - \hat{y}_t^{(i)}|}{\max(\hat{y}_t^{(i)}, y_t^{(i)})}$$

4.2 Classification Metrics

Classification metrics measure how accurately NILM algorithms can predict what appliance is running in each state. This is the ability of the algorithm to detect ON/OFF events. In our project, the event detection is based on the outputs from energy disaggregation.

4.2.1 Precision (Energy-based)

The Precision (energy-based) has the role to determine the amount of power assigned to a device that belongs to it. For i^{th} appliance it takes the form:

$$P_i = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

4.2.2 Recall (Energy-based)

Concerning the Recall (energy-based), it measures the correctly classified part of the power consumption.

For i^{th} appliance:

$$R_i = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

4.2.3 F1 Score (Energy-based)

The F1 score is the harmonic mean between Precision and Recall:

$$F = 2 \times \frac{P \times R}{P + R}$$

4.2.4 Accuracy

Accuracy is one of the most used classification metrics. It is the measurement of the rate of successful classification.

$$Accuracy = \frac{Correct\ Matches}{Total\ Possible\ Matches}$$

$Correct\ Matches$ refers to a correct prediction by the aggregation algorithm and the ground truth

5 Results and Discussion

The disaggregation results are shown in the following tables.

We used DAE and LSTM model according to the paper from Jack Kelly. However, training LSTM neural network is memory as well as time consuming. So we used GRU neural network, which has almost same architecture like LSTM but can run faster. As a result, very much efficient for real-time neural network. We also used Seq2Point neural network for real-time disaggregation, which is very much used for timeseries prediction.

GRU model outperforms LSTM in refrigerator data for window length of 50. But for lighting and microwave data, GRU and LSTM behaves similarly.

Seq2Point model outperforms all other models. Another interesting fact is. Accuracy of the models can be improved if window size is increased. However, due to computation cost, we had to limit our window size.

Also, we have to keep in mind that window length results in delay in output. A typical window length of 50 results in delay of at least 2.5s. So, it is a tradeoff.

And the fourth model, DAE, as described by Jack Kelly, behaves average w.r.t. other three models.

For Refrigerator energy disaggregation, if we want to limit window length, it is best to use house 1,3,5,6 as training data. However, other houses can be effective if window length is increased. It should be noted that, DAE behaves very poor for refrigerator energy disaggregation also if window length is increased.

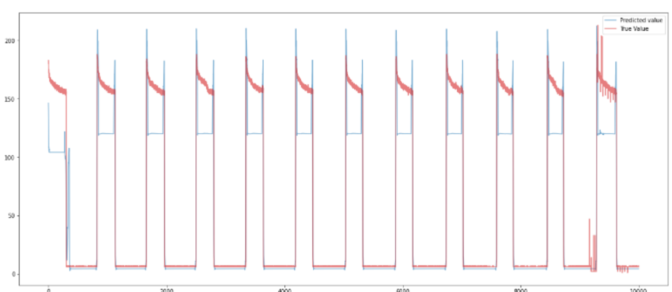

Figure 9: Fridge S2P

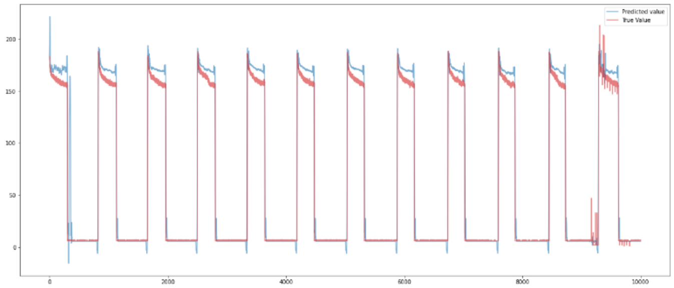

Figure 10: Fridge LSTM

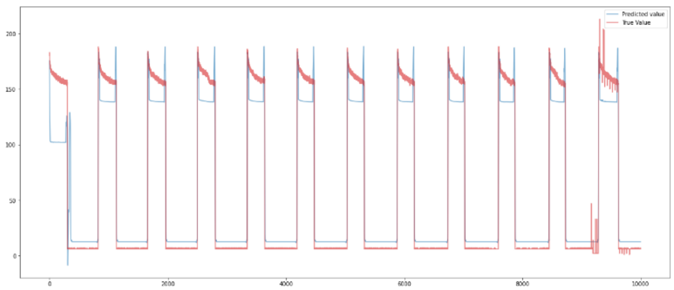

Figure 11: Fridge GRU

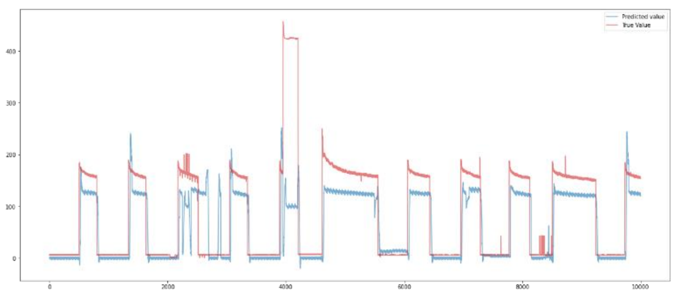

Figure 12: Fridge DAE

For Lighting energy disaggregation, almost all the neural networks works fine. It is suggested to train data using 1,3,4,5.

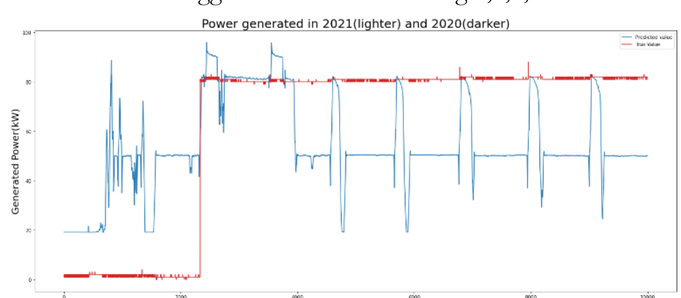

Figure 13: Light S2P

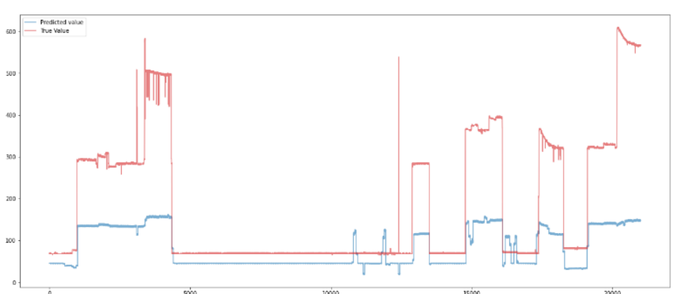

Figure 14: Light LSTM

Finally, for microwave energy disaggregation, training with GRU using data of house 2,3,5 shows better output. Rest of the neural networks fail to describe its multi-state property.

Table 4 Refrigerator Sequence to Point Model

Test house	Training House	MSE	Relative Error	Mean Absolute error	Recall	Precision	Accuracy	F1
Window 50								
1	2,3,5,6	6491.55	0.23	63.73	99.63	38.23	59.47	55.26
2	1,3,5,6	2802.12	0.19	30.24	99.47	89.14	94.39	94.03
3	1,2,5,6	6521.92	0.05	56.4	40.22	53.82	64.67	46.04
5	1,2,3,6	5155.1	0.16	50.17	99.65	40.2	40.2	57.35
Window 100								
1	2,3,5,6	5776.16	0.14	58.93	79.66	36.16	59.56	49.75
2	1,3,5,6	2416.08	0.109	24.176	97.5	90.36	94.27	93.79
5	1,2,3,6	8188.05	0.4	57.87	59.63	64.03	70.29	61.75
Window 200								
1	2,3,5,6	2865.9	0.23	30.94	89.41	76.61	90.48	82.52
2	1,3,5,6	2598.73	0.101	24.08	99.19	71.47	81.91	83.08
5	1,2,3,6	5367.96	0.36	46.1	74.41	74.99	79.7	74.7
Window 400								
1	2,3,5,6	2682.76	0.02	37.31	98.44	78.43	92.8	87.3
5	1,2,3,6	4864.87	0.36	45.55	96.05	74.28	84.98	83.77

Table 5: Refrigerator LSTM Model

Test house	Training House	MSE	Relative Error	Mean Absolute error	Recall	Precision	Accuracy	F1
Window 50								
1	2,3,5,6	5111.72	0.12	47.58	81.16	49.8	74.71	61.73
2	1,3,5,6	10366.53	0.18	65.48	40.38	52.78	64.14	45.76
5	1,2,3,6	8328.94	0.46	57.58	72.11	72.9	78.01	72.5

Table 6: Refrigerator GRU Model

Test house	Training House	MSE	Relative Error %	Mean Absolute error	Recall	Precision	Accuracy	F1
Window 50								
1	2,3,5,6	6810.98	19	63.62	80.87	34.92	57.33	48.77
2	1,3,5,6	2344.6	14.4	25.485	98.36	91.41	95.17	94.76
3	1,2,5,6	6585.06	6	59.33	40.07	53.32	64.41	45.76
5	1,2,3,6	8199.55	46	58.54	51.57	62	67.83	56.31

6	1,2,3,5	4433.66	21.1	52.28	100	50.52	50.52	67.13
Window 100								
1	2,3,5,6	5949.63	12	55.7	81.12	38.54	62.76	52.25
2	1,3,5,6	4574.06	29	37.13	79.81	89.02	86.66	84.16
5	1,2,3,6	7582.84		55.04	62.72	69.49	73.93	65.93

Table 7: Refrigerator DAE

Test house	Training House	MSE	Relative Error	Mean Absolute error	Recall	Precision	Accuracy	F1
Window 50								
1	2,3,5,6	5115.34	0.14	55.12	99.84	36.55	55.9	53.52
2	1,3,5,6	2830.32	0.24	30.96	99.29	82.16	90.03	89.92
5	1,2,3,6	3401.2	0.06	40.35	99.84	27.55	33.2	43.19
Window 100								
1	2,3,5,6	5665.86	0.18	57.93	99.57	35.83	54.56	52.7
2	1,3,5,6	2811.26	0.2179	31.04	98.9	81.17	89.23	89.16
5	1,2,3,6	5876.8	0.29	49.34	98.89	68.22	81.02	80.74
Window 200								
1	2,3,5,6	3199.51	0.12	33.85	99.46	36.89	56.6	53.82
2	1,3,5,6	3098.71	0.1365	30.63	98.405	78.24	87.03	87.17
5	1,2,3,6	4164.99	0.17	40.63	98.44	66.33	79.21	79.26
Window 400								
1	2,3,5,6	3849.77	0.08	37.45	97.04	37.61	58.31	54.21
2	1,3,5,6	3536.38	0.19	34.33	98.67	73.1	83.15	83.99
5	1,2,3,6	4240.67	0.22	40.83	96.02	62.71	75.35	75.87

Table 8: Light S2P

Test house	Training House	MSE	Relative Error %	Mean Absolute error	Recall	Precision	Accuracy	F1
Window 50								
5	1,2,3,4	12999.12	0.52	68.03	26.6	100	26.6	42.02
2	1,3,4,5	2076.26	0.39	34.64	77.7	65.89	92.07	71.31
Window 100								
5	1,2,3,4	14227.16	0.64	80.08	25.43	100	25.43	40.55
2	1,3,4,5	2030.17	0.38	36.15	60.44	58.08	89.44	59.23

Table 9: Lighting GRU

Test house	Training House	MSE	Relative Error %	Mean Absolute error	Recall	Precision	Accuracy	F1

Window 50								
2	1,3,4,5	2243.72	0.37	33.85	53.19	58.41	89.25	55.68
Window 100								
2	1,3,4,5	2052.26	0.33	32.9	54.47	59.42	89.5	56.84

Table 10: Lighting DAE

Test house	Training House	MSE	Relative Error %	Mean Absolute error	Recall	Precision	Accuracy	F1
Window 50								
5	1,2,3,4	12854.74	0.52	64.81	99.9	100	99.9	99.95
Window 100								
5	1,2,3,4	14259.52	0.53	62.35	99.97	100	99.97	99.98
Window 200								
5	1,2,3,4	16075.62	0.54	66.59	100	100	100	100

Table 11: Lighting LSTM

Test house	Training House	MSE	Relative Error %	Mean Absolute error	Recall	Precision	Accuracy	F1
Window 50								
2	1,3,4,5	1781.19	0.016	21.643	80.83	70.52	93.28	75.32
Window 100								

Table 12: Microwave S2P

Test house	Training House	MSE	Relative Error %	Mean Absolute error	Recall	Precision	Accuracy	F1
Window 50								
5	1,2,3	4015.8403	0.119	13.98	0.98	3.45	92.85	1.53
1	2,3,5	24326	0.359	23.03	50.29	28.69	97.68	36.53
Window 100								
5	1,2,3	1625.206	0.102	11.12	1.89	5.82	92.71	2.86
1	2,3,5	20858	0.45	20.57	82.73	29.4	97.13	43.41
Window 200								
5	1,2,3	6570.79	0.61	27.79	2.21	1.69	87.19	1.92

Table 13: Microwave GRU

Test house	Training House	MSE	Relative Error %	Mean Absolute error	Recall	Precision	Accuracy	F1
Window 50								

5	1,2,3	6906.781	0.3735	17.93	2.1	4.11	91.69	2.78
1	2,3,5	22150.93	0.209	21.88	70.32	43.14	98.37	53.47
Window 100								
5	1,2,3	1963.59	0.0133	12.47	2.44	3.83	91.01	2.98
1	2,3,5	25039	0.56	20.63	24	45.14	98.6	31.3
Window 200								
5	1,2,3	2429.75	0.2	13.17	2.67	6.12	92.16	3.72

Table 14: Microwave DAE

Test house	Training House	MSE	Relative Error %	Mean Absolute error	Recall	Precision	Accuracy	F1
Window 50								
5	1,2,3	2814.03	0.438	19.272	10.77	6.59	85.9	8.18
1	2,3,5	25342	0.511	22.339	84.199	29.76	96.56	43.98
Window 100								
5	1,2,3	2236.59	0.243	14.465	12.42	7.04	85.53	8.98
1	2,3,5	23628	0.44	21.73	79.08	19.57	94.49	31.3
Window 200								
5	1,2,3	942.74	0.247	9.65	9.47	8.25	88.57	8.82
1	2,3,5	25110	0.619	20.17	48.02	23.06	96.59	31.15
Window 400								
5	1,2,3	1807.4752	0.1308	13.205	19.18	8.26	82.74	11.54

Table 15: Microwave LSTM

Test house	Training House	MSE	Relative Error %	Mean Absolute error	Recall	Precision	Accuracy	F1
Window 50								
5	1,2,3	6142.835	0.47	18.47	0.7	5.13	91.63	3.57
Window 100								
5	1,2,3	6487	0.5338	27.33	3.58	2.55	86.8	2.984
1	2,3,5	26112.678	0.1814	28.24	60.23	26.48	97.25	36.79

6 Conclusions

This project allowed us to confirm the results obtained for example in paper by Jack Kelly. From the MAE and accuracy, it is better to use sequence to point neural network then other models. However, it needs larger window size to have higher classification metrics score. In that case GRU can be helpful.

References

[1] G. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE,* vol. 80, no. 12, pp. 1870-1891, 1992.

[2] A. Faustine, N. H. Mvungi, S. Kaijage and K. Michael, "A Survey on Non-Intrusive Load Monitoring Methodies and Techniques for Energy Disaggregation Problem," *arXiv:1703.00785,* 2017.

[3] R. H. Jorge, Á. L. Murciego, A. L. Barriuso, D. H. d. l. Iglesia, G. V. González, J. M. C. Rodríguez and R. Carreira, "Non Intrusive Load Monitoring (NILM): A State of the Art," in *Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection*, 2017.

[4] B. Najafi, S. Moaveninejad and F. Rinaldi, "Data Analytics for Energy Disaggregation: Methods and Applications," *Big Data Application in Power Systems,* pp. 377-408,, 2018,.

[5] J. Kelly and W. Knottenbelt, "Neural NILM: Deep Neural Networks Applied to Energy Disaggregation," in *ACM BuildSys'15*.

[7] K. Ehrhardt-Martinez, K. A. Donnelly and J. A. Laitner, "Advanced metering initiatives and residential feedback programs: A meta-review for household electricity saving opportunities," 2010.