Mini Project-II Report

On

NON-INTRUSIVE LOAD MONITORING(NILM)

Submitted By

Erigi Vaishnavi - 20BCS044
Golla Anjaiah - 20BCS048
Rishabh Gautam - 20BCS112
P. Vishnu Kumar - 20BCS138

Under the guidance of

Dr. Prakash Pawar

Assistant Professor, Electronics & Communication Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD

CONTENT

1.Abstract

Non-Intrusive Load Monitoring (NILM) is an innovative approach to understanding and optimizing energy consumption patterns without intrusive equipment. This report delves into the significance and methodologies of NILM, revealing its effectiveness in disaggregating loads and identifying individual appliances. The project's objectives centered around applying NILM techniques to real-world scenarios, providing insights into energy efficiency and conservation. The findings highlight the potential of NILM in optimizing energy usage and its implications for sustainable energy management. As a result, the report offers practical recommendations for enhancing energy efficiency, showcasing the promising role of NILM in shaping the future of energy conservation and smart grid technologies.

2. Problem Statement

The challenge of developing an efficient Non-Intrusive Load Monitoring (NILM) system lies in addressing various hurdles, including accurate data acquisition, appliance signature extraction, real-time monitoring, enhanced accuracy, scalability, user-friendly interfaces, privacy measures, cost-effectiveness, and integration with smart grids. This project aims to overcome these obstacles to create a robust NILM solution, empowering users to optimize energy consumption and promote sustainable energy practices.

3. Introduction

A way to understand and save energy in your home or workplace without having to change your appliances or use fancy gadgets. That's what Non-Intrusive Load Monitoring (NILM) is all about. It's like having an energy detective that can figure out which devices are using power, and it does this without bothering you or your appliances. In this report, we're going to take a closer look at how NILM is making a big difference in how we manage energy. Think of it as a superpower for a world that wants to use energy wisely and take care of our planet. We'll show you how NILM works, what secrets it can uncover, and how it can help us save both money and energy. You'll get a behind-the-scenes look at the methods that make NILM so smart, the fascinating things it can find out, and some practical tips for using it. Our hope is that the information we share in this report will empower people, businesses, and governments to make wiser and more eco-friendly energy choices. NILM is a true game-changer in the world of energy management and conservation.

4. Objective

i. **Understand how NILM works:** We want to help you understand the magic behind NILM. It's like peeling back the curtain to see how it figures out which appliances are using electricity without needing any special tools.

ii. **Show NILM Benefits:** We'll show you why NILM is so cool. It can help you save money on your energy bills and reduce your impact on the environment. Who wouldn't want that?

iii. **Teach Practical Usage:** It's not just about knowing how NILM works; it's about knowing how to use it in your daily life. We'll give you tips and tricks to make the most of it.

iv. **Promote Eco-Friendly Choices:** We all want to do our bit for the planet. NILM can be your sidekick in making more eco-friendly energy choices, whether you're at home or work.

v. **Make Energy Management Simple**: Energy management can sound complicated, but it doesn't have to be. We'll break it down into easy steps and language so that everyone can understand and join the energy-saving superhero team.

5. Literature Review

Energy disaggregation, also known as nonintrusive load monitoring (NILM) or nonintrusive appliance load monitoring (NIALM), is the practice of examining variations in the voltage and current entering a home to determine which appliances are utilized there and how much energy each one uses. Utility providers utilize NILM-equipped electric meters to track the precise consumption of electricity in various houses. NILM is thought to be a less expensive option than mounting separate monitors on every appliance. But it raises questions about privacy.

5.1 Measurement Techniques in NILM:

Precise measurement methods are the cornerstone of NILM. These include multi-point sensing, intricate measurements, straightforward measurements, and cutting-edge methods like the Switch Continuity Principle and Measurement Reading Flux. To collect high-resolution load data, researchers have created tools like the Precision Ammeter and Arduino Power Meter Reader. These methods are crucial for gathering the comprehensive appliance-level data that serves as NILM's foundation.

5.2 Load Monitoring Models and Inference

NILM breaks down appliances using a variety of models and inference techniques. A foundational element of many NILM algorithms are Hidden Markov Models (HMMs). The capabilities of HMMs are expanded by Factorial HMM and HSMM, enabling more precise event detection. The Combined Load HMM is an interesting concept that combines several load sources to increase accuracy. The efficiency of NILM algorithms is further increased by utilizing the Viterbi Algorithm with Sparse Transitions.

5.3 Challenges and Future Directions

Researchers struggle with problems pertaining to scalability, appliance detection, sensor noise, and real-time monitoring. These difficulties highlight the necessity of continuous innovation in NILM. In the future, NILM may be integrated with smart grids and the Internet of Things (IoT), and novel event detection methods for increasingly sophisticated appliance behaviours may be investigated.

6. Methodology

6.1 About Data Set: The REDD (Reference Energy Disaggregation Data Set) dataset is a publicly available dataset used for research in energy disaggregation, also known as non-intrusive load monitoring (NILM). It provides time-series data on electrical consumption at the appliance level, which can be used to develop algorithms and models for disaggregating total electricity consumption into individual appliance usage.

- It has 2 types of data
    1. is low frequency
    2. high frequency
- low frequency calculates the consumed energy by appliances per 3secs and high frequency calculates the consumed energy by appliances per 5 millisecs,
- We are using low frequency data. It has 6 houses.
- from 6 houses we are taking 2 houses,
    1. house-1.
    2. house-2.
- house-1 has 20 channels which means mains_1, and mains_2 and remaining channels are 18 individual appliances.
- house-2 has 11 channels which means mains_1, mains_2 and remaining channels are 9 individual appliances.

6.2 Data Preprocessing:

Merging data: Construct File Paths: The function begins by constructing the file path based on the 'house' parameter to access data files associated with that house.

Read Initial Data: The function reads the data from the first data file, 'channel_1.dat.' It interprets the contents of this file as a time series dataset and creates a Pandas DataFrame to hold this data. The DataFrame is given specific column names and data types to ensure consistency in the dataset.

Iterate Through Data Files: The function iterates through other data files for the same house, typically named 'channel_2.dat,' 'channel_3.dat,' and so on. For each file, it reads the data and interprets it as another DataFrame, maintaining the same column names and data types as the initial data.

Merge DataFrames: After reading each data file, the function merges the newly acquired data into the initial DataFrame. It performs inner joins on the 'unix_time' column, ensuring that only rows with matching time values are combined. This process gradually builds a consolidated time series dataset with multiple channels of data, all aligned by their respective 'unix_time' values.

Timestamp Conversion: Once all the data files have been merged into a single DataFrame, the function converts the 'unix_time' column into timestamp values. This is often helpful for time-based data analysis.

Set Timestamp as Index: The timestamp column is set as the index of the DataFrame, which is a common practice when working with time series data. This change makes it easier to perform time-based operations and align data points.

Remove Unnecessary Columns: The function may remove redundant columns, such as the 'unix_time' and 'timestamp' columns, as they are no longer needed, having served their purpose in merging and timestamp conversion.

Return the Resulting DataFrame: The final step is to return the resulting DataFrame. This DataFrame represents a consolidated time series dataset for the specified house, where timestamps serve as the index, and various channels of data are available as columns. This dataset is typically ready for further analysis, visualization, or any other tasks related to time series data processing.

7. Machine Learning Model

7.1 Decision tree regression: A decision tree regression model is used for predicting the energy consumption of specific appliances (e.g., refrigerator) in House 1.
- The code trains the model on the training data and evaluates it on the validation set.
- It varies the parameter `min_samples_split` to optimize the model.
- The model with the lowest validation loss is selected as the best model.

The prepared dataframe has been trained and tested on three scenarios namely,

i. HOUSE 1 data : Data =>[features = mains_1,mains_2],[output = refrigerator data]

ii. HOUSE 2 data(trained on house 1 data) : Data =>[features = mains_1,mains_2],[output = refrigerator data]

iii. HOUSE 1 data[training and prediction of all other appliances): Data =>[features = mains_1,mains_2],[output = iterative data from oven to washer dryer excluding refrigerator]

This data is trained over the Decision Tree Regression Model**.**

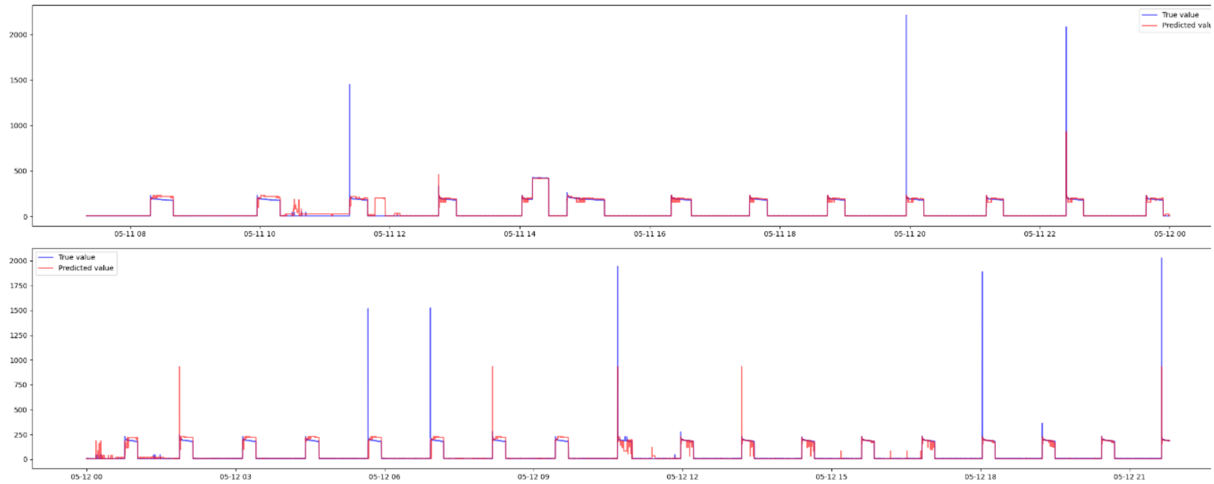
7.2 Model Evaluation
- The code calculates the Mean Square Error (MSE) and Mean Absolute Error (MAE) to evaluate the performance of the model on the test set.
- It also plots the real and predicted energy consumption for the refrigerator in House 1 over six days.

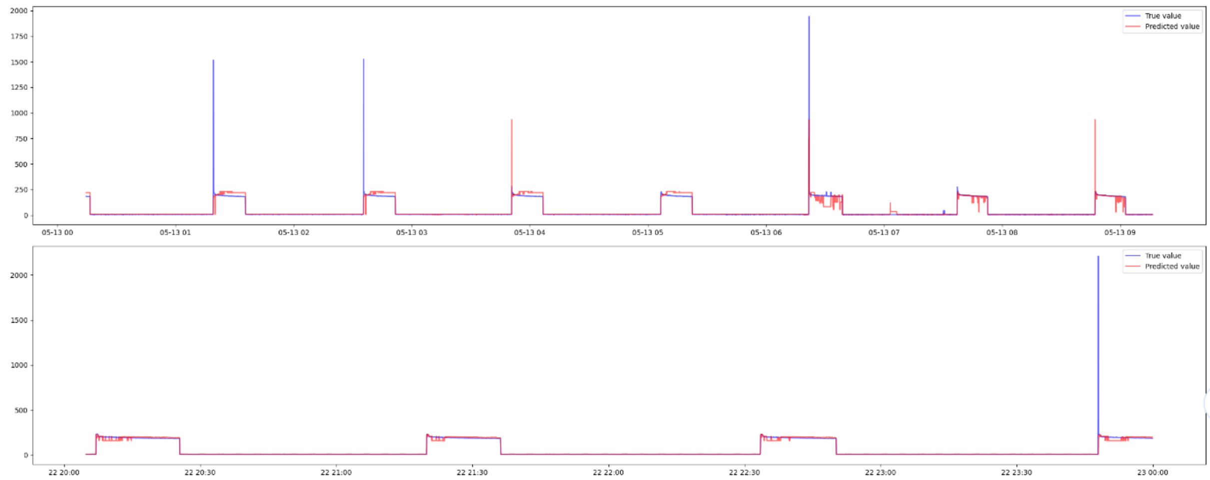
7.3 Transfer Learning to House 2
- The best model (trained on House 1) is used to predict the energy consumption of the refrigerator in House 2.
- It calculates the MSE and MAE for the predictions in House 2.
- It plots the real and predicted consumption for the refrigerator in House 2.

7.4 Expanding to Other Appliances
- The code extends the decision tree model to predict the energy consumption of other appliances in House 1.
- The process is similar to the refrigerator prediction.
- It calculates and prints MSE and MAE for each appliance.
- Finally, it plots the real and predicted consumption for each appliance in House 1 over six days.

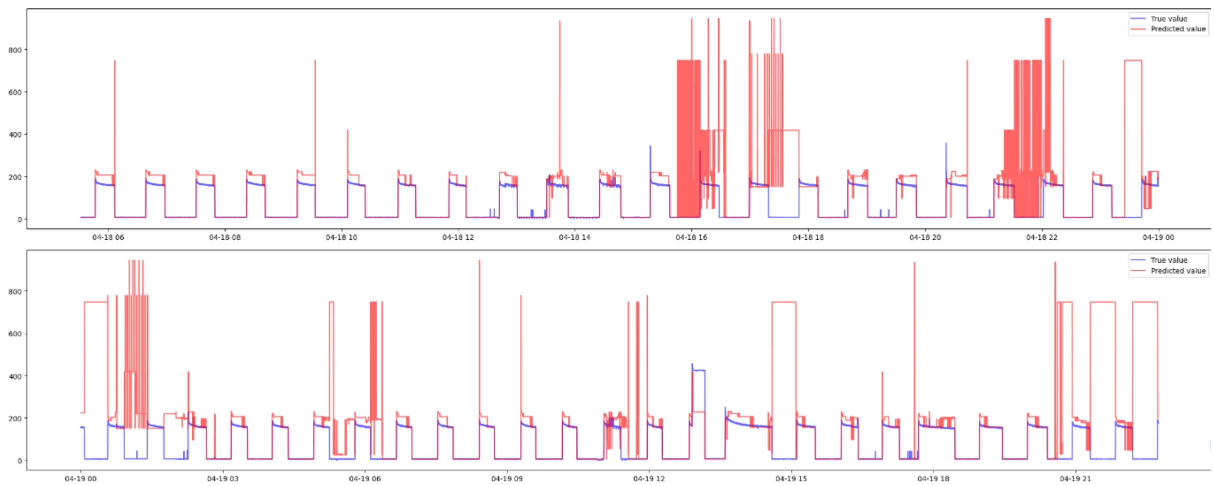
7.5 RESULTS:

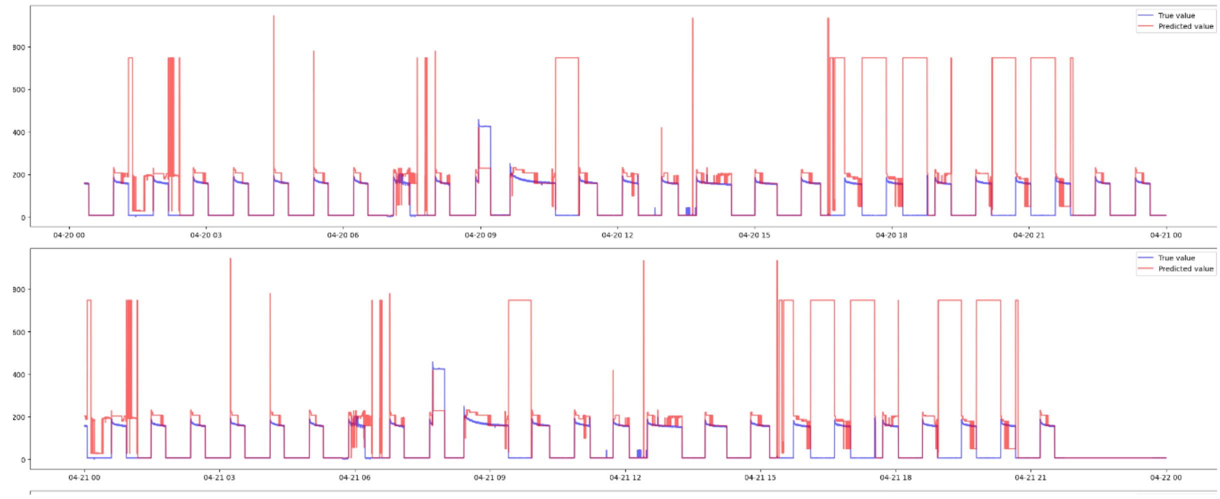
HOUSE 1 DATA: REAL AND PREDICTED VALUES GRAPH OF REFRIGERATOR:





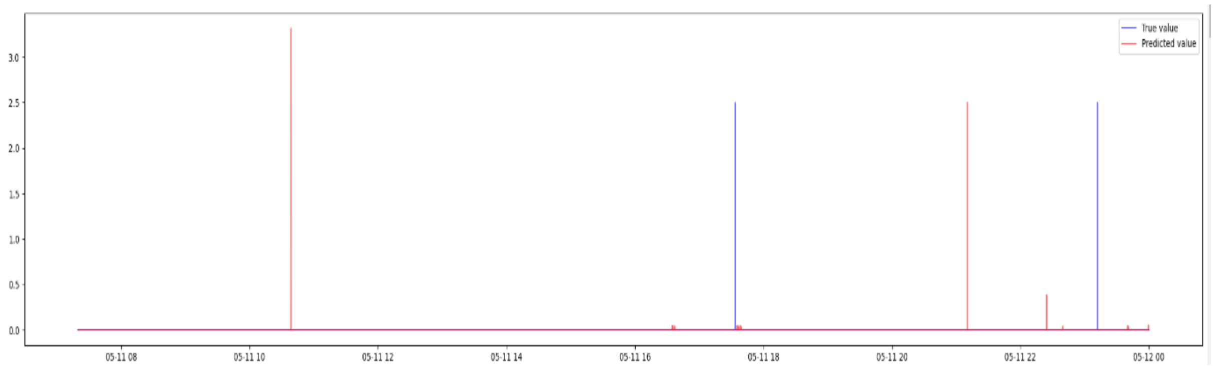
HOUSE 2 DATA: REAL AND PREDICTED VALUES OF REFRIGERATOR BASED ON TRAINED MODEL FROM HOUSE 1 DATA:



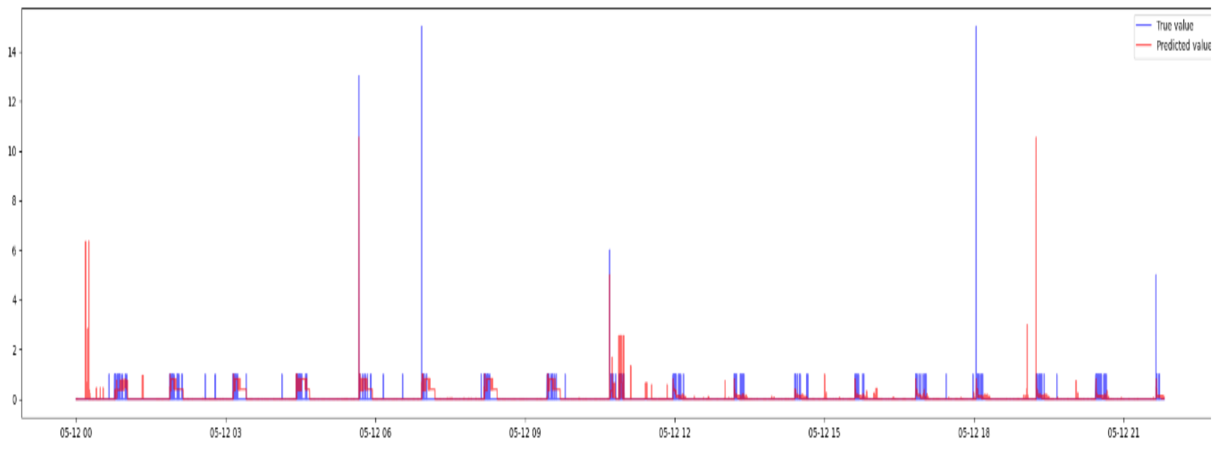


HOUSE 1 DATA: REAL AND PREDICTED DATA OF ALL OTHER APPLIANCES:

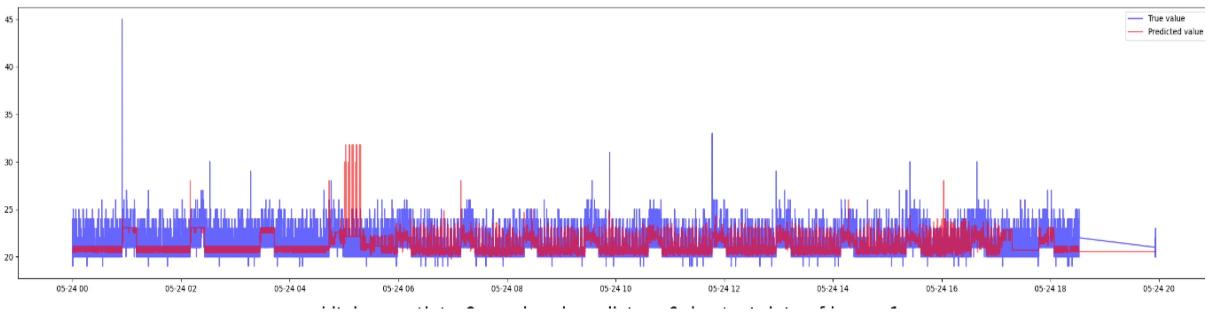
OVEN:



DISH WASHER:



KITCHEN OUTLETS:



In this way the Decision tree model(with least loss among minimum split samples) has been over 20 appliances and validated using the test data.

LOSS COMPARISON AMONG THREE SCENARIOS

1: HOUSE 1 data (training and testing on house 1 data only)
Mean square error on test set: 1634.5797666188705
Mean absolute error on the test set: 12.686127417077758

2: HOUSE 2 data(trained on house 1 and tested on house 2)
Mean square error on test set: 32245.25362228206
Mean absolute error on the test set: 64.75419454670589

3: HOUSE 1(trained and tested on all other appliances of house 1)
mean of oven_3: 15.63 - mse: 18555.08 - mae: 11.30
mean of oven_4: 17.11 - mse: 7454.75 - mae: 4.92
mean of dishwaser_6: 25.35 - mse: 831.49 - mae: 3.38
mean of kitchen_outlets_7: 21.25 - mse: 4.54 - mae: 1.59
mean of kitchen_outlets_8: 27.71 - mse: 99.51 - mae: 3.43
mean of lighting_9: 28.29 - mse: 1594.43 - mae: 23.99
mean of washer_dryer_10: 3.07 - mse: 934.44 - mae: 2.44
mean of microwave_11: 18.92 - mse: 12442.92 - mae: 13.00
mean of bathroom_gfi_12: 6.73 - mse: 3471.07 - mae: 3.44
mean of electric_heat_13: 0.11 - mse: 0.53 - mae: 0.05
mean of stove_14: 0.10 - mse: 0.23 - mae: 0.04
mean of kitchen_outlets_15: 5.34 - mse: 832.75 - mae: 1.63
mean of kitchen_outlets_16: 1.93 - mse: 861.34 - mae: 0.67
mean of lighting_17: 18.97 - mse: 136.66 - mae: 3.18
mean of lighting_18: 15.68 - mse: 383.82 - mae: 13.12
mean of washer_dryer_19: 0.00 - mse: 0.00 - mae: 0.00
mean of washer_dryer_20: 27.54 - mse: 2087.58 - mae: 1.79

8. Deep Learning Model

We are using Fully Connected Neural Network (FCNN) and Long Short-Terms Memory (LSTM) Deep Learning models to perform NILM on electricity consumption data. Both models are trained on data from one household and tested on data from another household.

8.1 Fully Connected Neural Network Model :

- Here we defined a Fully connected layer with multiple dense layers. The network takes two input features ('mains_1' and 'mains_2') and aims to predict the consumption of the 'refrigerator' appliance.
- The model uses dropout and ReLU activation functions.
- We used the Mean Squared Error (MSE) as the loss function and Adam optimizer for training.
- Training is performed on the data related to 'house 1'.
- The best model is saved during training based on validation performance.
- You load the saved model and make predictions on the test data. The model's performance is evaluated using MSE and MAE (Mean Absolute Error).

8.2 Data Preprocessing and Training (X_train1, y_train1, X_test1, y_test1):

- We preprocess the data by creating sequences of features (X) and the target appliance's consumption (y) for training and testing.
- Then we flatten the sequences into input data (X_train_fc) suitable for the FCNN model.
- Training and validation loss curves are plotted.

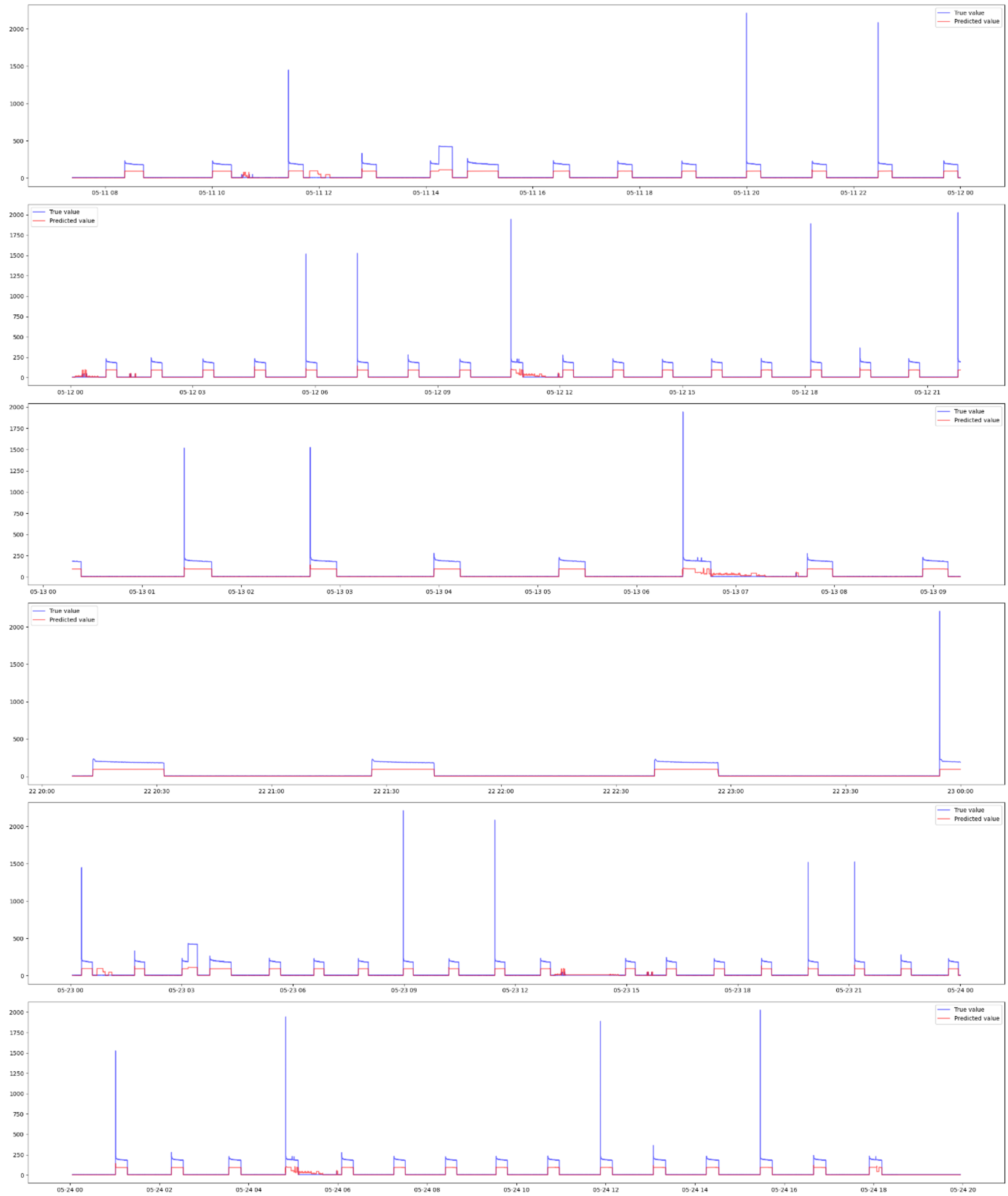
Testing on House 2:

We applied the model trained on 'house 1' to test data from 'house 2'. This represents a cross-household testing scenario. Then we calculated the model's performance on 'house 2

8.3 Results First FCNN model:

Training on house-1 and test on house-1

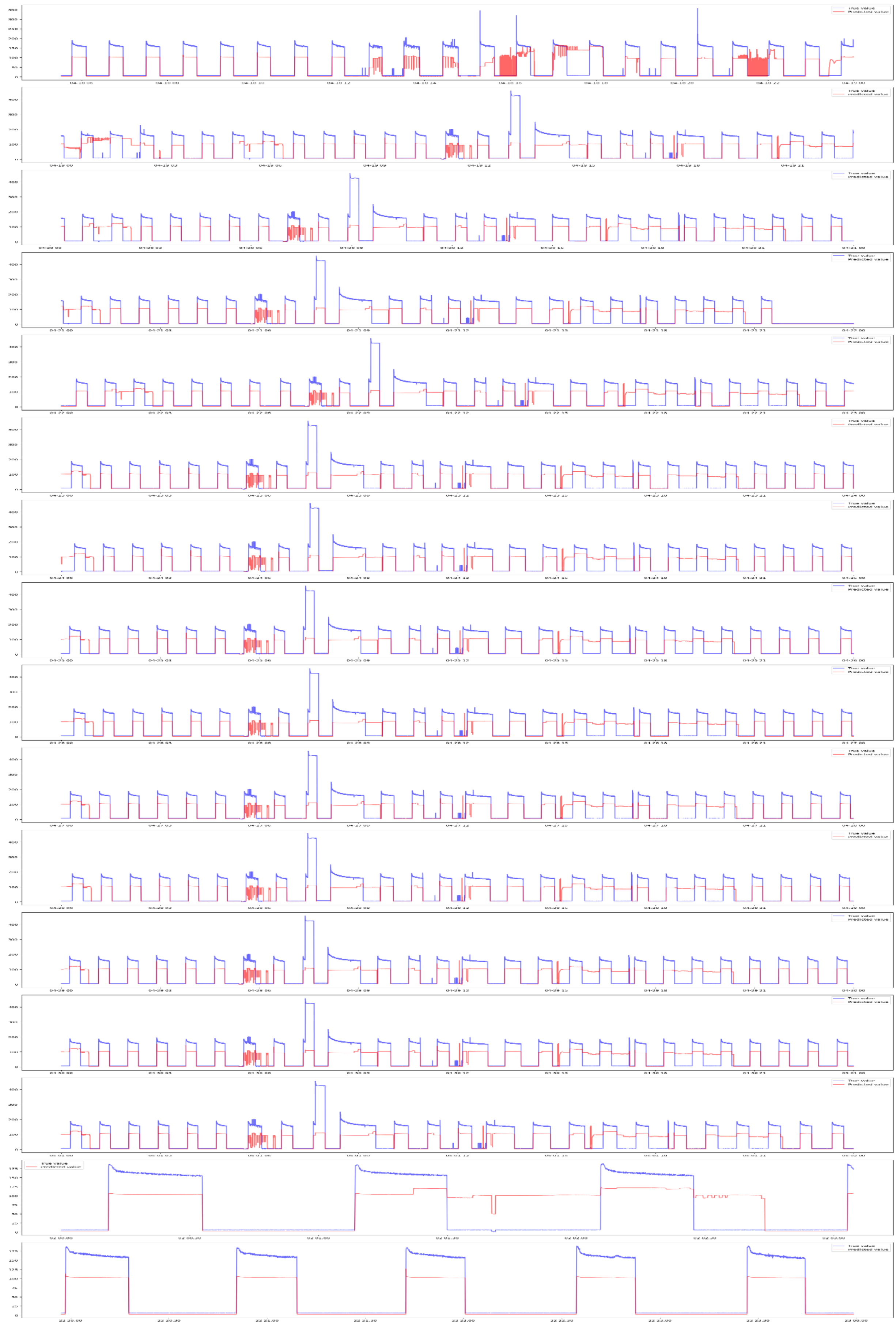
FC model: real and predict Refrigerator on 6 test day of house 1



Mean square error on test set: 4753.797455237583

Mean absolute error on the test set: 35.35885758645867

FC model for refrigerator: train on house 1, predict on house 2

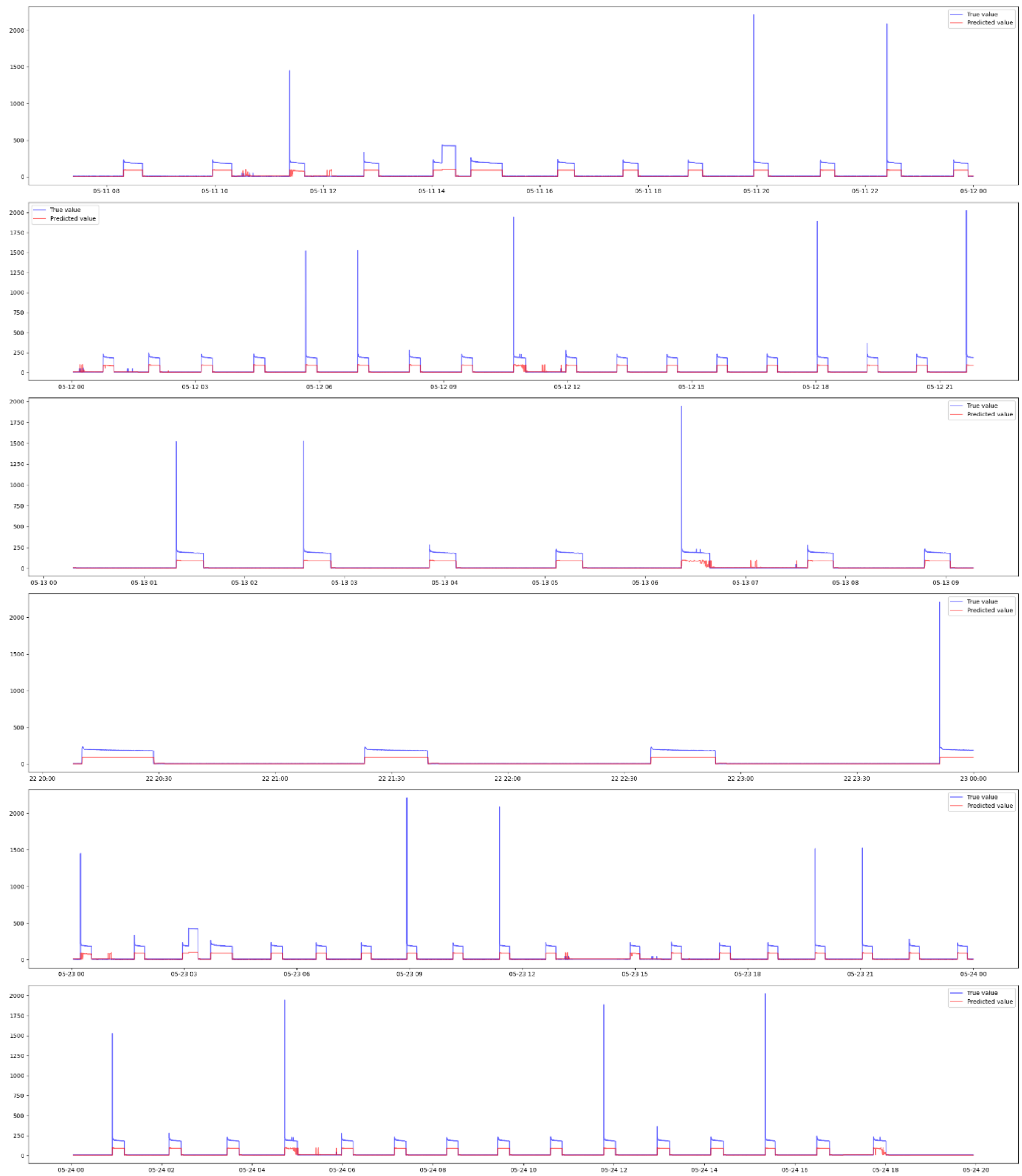


Training on house-1 and predicting on house-2
Mean square error on test set: 3824.605244696324
Mean absolute error on the test set: 39.6381635565504

Second FCNN Model (Rebuild Model):

Training on house-1 and predicting on house-1

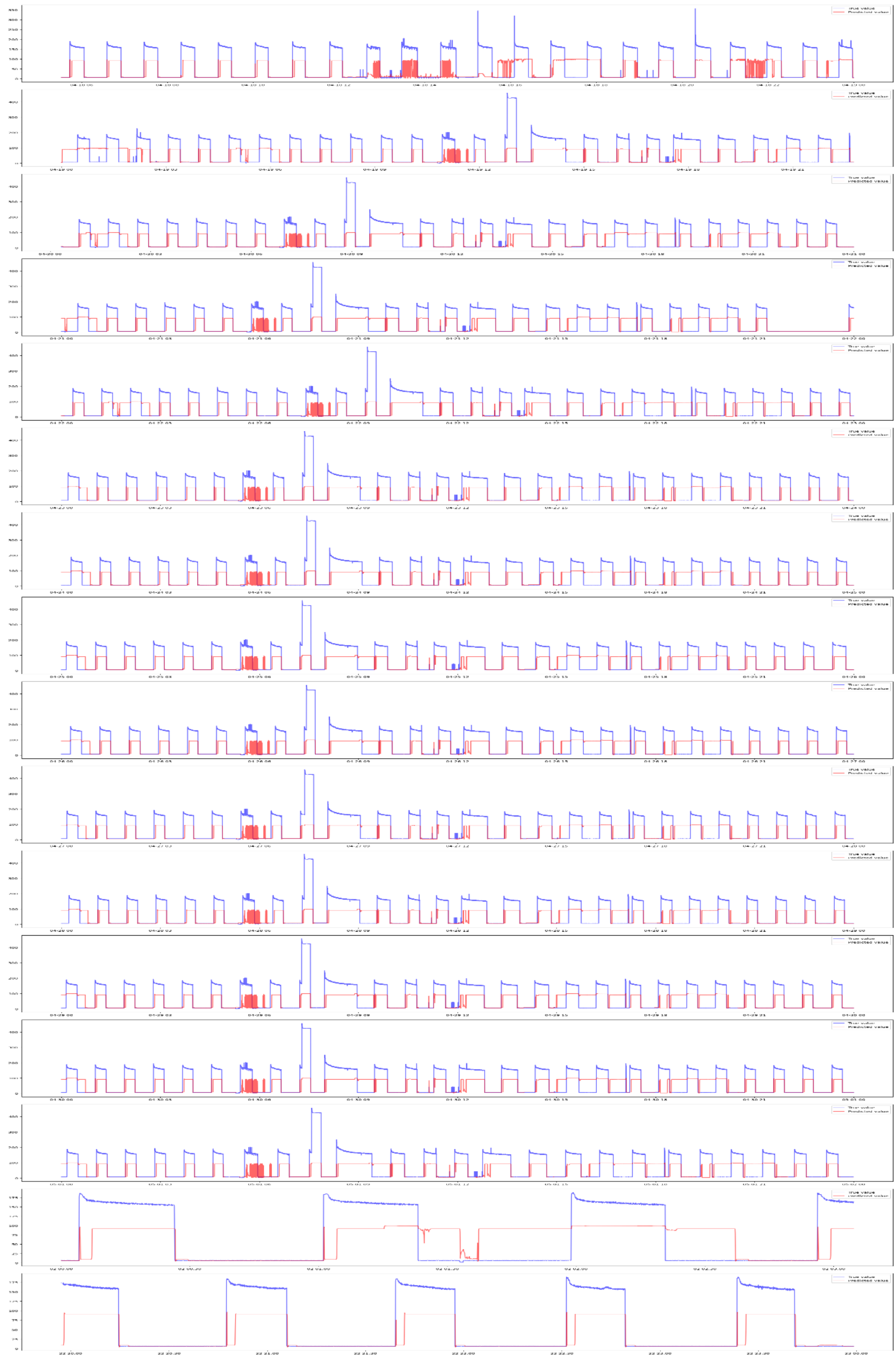
Real and predict Refrigerator on 6 test day of house 1



Mean square error on test set: 6000.929270239348
Mean absolute error on the test set: 38.229261880170654

Training on house-1 and predicting on house-1

FC model look back 50 for refrigerator: train on house 1, predict on house 2



Mean square error on test set: 5667.917472587572
Mean absolute error on the test set: 48.038417214872624

Rebuild the FNN model for House 1, but this time using the past 50 consecutive main values to predict the present value of the refrigerator.

9. Long Short-Term Memory (LSTM) Model (model)

- ➔ We define an LSTM model with dropout layers.
- ➔ The LSTM model is similar to the FCNN model but uses LSTM layers, which are suitable for sequence data.
- ➔ Training is done using data from 'house 1', and the model's performance is evaluated on 'house 1' test data.

Testing LSTM Model on House 2:
Similar to the FCNN, we test the LSTM model on data from 'house 2' and evaluate its performance. Simulating a real-world scenario where you train a model on one household and apply it to another.

10. Results and Comparisons

Machine learning Model

- HOUSE 1 data (training and testing on house 1 data only):
- Mean square error on test set: 1634.5797666188705.
- Mean absolute error on the test set: 12.686127417077758.
- HOUSE 2 data(trained on house 1 and tested on house 2):
- Mean square error on test set: 32245.25362228206
- Mean absolute error on the test set: 64.75419454670589

Deep Learning Model

- Training on house-1 and test on house-1:
- Mean square error on test set: 4753.797455237583.
- Mean absolute error on the test set: 35.35885758645867.
- Training on house-1 and predicting on house-2:
- Mean square error on test set: 3824.605244696324.
- Mean absolute error on the test set: 39.6381635565504

In The Machine Learning Model, On the house1 data we trained a model when we tested for the house1, the mean squared error and the mean absolute error are 1634.5797666188705 and 12.686127417077758. When we tested the same model (which was trained from the house1 data) for the house2 data the mean squared error and the mean absolute error are 32245.25362228206 and 64.75419454670589. mean squared error and the mean absolute error increase.

In The Deep Learning Model, On the house1 data we trained a model when we tested for the house1, the mean squared error and the mean absolute error are 4753.797455237583 and 35.35885758645867. When we tested the same model (which was trained from the house1 data) for the house2 data the mean squared error and the mean absolute error are 3824.605244696324 and 39.6381635565504. In the DL model mean squared error decreased but mean absolute error slightly increased.
So we can say generalization of the Deep Learning Model is good as compare to the Machine Learning Model. Machine learning model work well when we tested for the same house. But when we tested for the different houses it did not perform well. Deep learning model perform very well for the same house as well as for the differents house. So we ca make assumption DL model performance is better than the ML model in this case.

10. Conclusion

In this project, we used both Machine Learning (ML) and Deep Learning (DL) techniques to understand how people use electricity in their homes without needing to install special equipment. The ML model worked well when we tested it in the same house where it was trained, which can help manage energy use within a single household. But when we tried it in different houses, it didn't perform as well, so it needs to be improved for cross-house predictions. The DL models, especially the first Fully Connected Neural Network (FCNN) model, showed good results in predicting energy use in a single house. It could also be used in other houses, but with a bit less accuracy. Overall, this project shows that Non-Intrusive Load Monitoring (NILM) has the potential to help households better manage and reduce their energy use. In the future, we aim to focus on refining our models and expanding the dataset to encompass a broader range of household types and usage patterns. By doing so, we anticipate even greater accuracy and utility.

11. References

- https://en.m.wikipedia.org/wiki/Nonintrusive_load_monitoring#:~:text=Nonintrusive%20load%20monitoring%20(NILM)%2C,as%20their%20individual%20energy%20consumption
- https://jack-kelly.com/files/writing/neural_nilm.pdf
- https://www.youtube.com/watch?v=AilBDbMA0Rk
- http://makonin.com/
- http://redd.csail.mit.edu/
- https://docs.openenergymonitor.org/electricity-monitoring/index.html
- https://drive.google.com/drive/folders/1Cfgg0iFtTrChIssLzfcafnd-kY5cMtzd?usp=sharing
- https://www.kaggle.com/datasets/alimasaleem/nilm-data
- https://drive.google.com/drive/folders/1AiJ5GEJTIvMrJJVaXVdxYlyIO9BwIJjM?usp=sharing