ENGINEERING
W Booth School of Engineering
Practice and Technology

McMaster University

SEP 769 – Cyber Physical Systems

Final Project Report (Deep Learning)

Energy Consumption of Household Appliances

Group Members:

Karan Patel: 400136862

Dhruv Upadhyay: 400490730

Harshil Soni: 400420715

Dhruv Padia: 400420904

Instructor:

Anwar Mirza, PhD, DIC

1. Introduction

This document serves as a comprehensive exposition of a formal project aimed at discerning and analysing household appliance energy consumption patterns through the integration of advanced smart home technology and meticulously collected weather data. The central objective of this endeavour is to conduct an exhaustive analysis of energy utilization trends across a spectrum of household appliances, facilitated by a meticulously curated dataset enriched with real-time meteorological information. This dataset encompasses a time frame of 350 days, providing minute-by-minute granularity essential for comprehending the intricate dynamics of energy consumption. Noteworthy parameters, including indoor temperature, humidity levels, and precise energy consumption data for a diverse array of appliances, are captured. The principal aspiration of this project is the development of robust regression models calibrated to predict energy usage patterns specific to individual appliances. These models are constructed using data gathered from an array of sensors, promising an innovative perspective on the complex interplay between domestic environments, technological ecosystems, and the variable influences of weather phenomena.

2. Project Background

The increasing global emphasis on sustainability and energy efficiency has prompted a growing interest in understanding the intricate dynamics of household energy consumption. With a significant portion of energy usage attributed to residential spaces, the need to uncover consumption patterns and devise effective strategies for optimizing energy utilization has never been more critical. Traditional methods of energy analysis often fall short in capturing the nuanced interplay of factors influencing energy consumption within homes. This project arises in response to these challenges, aiming to harness the power of smart home technology and weather data to gain a deeper understanding of energy consumption trends.

Smart home systems, equipped with a plethora of sensors and interconnected devices, have revolutionized the way we interact with our living spaces. These systems offer a unique opportunity to gather real-time data on various environmental conditions and appliance operations. By combining this wealth of information with weather data, including temperature and humidity, a more holistic perspective on energy consumption can be achieved. The integration of such diverse datasets not only enhances our understanding of energy use but also paves the way for the development of predictive models that can anticipate consumption patterns based on contextual factors.

This project's foundation rests upon a comprehensive dataset meticulously compiled over the course of 350 days. This dataset captures minute-by-minute records of household temperature, humidity, and the energy consumption of a wide array of appliances. The fusion of smart home data and weather information serves as the bedrock for our investigation into the intricate relationship between environmental conditions and appliance-specific energy usage.

3. Problem Statement

At the crux of this project lies a fundamental problem statement: How can the intricate relationship between household appliance energy consumption patterns and the confluence of weather conditions and indoor environmental variables be effectively characterized and modelled? The project addresses the intricate challenge of unravelling the multifaceted dependencies that govern energy utilization trends, offering a proactive avenue for managing energy resources within a domestic context. By devising accurate regression models that encapsulate the intertwined effects of temperature, humidity, and appliance-specific attributes, the project endeavours to furnish homeowners with informed insights, thereby empowering judicious practices in energy consumption. This formal endeavour holds the potential to contribute substantively to the domains of energy conservation and smart home technology implementation.

4. Project Objectives

- <u>Thoroughly Load and Analyse the Smart Home Dataset in Conjunction with Weather Information to Extract Profound Insights</u>: Explore the integrated dataset meticulously, aiming to extract comprehensive insights that illuminate the intricate interplay between household appliance energy consumption and varying weather conditions.

- Utilize Effective Data Visualization Techniques to Scrutinize Dataset Characteristics and Decipher Complex Appliance Energy Consumption Trends: Employ advanced visualization methods to dissect the dataset's inherent patterns, comprehending the distribution of data and unravelling the nuanced energy consumption trends among diverse appliances.
- Execute Robust Pre-Processing Protocols to Manage Missing Data and Normalize Dataset Attributes, Paving the Way for Effective Model Construction: Apply meticulous pre-processing techniques, adeptly managing any missing data points, while concurrently normalizing dataset attributes to establish an optimal foundation for subsequent modelling endeavours.
- Construct and Fine-Tune a Suite of Diverse Machine Learning Models, including Linear Regression, Decision Trees, and Random Forests, to Accurately Predict Appliance Energy Consumption: Ingeniously formulate, calibrate, and fine-tune a range of machine learning models, encompassing Linear Regression, Decision Trees, and Random Forests, with the primary objective of attaining precise predictions of appliance energy consumption.
- Architect and Execute an Advanced Feedforward Neural Network (FNN) Model to Provide an Intriguing Comparative Analysis with Traditional Models: Engineer a sophisticated Feedforward Neural Network (FNN) model, implementing it meticulously to facilitate a captivating comparative exploration against conventional machine learning models, enriching the understanding of predictive capabilities.
- Conduct Rigorous Performance Assessment and Metric-Based Comparison of Diverse Models to Ascertain Optimal Predictive Accuracy: Engage in a comprehensive evaluation process, meticulously assessing the performance of the diverse models, guided by appropriate metrics, to discern the model or models that best encapsulate appliance energy consumption patterns.

5. Theory and Dataset

- Theory

  In the realm of cyber-physical systems, the integration of smart technologies and real-time data has ushered in transformative capabilities for understanding and optimizing various processes, including energy consumption in households. Smart home technology, equipped with sensors and interconnected devices, has paved the way for granular data collection and analysis within domestic environments. This synergy between technology and data forms the theoretical foundation for our project, enabling us to unravel the intricate relationships between household appliance energy consumption and external factors such as weather conditions.

  Regression models, a key component of our analysis, facilitate the prediction of energy usage patterns based on a combination of variables. Linear Regression, Decision Trees, and Random Forests are established methodologies for modelling relationships between variables. Additionally, the integration of advanced machine learning models like the Feedforward Neural Network (FNN) allows us to explore more complex dependencies within the data.
- Dataset

  Our project's foundation rests upon a meticulously curated dataset sourced from Kaggle, a prominent platform for data science enthusiasts and researchers. This dataset encompasses a span of 350 days and offers a detailed account of energy consumption patterns in households, captured at a 1-minute interval. Comprising readings from smart meters and enriched with weather information for a specific region, this dataset presents a holistic view of energy consumption dynamics within real-world settings.

  The dataset is provided in CSV format, allowing for easy access and manipulation using data processing tools and programming languages. It includes a multitude of attributes, with each row representing a minute of recorded data. Key attributes encompass:

  Timestamp: We have ranged the data from 01/01/2016, 05:00 AM for better understanding.

  Energy Consumption: The energy consumed by various household appliances during the specified minute in kW.

  Indoor Temperature: The indoor temperature of the household during that minute.

  Humidity: The humidity level within the household at that particular time.

  Weather Data: Weather-related information such as outdoor temperature, humidity, and other meteorological conditions for the specific region.

The integration of indoor environmental conditions and external weather data provides a unique opportunity to explore the intricate interplay between these factors and appliance energy consumption. By analyzing this dataset, we aim to uncover patterns, correlations, and insights that contribute to a more profound understanding of household energy utilization.

6. Project Implementation

- Data Analysis and Visualization

  Load the smart home dataset and weather information using Python and Pandas.

  Perform exploratory data analysis (EDA) to identify trends and correlations.

  Create visualizations (line plots, scatter plots, etc.) to visualize appliance energy usage patterns.

  Analyse the correlation between weather conditions and energy consumption of appliances.

- Data Pre-processing

  Handle missing or erroneous data points through imputation or removal.

  Normalize the data to ensure consistent scales for different features.

  Split the dataset into training and testing sets for model evaluation.

- Regression Models

  Implement Linear Regression, Decision Trees, and Random Forests models using scikit-learn.

  Perform hyperparameter tuning for each model to optimize performance.

  Evaluate the regression models using metrics such as Mean Squared Error (MSE) and R-squared.

- Feedforward Neural Network (FNN) Model

  Design a Feedforward Neural Network architecture using TensorFlow/Keras.

  Experiment with different activation functions and layer configurations.

  Train the FNN model using the pre-processed dataset and validate its performance.

- Feature Importance Analysis

  Identify the most relevant features for accurate energy consumption predictions.

  Use techniques like Recursive Feature Elimination (RFE) or feature importance scores from models.

  Provide insights into which features or data combinations are crucial for specific appliance predictions.

7. Explanation of Source Code

Step:1 Importing the necessary libraries and importing the csv file.

```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
        %matplotlib inline
        import seaborn as sns
        from sklearn.model_selection import train_test_split
        from sklearn.linear_model import LinearRegression
        from sklearn.metrics import mean_squared_error
        from sklearn.preprocessing import StandardScaler
        import numpy as np
```

```
In [2]: data = pd.read_csv('HomeC.csv', low_memory=False)
        data.head()
```

Step:2 Knowing the datatypes and removing the data row with garbage values.

```
In [ ]: data.info()
```

```
In [ ]: data.tail()
```

```
In [ ]: data=data[:-1]
        data.tail()
```

Step:3 Converting the time values and ranging it from 01-01-2016, 5:00 AM

```
In [ ]: #given that the time is in UNIX format, let's check
        time_index = pd.date_range('2016-01-01 05:00', periods=len(data),  freq='min')
        time_index = pd.DatetimeIndex(time_index)
        data = data.set_index(time_index)
```

Step:4 Combining the similar equipment readings and checking again if the values are in required format.

```
In [ ]: data.columns
```

```
In [ ]: data.columns = [col.replace(' [kW]', '') for col in data.columns]
        data.columns
```

```
In [ ]: data['Furnace'] = data[['Furnace 1','Furnace 2']].sum(axis=1)
        data['Kitchen'] = data[['Kitchen 12','Kitchen 14','Kitchen 38']].sum(axis=1)
        data.drop(['Furnace 1','Furnace 2','Kitchen 12','Kitchen 14','Kitchen 38','ico
        data.head(5)
```

```
In [ ]: data.info()
```

```
In [ ]: data['cloudCover'].unique()
```

```
In [ ]: data['cloudCover'].replace(['cloudCover'], method='bfill', inplace=True)
        data['cloudCover'] = data['cloudCover'].astype('float')
```

Step:5 Plotting the data to the know the relation each data with time and neglecting the non impacting parameter.

```
In [ ]: energy_per_day = energy_data.resample('D').sum()
        plt.figure(figsize=(20,10))
        plt.title("Overall energy consumption per day")
        sns.lineplot(data = energy_per_day.filter(items=[ 'Dishwasher', 'Home office',
        #'Dishwasher', 'Furnace', 'Home office', 'Fridge','Wine cellar', 'Garage door'
```

```
In [ ]: energy_per_day = energy_data.resample('D').sum()
        plt.figure(figsize=(20,10))
        plt.title("Overall energy consumption per day")
        sns.lineplot(data = energy_per_day.filter(items=[ 'Barn', 'Well', 'Garage door
        #'Dishwasher', 'Furnace', 'Home office', 'Fridge','Wine cellar', 'Garage door'
```

```
In [ ]: energy_per_month = energy_data.resample('M').sum() # for energy we use sum to
        plt.figure(figsize=(20,10))
        sns.lineplot(data= energy_per_month.filter(items=[  'Dishwasher', 'Furnace', '
                                                   'Wine cellar', 'Garage door', 'Kitchen',
                                                   'Microwave', 'Living room']) , dashes=Fal
```

Step:6 Knowing the relationship between sensor data and consumption by plotting the graph.

```
In [ ]: sns.regplot(x=energy_per_day['Kitchen'], y= sensor_per_day['temperature'])
```

```
In [ ]: sns.regplot(x=energy_per_day['Fridge'], y= sensor_per_day['temperature'])
```

```
In [ ]: sns.regplot(x=energy_per_day['Furnace'], y= sensor_per_day['temperature'])
```

Step:7 Feature Importance, scaling down the values from actual to values between 0 & 1 and plotting the value.

```
In [ ]: # Standardize the features
        scaler = StandardScaler()
        X_train_scaled = scaler.fit_transform(X_train)
        X_test_scaled = scaler.transform(X_test)
```

```
In [ ]: # Linear Regression Model
        model_lr = LinearRegression()
        model_lr.fit(X_train_scaled, y_train)
```

```
In [ ]: y_pred_lr = model_lr.predict(X_test_scaled)
```

```
In [ ]: mse_lr = mean_squared_error(y_test, y_pred_lr)
```

```
In [ ]: feature_importance_lr = np.abs(model_lr.coef_)
        feature_names = X.columns

        # Visualize feature importance
        plt.figure(figsize=(12, 6))
        plt.barh(feature_names, feature_importance_lr)
        plt.xlabel('Feature Importance')
        plt.ylabel('Features')
        plt.title('Linear Regression - Feature Importance')
        plt.show()
```

Step:8 After feature importance, trainig the 80% of the total data on different algorithms and mapping the accuracy of each model and then testing on 20% of the data.

```
In [ ]: # Random Forest Regressor
        model_rf = RandomForestRegressor(n_estimators=100, random_state=42)
        model_rf.fit(X_train_scaled, y_train)
        y_pred_rf = model_rf.predict(X_test_scaled)
        r2_rf = r2_score(y_test, y_pred_rf)
        print(f"Random Forest R-squared: {r2_rf}")
```

```
In [ ]: # Support Vector Regressor
        model_svr = SVR()
        model_svr.fit(X_train_scaled, y_train)
        y_pred_svr = model_svr.predict(X_test_scaled)
        r2_svr = r2_score(y_test, y_pred_svr)
        print(f"SVR R-squared: {r2_svr}")
```

```
In [ ]: # K-Nearest Neighbors Regressor
        model_knn = KNeighborsRegressor()
        model_knn.fit(X_train_scaled, y_train)
        y_pred_knn = model_knn.predict(X_test_scaled)
        r2_knn = r2_score(y_test, y_pred_knn)
        print(f"K-Nearest Neighbors R-squared: {r2_knn}")
```

```
In [ ]: # Multi-Layer Perceptron (Feedforward Neural Network)
        from keras.models import Sequential
        from keras.layers import Dense

        model_fnn = Sequential()
        model_fnn.add(Dense(128, input_dim=X_train_scaled.shape[1], activation='relu')
        model_fnn.add(Dense(64, activation='relu'))
        model_fnn.add(Dense(1))  # Output layer with linear activation

        model_fnn.compile(optimizer='adam', loss='mean_squared_error')
        model_fnn.fit(X_train_scaled, y_train, epochs=20, batch_size=32, verbose=1)

        y_pred_fnn = model_fnn.predict(X_test_scaled)
        r2_fnn = r2_score(y_test, y_pred_fnn)
        print(f"FNN R-squared: {r2_fnn}")
```

Note: Source Code and PDF attached for output of each module.

8. Results and Discussion

Learning Algorithm	KNN Regressor	Radom Forest Regressor	Support Vector Regressor	FNN (Multi-Layer Perception)
Accuracy	91.57%	97.22%	--	94.08%

9. Recommendations for Future Work

Future studies could concentrate on improving forecast accuracy and comprehending model behaviour. Use advanced time-series models to capture temporal trends, ensemble methods for better forecasts, and interpretable AI techniques for insights. For energy-efficient applications, investigate model generalisation and integration into real-time systems.

10. References

- https://www.kaggle.com/datasets/taranvee/smart-home-dataset-with-weather-information