Hourly Electricity Consumption and Production

Name: Prabhat Kumar Mishra

Reg.no: 22MCA0036

Name: Sanghamitra Das

Reg.no: 22MCA0049

Name: Ankita Mishra

Reg.no: 22MCA0156

Abstract

The purpose of this project was to predict energy consumption using the data of Finland's transmission system operator. The objective of this project was to test if a machine learning model can yield good enough results in a complex forecasting problem, exploring machine learning techniques and developing a data-driven model for forecasting energy. We used a long-short-term memory (LSTM) model to train the data. The model was evaluated using root mean squared error (RMSE) to be directly comparable to energy readings in the data. The result shows that electricity consumption can be predicted using machine-learning algorithms

Keywords: Machine Learning, Energy Consumption Prediction, Time Series Forecasting, Long-Short Term Memory.

Introduction

Electric generation and consumption are essential for contemporary society's sustainable growth. Accurate forecasting of future electricity use and production is essential for policy-making, resource management, and infrastructure planning due to the rising demand for energy and the scarcity of available resources. Large datasets can include complex patterns that can be analyzed and predicted using data mining techniques. In this data mining research, we seek to investigate the connections between a number of variables, including climate, population expansion, economic growth, and power generation and consumption. In order to create a predictive model that can predict future patterns in electricity consumption and production, the project will employ historical data on electricity production and consumption, as well as external influences. This study will offer insightful information about future energy needs.

Data mining

The process of analyzing enormous amounts of data to discover patterns, relationships, and insights that can be used in decision-making is known as data mining. Finding relevant data, projecting future trends, and sorting through enormous databases using statistical and machine-learning techniques are all part of this process. Data mining is used in many sectors, including business, healthcare, finance, and government. It can be used to identify fraud, anticipate illness outbreaks, identify client purchase trends, and improve resource allocation. The phases of the data mining process include data collection, data cleaning and preprocessing, data transformation, and data modeling, to name just a few. Once a model has been developed, it can be applied to make predictions about the future or classify fresh data using the trends found in the training sample. Data in general

Need for data mining

Data mining is important for a number of reasons, including:

- **Discovering hidden patterns and insights:** Businesses and organizations can use data mining to sift through enormous amounts of data in search of patterns, correlations, and trends that might not be immediately apparent. By exposing these hidden insights, organizations can make better decisions and acquire a competitive advantage.

- **Enhancing decision-making:** By studying enormous amounts of data, data mining can help businesses and organizations make better decisions. For example, data mining can be used to ascertain customer preferences, project demand for goods and services, and enhance pricing strategies.

- **Detecting fraud and anomalies:** Data mining can be used to uncover anomalies and fraudulent activities that standard detection methods might miss. For instance, companies that deal with credit cards use data mining.

- **Predicting future trends:** data mining can be used to forecast future trends and produce precise predictions by analyzing historical data. For instance, data mining can be used to estimate market prices, predict disease outbreaks, and anticipate consumer behavior.

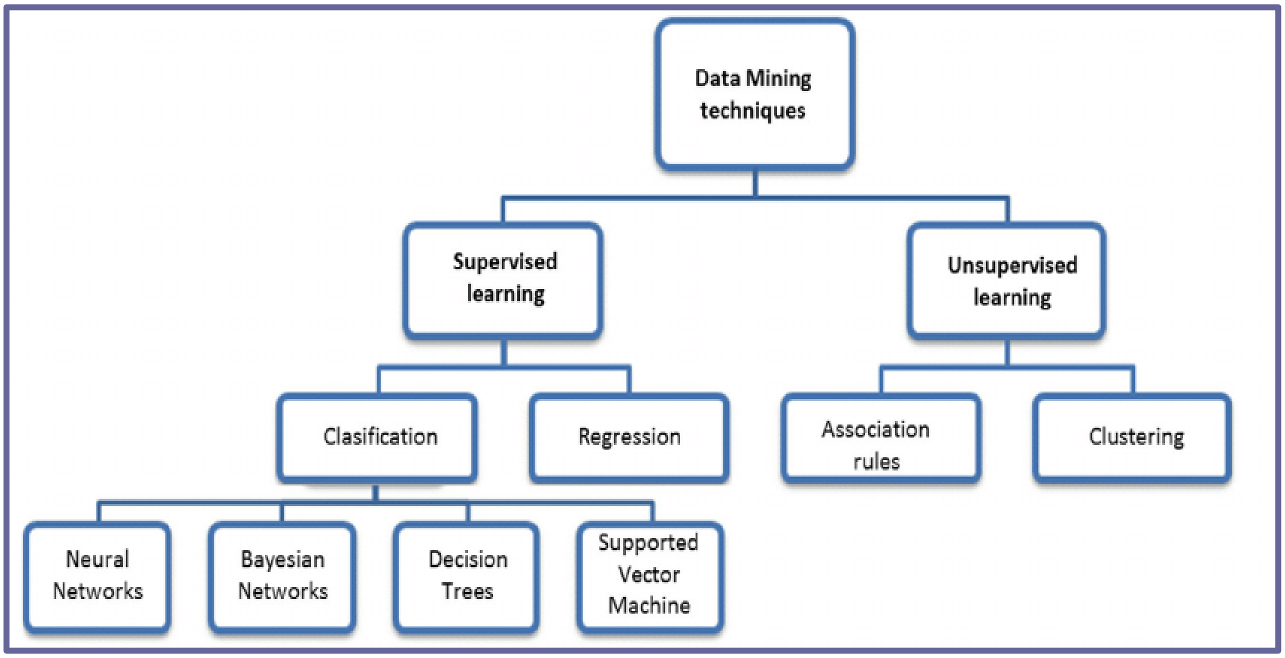
Tasks involved in data mining

Data mining involves a variety of tasks, which may vary depending on the specific application or industry. However, some common tasks that are typically included in data mining are:

- **Data cleaning and preprocessing:** This involves preparing data for analysis by removing noise, missing values, outliers, and irrelevant information.
- **Data exploration:** This involves exploring the data to gain a better understanding of its characteristics, such as its distribution, range, and correlations.
- **Pattern identification:** This involves identifying patterns and trends in the data using statistical and machine-learning techniques, such as clustering, classification, and regression.
- **Association rule mining:** This involves discovering associations and relationships between variables in the data, such as identifying which products are often purchased together.
- **Anomaly detection:** This involves identifying unusual or unexpected data points or patterns in the data that may indicate fraud, errors, or other issues.
- **Predictive modeling:** This involves building models to predict future outcomes based on historical data, such as forecasting sales or predicting customer behavior.
- **Text mining:** This involves analyzing text data, such as customer reviews or social media posts, to extract useful information and insights.
- **Visualization:** This involves creating visual representations of the data to help understand and communicate the results of data mining analyses.

In general, these processes are iterative and may call for numerous iterations of data pretreatment, modeling, and validation to produce insightful findings and useful information.

Data mining techniques



Knowledge discovery in database

KDD stands for Knowledge Discovery in Databases. It refers to the process of discovering useful knowledge or patterns from large amounts of data stored in databases. KDD is an iterative and interactive process that involves multiple steps, including data cleaning, data integration, data selection, data transformation, data mining, pattern evaluation, and knowledge representation.

The KDD process starts with identifying the problem and understanding the domain and the data that will be used. Next, the data is preprocessed, which involves cleaning the data, integrating it from multiple sources, selecting the relevant attributes, and transforming the data into a suitable format for analysis.

After data preprocessing, the data mining step is performed, which involves using various techniques to extract patterns or knowledge from the data. Some common data mining techniques include clustering, classification, regression, and association rule mining.

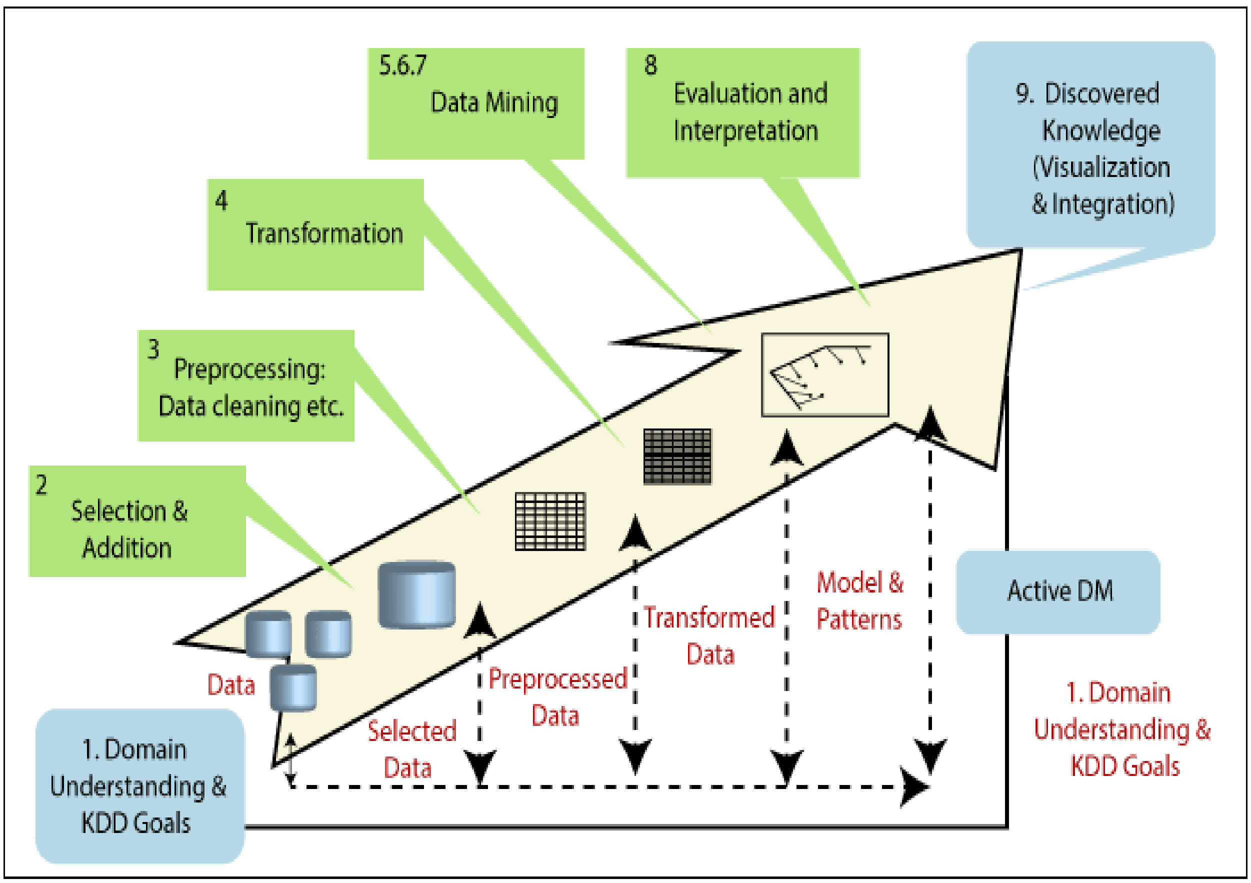
Once patterns have been discovered, the patterns are evaluated and interpreted to determine their usefulness and relevance to the problem being solved. Finally, the knowledge is represented in a way that is understandable and useful for the end user.

The KDD process is a systematic and iterative approach to extracting useful knowledge from large amounts of data. It is widely used in a variety of fields, including business, finance, healthcare, and government, to improve decision-making and gain insights into complex problems.

The steps involved in KDD are –

1. Problem understanding
2. Data Preparation
3. Data selection
4. Data transformation
5. Data mining
6. Pattern evaluation
7. Knowledge representation

Lifecycle of KDD



Advantages and Disadvantages:

Advantages of KDD -

1. **Improves decision-making:** KDD provides valuable insights and knowledge that can help organizations make better decisions.
2. **Increased efficiency:** KDD automates repetitive and time-consuming tasks and makes the data ready for analysis, which saves time and money.
3. **Better customer service:** KDD helps organizations gain a better understanding of their customer's needs and preferences, which can help them provide better customer service.
4. **Fraud detection:** KDD can be used to detect fraudulent activities by identifying patterns and anomalies in the data that may indicate fraud.
5. **Predictive modeling:** KDD can be used to build predictive models that can forecast future trends and patterns.

Disadvantages of KDD -

1. **Privacy concerns:** KDD can raise privacy concerns as it involves collecting and analyzing large amounts of data, which can include sensitive information about individuals.
2. **Complexity:** KDD can be a complex process that requires specialized skills and knowledge to implement and interpret the results.
3. **Unintended consequences:** KDD can lead to unintended consequences, such as bias or discrimination if the data or models are not properly understood or used.
4. **Data Quality:** KDD process heavily depends on the quality of data, if data is not accurate or consistent, the results can be misleading
5. **High cost:** KDD can be an expensive process, requiring significant investments in hardware, software, and personnel.
6. **Overfitting:** The KDD process can lead to overfitting, which is a common problem in machine learning where a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new unseen data.

KDD Vs Data Mining

Parameter	KDD	Data Mining
Definition	KDD refers to a process of identifying valid, novel, potentially useful, and ultimately understandable patterns and relationships in data.	Data Mining refers to a process of extracting useful and valuable information or patterns from large data sets.
Techniques Used	▪ Data cleaning ▪ data integration ▪ data selection ▪ data transformation ▪ data mining ▪ pattern evaluation ▪ knowledge representation and visualization.	▪ Association rules ▪ Classification ▪ Clustering ▪ Regression ▪ decision trees ▪ neural networks ▪ dimensionality reduction.
Output	Structured information, such as rules and models, can be used to make decisions or predictions.	Patterns, associations, or insights can be used to improve decision-making or understanding.
Focus	The focus is on the discovery of useful knowledge, rather than simply finding patterns in data.	The focus is on the discovery of patterns or relationships in data.
Role of domain expertise	Domain expertise is important in KDD, as it helps in defining the goals of the process, choosing appropriate data, and interpreting the results.	Domain expertise is less critical in data mining, as the algorithms are designed to identify patterns without relying on prior knowledge.

Machine Learning

Machine learning is a relatively old field with classical methods and algorithms since the 1960s. This field of study considers a subfield of Artificial Intelligence. It’s defined as the field of study that gives computers the ability to learn without being explicitly programmed and to learn from experience. Machine learning classical algorithms include Naïve Bayes Classifier, Support Vector Machines, and more. These algorithms must be trained on large amounts of data.

Machine Learning models can be divided into the following four categories depending on the amount and type of supervision they need while training.

- **Supervised learning:** The training data is already supplied with predefined labels or outputs. Regression and Classification are the most common problems that are solved by supervised learning.

- **Unsupervised learning**: Where the training dataset is unlabeled, where the model will learn the hidden patterns from the dataset by itself without any supervision. The techniques used in unsupervised learning are Clustering, Association Rule Learning, and Dimensionality Reduction.

- **Semi-supervised learning** is the process of training the model with both labeled and unlabeled data. And this useful method used when extracting features is difficult from the data. (Salian, 2018)

- **Reinforcement learning:** In this model, the agent is developed to interact in a specific environment so that its performance for executing certain tasks improves from the interactions. In other words, the agent decides what to do to perform the given task, where data has no outputs.

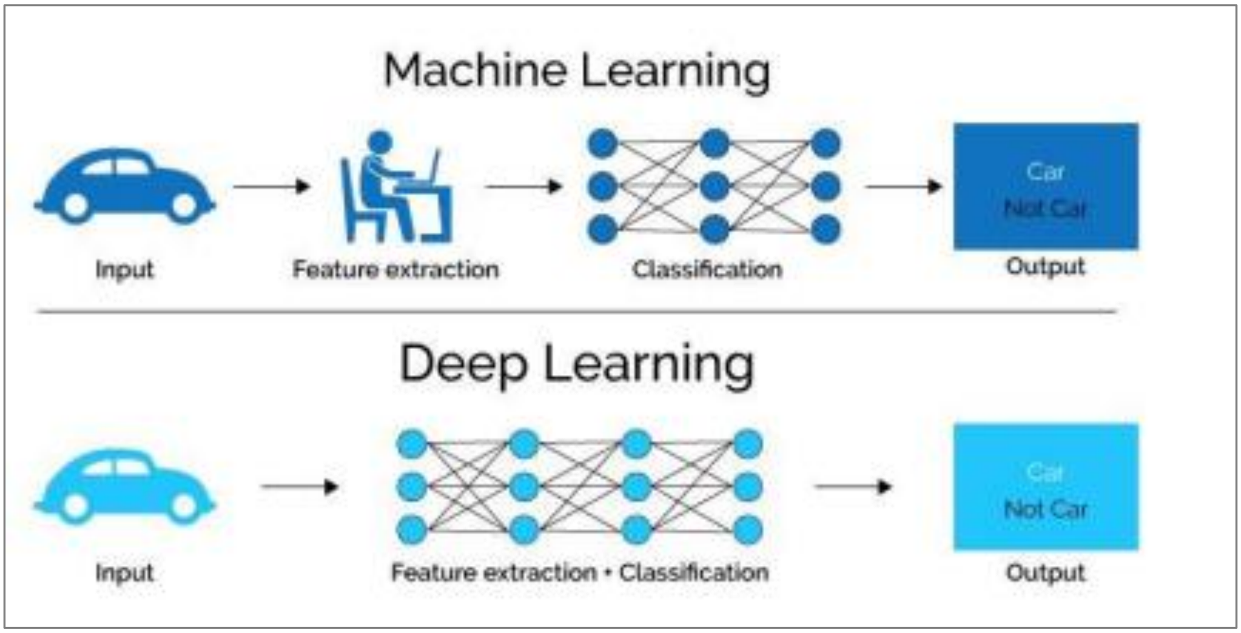
Deep Learning

Unlike machine learning, deep learning is a young subfield of Artificial intelligence and a subfield of machine learning, and it is based on artificial neural networks. Understanding the need for deep learning comes through these points:

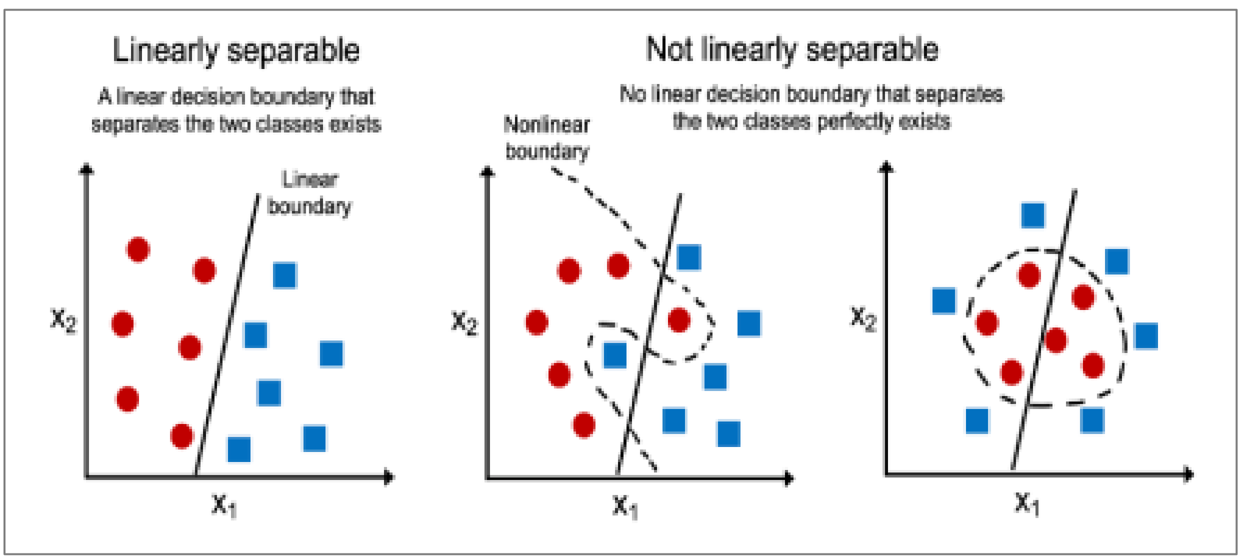
- **Feature Engineering:** It’s the key step in model building, where the required feature extracted for a problem from raw data and then select the important ones that improve the model performance This process can be automated by using deep learning. (Pai, 2020)

- **Decision boundary:** Unlike machine learning, where algorithms learn decision boundaries for linear data, deep learning algorithms can learn decision boundaries for non-linear data.

Feature Engineering Comparison (Pai, 2020)



linear vs non-linear data



Most deep learning neural networks are Feed-Forward Neural Networks. It means that data only flow from input to output such as ANN and CNN.

Artificial Neural Network (ANN)

Artificial Neural Network consists of three main types of layers (input, output, and hidden). Each layer consists of neurons (nodes), which are connected to neurons of the next layer. Each connection is associated with a numeric number called "weight"

Each neuron (node) in the ANN has some weight assigned to it and a transfer function is used to calculate the Weight sum of the inputs and the bias. After the transfer function calculates the sum, the activation function obtains the result until it received the output so it can then fire the appropriate result from the node. (Artificial Neural Networks for Machine Learning – Every aspect you need to know about, 2022)

Convolutional Neural Network (CNN)

Convolutional Neural Network was mainly created for processing data that has a grid pattern such as image and video recognition, recommendation systems, and image analysis and classification. These networks are composed of three types of layers (convolutional, Pooling, and fully connected layer). The first two layers (convolutional, pooling) perform feature extraction, whereas the third layer (fully connected) maps the extracted features into the final output.

Recurrent Neural Networks (RNN)

Recurrent Neural Network is a class of Artificial neural networks where the connections between nodes can create a cycle, where the output from some nodes affects subsequent input to the same nodes. The RNN is different from other neural networks since it has a loop that allows information to be passed from one step of the network to the next.

One of the challenges with Recurrent Neural Networks is vanishing an exploding gradient problem which is a common problem in all types of neural networks. Recurrent Neural Networks have also a cyclic connection making them powerful for modeling sequences. capable of learning order dependence in sequence problems such as time series forecasting.

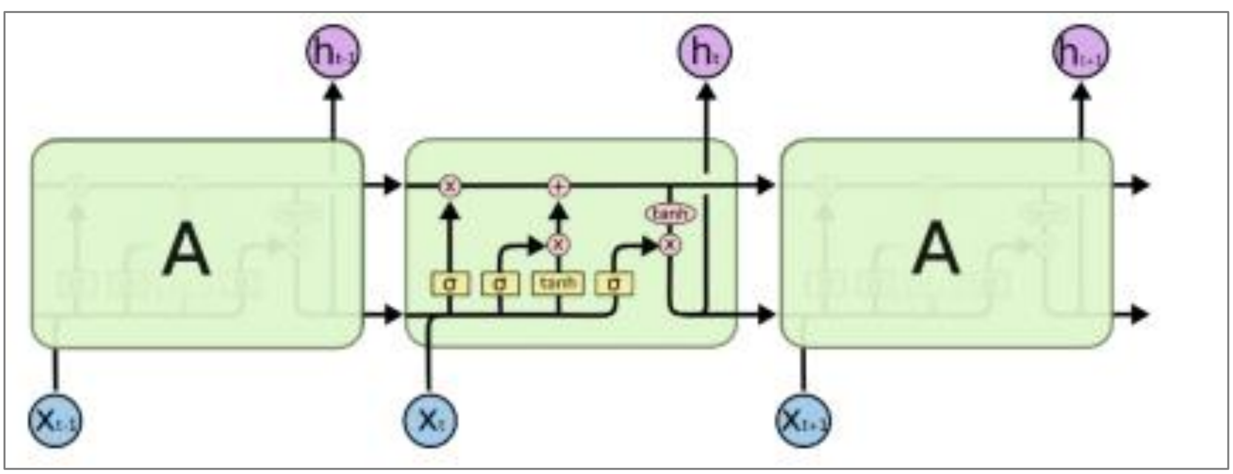
Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a special kind and powerful Recurrent Neural Network approach, which has capable of learning long-term dependencies. The LSTM model has achieved the best results for many problems on sequential data by remembering information for long periods.

The main idea behind LSTM cells is to learn the important parts of the sequence and forget the less important ones by the so-called gates. An LSTM module has a cell state and three gates (input, output, forget) which provides them with the power to selectively learn, unlearn or retain information from each of the units. The cell state helps the information flow through the units without being altered by allowing only a few linear interactions. Each unit has an input, output, and forget gate which can add or remove the information to the cell state.

The forget gate decides which information from the previous cell state should be forgotten for which it uses a sigmoid function. The input gate controls the information flow to the current cell state using a point-wise multiplication operation of ‘sigmoid’ and ‘tanh’ respectively. Finally, the output gate decides which information should be passed on to the next hidden state (Time Series - LSTM Model, 2022). One of the purposes to de- 7 sign LSTMs was to address the vanishing and exploding gradient problems of conventional RNNs

The repeating module in an LSTM



Time Series Forecasting

Time Series Time series data represents a series of data over a specified period. In time series, time is often the independent variable, and the goal is usually to make a forecast for the future.

Time series can be split according to time into: -

Continuous Time Series: Where Observations are measured in continuous time series for each one of time, such as temperature.

Discrete-Time Series: Observations are measured at discrete points in time, such as population change in a city or energy consumption. Discrete-time series observations are usually recorded at equal time intervals (daily, weekly, monthly, yearly). The time series data can be split according to the dependent variables count into univariate, where a single variable is measured over time, and multivariate, where multiple variables are measured over time. While time series analysis is all about understanding the dataset through extracting meaningful statistics and other characteristics of the data; forecasting is all about using a model to predict future values based on previously observed values

Time Series Characteristics

Time Series has main three characteristics which can be modeled to obtain accurate forecasts.

- **Autocorrelation:** is intended to measure the relationship between a variable's present value and any past values that we may have access to.
- **Seasonality:** It is the presence of variations that occur at specific regular intervals of less than a year, such as weekly, monthly, or quarterly.
- **Stationarity:** When time series statistical properties do not change over time. In other words, it has constant mean and variance, and covariance is independent of time.

Time Series Forecast Methods

Time series forecasting is the use of historical information values and associated patterns to predict future ones. As well all forecasting methods are not guaranteed to succeed, so we use machine learning for this purpose.

Time series forecasting is also an important area of machine learning (ML), and it can be cast as a supervised learning problem.

Through the years, many studies evaluate the performance of classical and machine learning methods. Some of these classical methods come to be famous such as: -

- **Autoregressive Moving Average (ARMA)**
- **Autoregressive Integrated Moving Average (ARIMA)**
- **Seasonal Autoregressive Integrated Moving-Average (SARIMA)**

Also, machine learning has:

- **Multi-Layer Perceptron (MLP)**
- **Bayesian Neural Networks (BNN)**
- **Generalized Regression Neural Networks (GRNN)**

Two other modern algorithms are Recurrent Neural Networks (RNN) and Long Short-term Memory (LSTM).

--------------------Proposed work --------------------

1 Dataset

The data was imported from Finland's transmission system operator as a CSV file "events.csv",

There was a total number of 52965 observations and 5 variables in this dataset and no missing values were found.

The minimum load volume is 5341 MWh, and the maximum load volume is 15105 MWh along with an average volume of 9488.750519 MWh.

The model will be trained using a dataset from Fin Grid (Finland's transmission system operator), which contains 6-year of electrical consumption in Finland, where it is univariate time series as it contains one feature.

2 Data Exploration

2.1 First five rows

	Start time UTC	End time UTC	Start time UTC+03:00	End time UTC+03:00	Electricity consumption in Finland
0	2015-12-31 21:00:00	2015-12-31 22:00:00	2016-01-01 00:00:00	2016-01-01 01:00:00	10800.0
1	2015-12-31 22:00:00	2015-12-31 23:00:00	2016-01-01 01:00:00	2016-01-01 02:00:00	10431.0
2	2015-12-31 23:00:00	2016-01-01 00:00:00	2016-01-01 02:00:00	2016-01-01 03:00:00	10005.0
3	2016-01-01 00:00:00	2016-01-01 01:00:00	2016-01-01 03:00:00	2016-01-01 04:00:00	9722.0
4	2016-01-01 01:00:00	2016-01-01 02:00:00	2016-01-01 04:00:00	2016-01-01 05:00:00	9599.0

2.2 Last five rows

	Start time UTC	End time UTC	Start time UTC+03:00	End time UTC+03:00	Electricity consumption in Finland
52961	2021-12-31 16:00:00	2021-12-31 17:00:00	2021-12-31 19:00:00	2021-12-31 20:00:00	11447.0
52962	2021-12-31 17:00:00	2021-12-31 18:00:00	2021-12-31 20:00:00	2021-12-31 21:00:00	11237.0
52963	2021-12-31 18:00:00	2021-12-31 19:00:00	2021-12-31 21:00:00	2021-12-31 22:00:00	10914.0
52964	2021-12-31 19:00:00	2021-12-31 20:00:00	2021-12-31 22:00:00	2021-12-31 23:00:00	10599.0
52965	2021-12-31 20:00:00	2021-12-31 21:00:00	2021-12-31 23:00:00	2022-01-01 00:00:00	10812.0

2.3 Dataset Structural Information

```
Data columns (total 5 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Start time UTC                    52966 non-null  object
 1   End time UTC                      52966 non-null  object
 2   Start time UTC+03:00              52966 non-null  object
 3   End time UTC+03:00                52966 non-null  object
 4   Electricity consumption in Finland  52966 non-null  float64
dtypes: float64(1), object(4)
memory usage: 2.0+ MB
```

2.4 Statistical Description of the Dataset

	Electricity consumption in Finland
count	52966.000000
mean	9488.750519
std	1576.241673
min	5341.000000
25%	8322.000000
50%	9277.000000
75%	10602.000000
max	15105.000000

3 Data Preprocessing

Data Preprocessing refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data

	DateTime	Consumption
0	2016-01-01 01:00:00	10800.0
1	2016-01-01 02:00:00	10431.0
2	2016-01-01 03:00:00	10005.0
3	2016-01-01 04:00:00	9722.0
4	2016-01-01 05:00:00	9599.0

Since we are dealing with time series data, we will edit the index from 1 2 3… --> Date Time format.

Total Number of Years: 7
[2016 2017 2018 2019 2020 2021 2022]

Assumptions

```
By assuming the week starts on Monday and ends on Sunday.
The closest start would be on Monday 4-1-2016
The closest end would be on Sunday 26-12-2021
So, we omit the first 71 rows and the last 121 rows.
```

3.1 Data set after feature extraction

First five rows -

DateTime	Consumption	Month	Year	Date	Time	Week	Day
2016-01-13 02:00:00	11968.0	1	2016	2016-01-13	02:00:00	2	Wednesday
2016-01-13 03:00:00	11643.0	1	2016	2016-01-13	03:00:00	2	Wednesday
2016-01-13 04:00:00	11418.0	1	2016	2016-01-13	04:00:00	2	Wednesday
2016-01-13 05:00:00	11470.0	1	2016	2016-01-13	05:00:00	2	Wednesday
2016-01-13 06:00:00	11557.0	1	2016	2016-01-13	06:00:00	2	Wednesday

Last 5 rows –

DateTime	Consumption	Month	Year	Date	Time	Week	Day
2021-12-06 15:00:00	12855.0	12	2021	2021-12-06	15:00:00	49	Monday
2021-12-06 16:00:00	12799.0	12	2021	2021-12-06	16:00:00	49	Monday
2021-12-06 17:00:00	13004.0	12	2021	2021-12-06	17:00:00	49	Monday
2021-12-06 18:00:00	13255.0	12	2021	2021-12-06	18:00:00	49	Monday
2021-12-06 19:00:00	13303.0	12	2021	2021-12-06	19:00:00	49	Monday

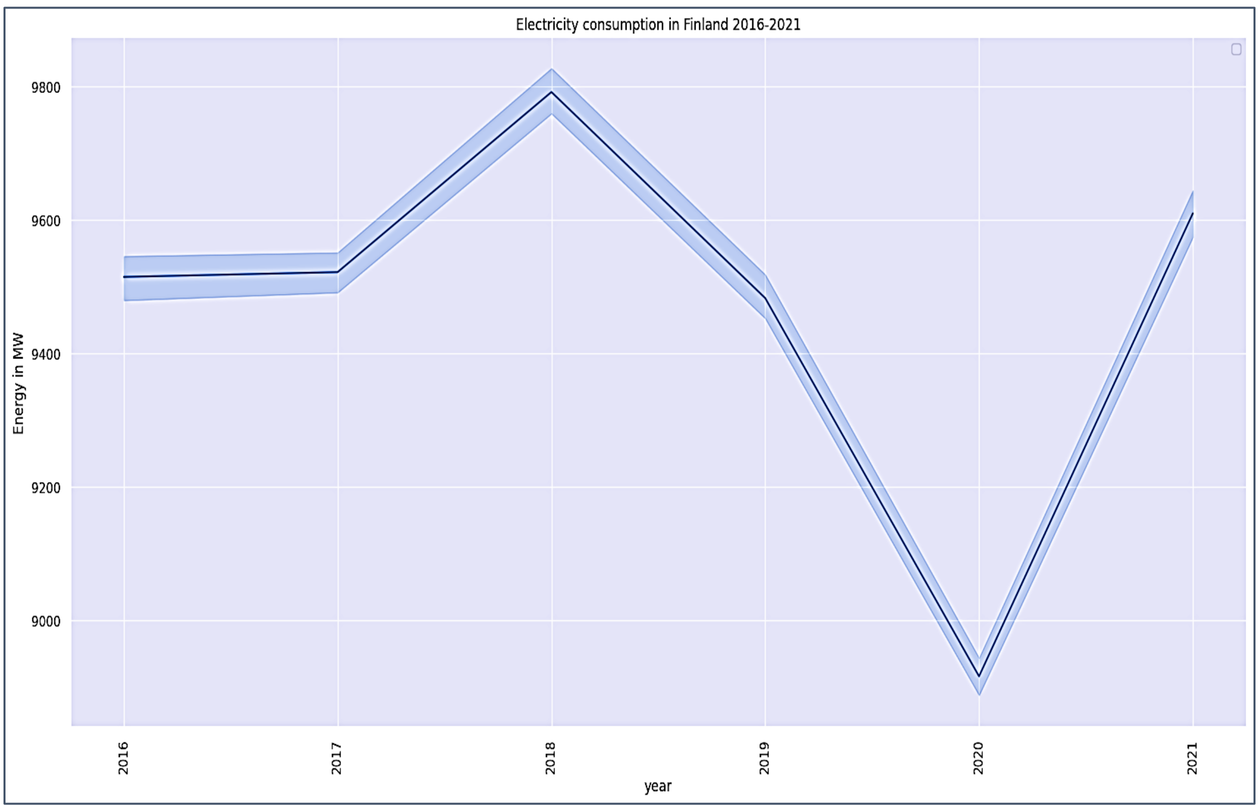
4 Data Visualizations

Accessible methods to examine and comprehend data are more crucial than ever in our increasingly data-driven environment.

The graphic display of information and data is known as data visualization. Data visualization tools offer an easy approach to seeing and comprehending trends, outliers, and patterns in data by utilizing visual elements like charts, graphs, and maps. Additionally, it offers a great tool for staff members or business owners to clearly deliver data to non-technical audiences.

To analyze vast volumes of data and make data-driven decisions, data visualization tools and technologies are crucial in the world of big data.

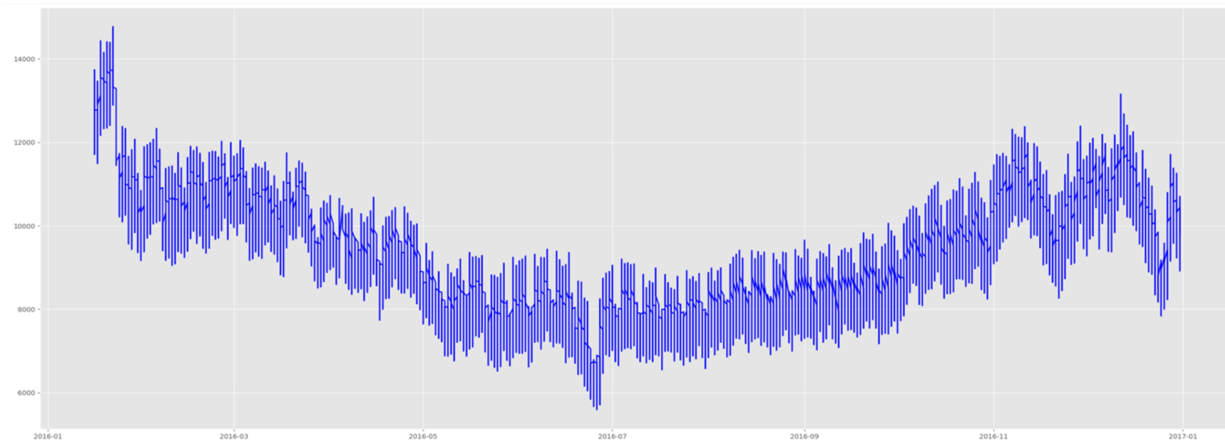
4.1 Energy consumption in Finland from 2016-2021



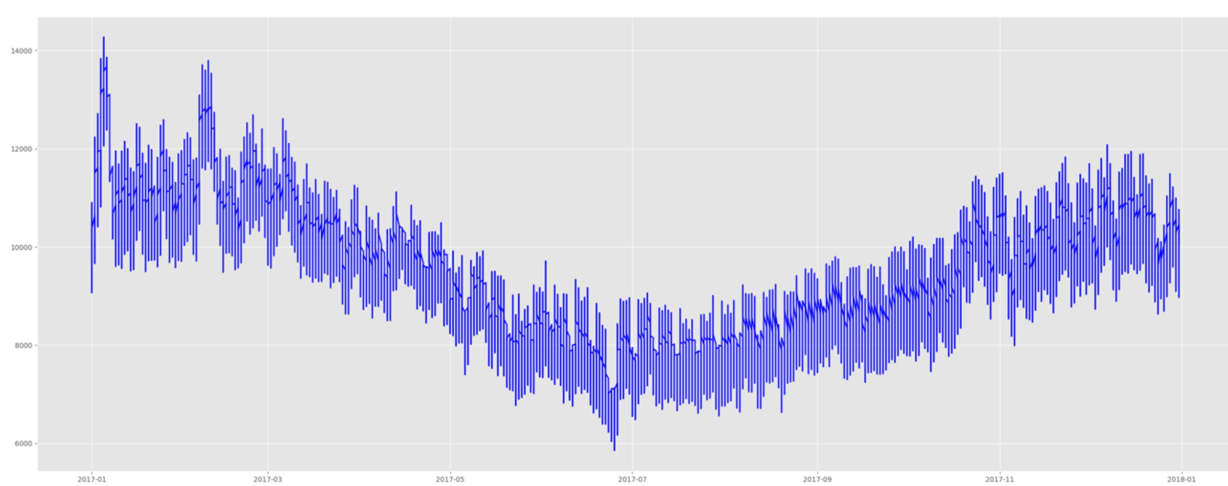
- Maximum energy was consumed during the year 2018.
- Minimum energy was consumed during the year 2020.

4.2 Energy consumption each year

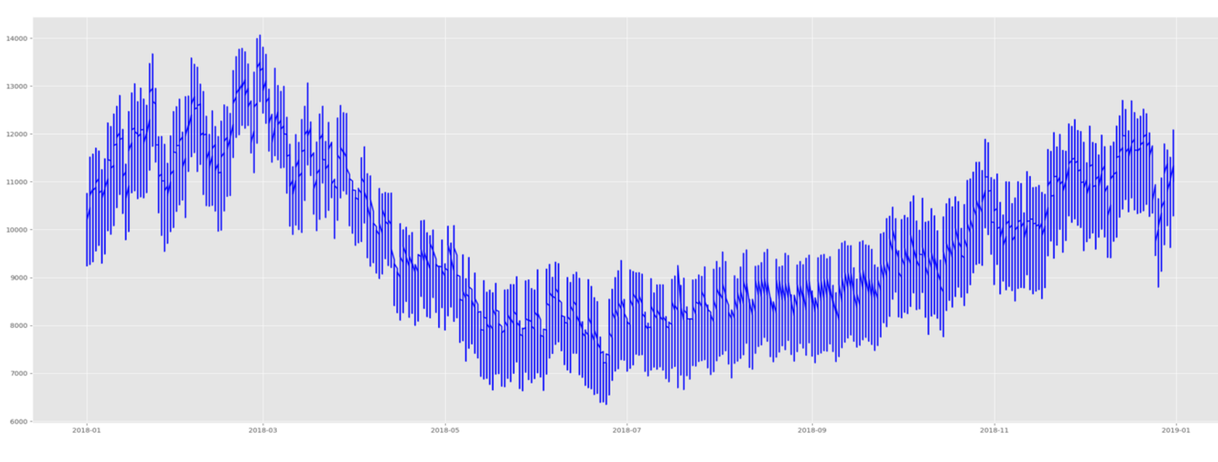
Energy Consumption in 2016



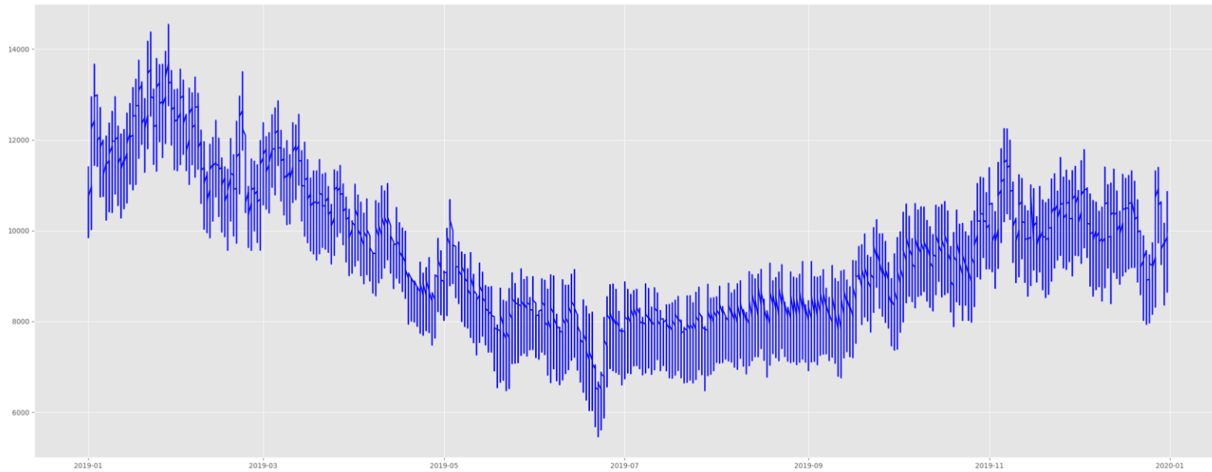
Energy Consumption in 2017



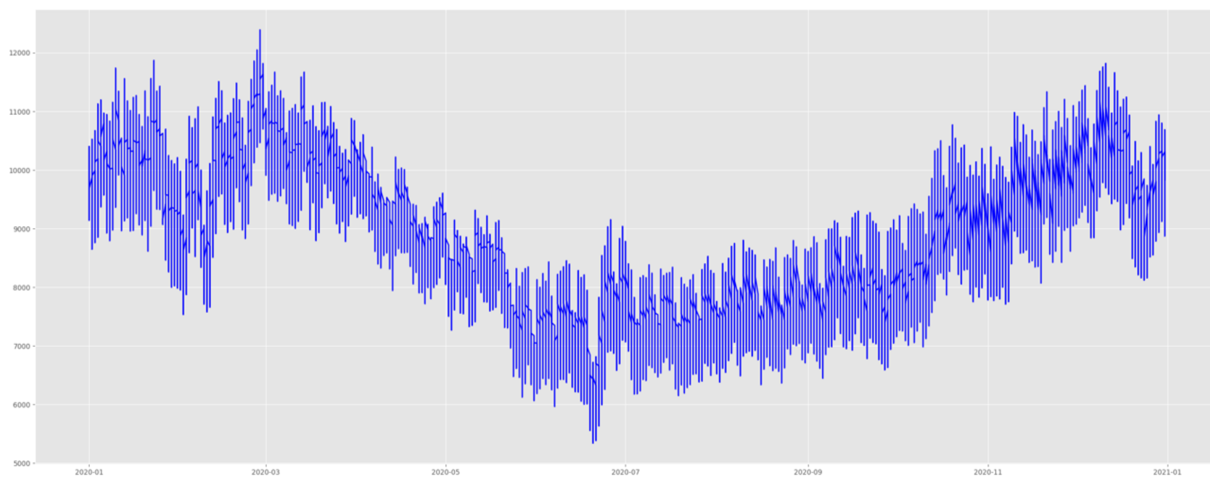
Energy Consumption in 2018



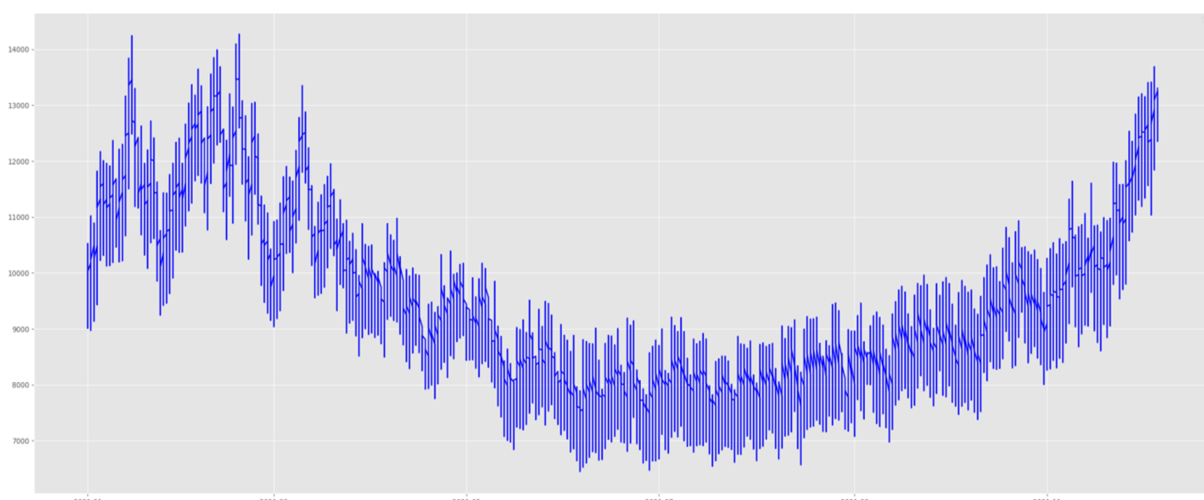
Energy Consumption in 2019



Energy Consumption in 2020

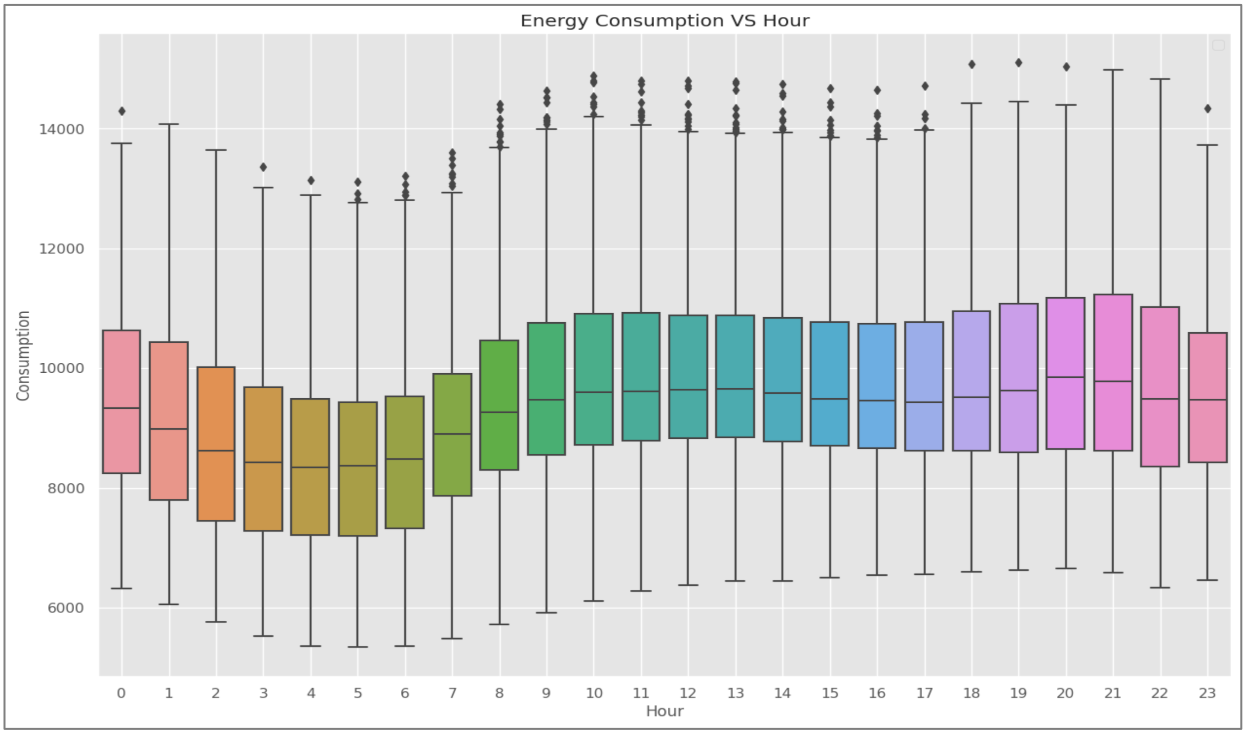


Energy Consumption in 2021

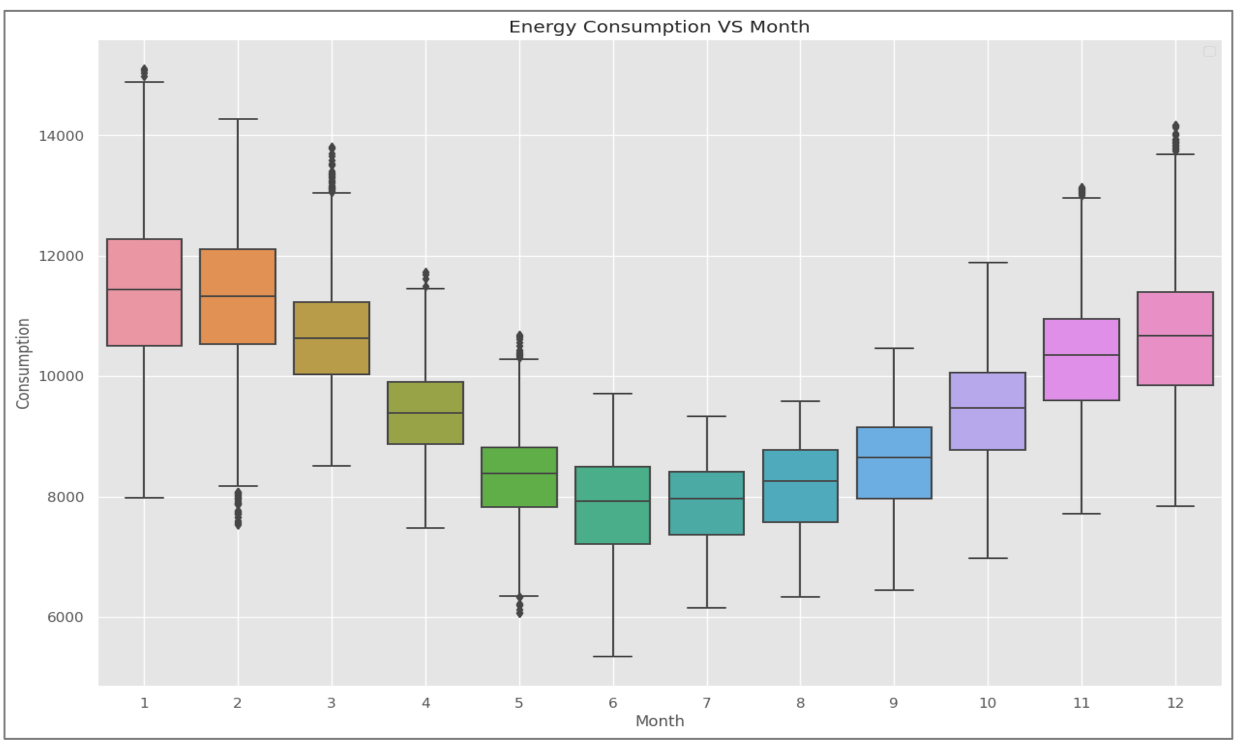


4.3 Visualizing the energy consumption of Finland using box-plot

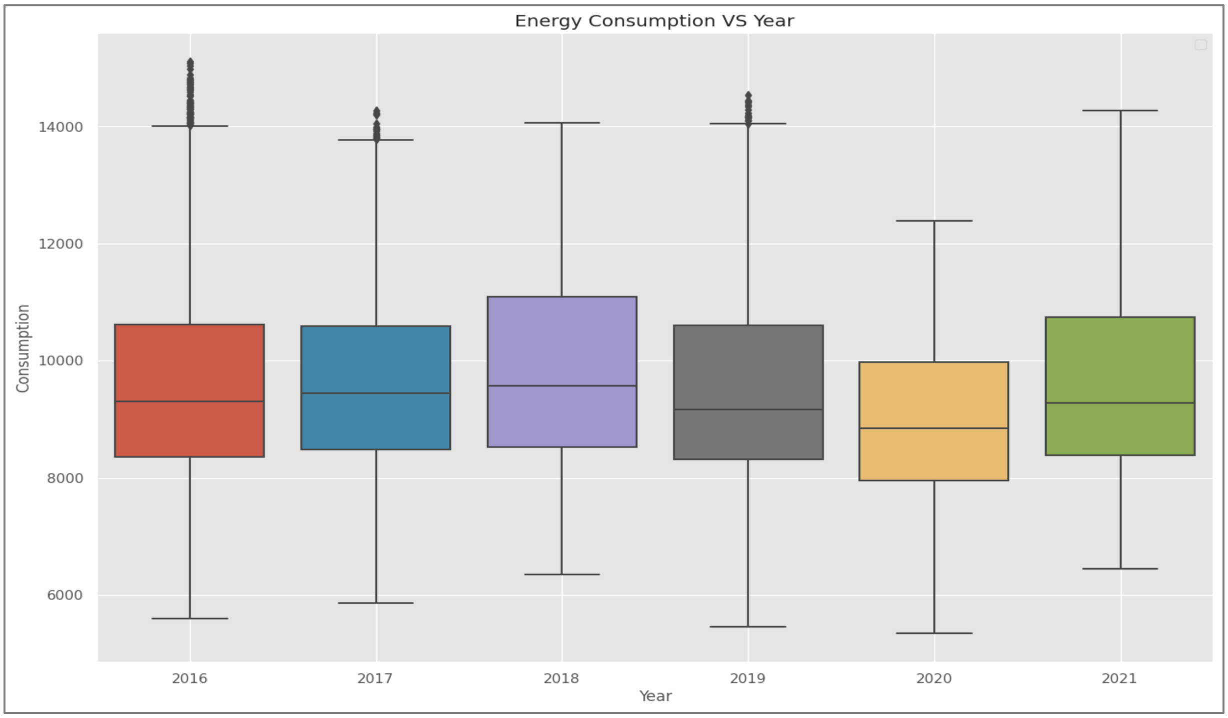
Energy Consumption Vs Hours



Energy Consumption Vs Month



Energy Consumption Vs Years



5 LSTM Model

5.1 Train, Validation, and Test Dataset

For predicting day consumption, data were down-sampled using resample function. This function changed the data from hourly frequency to daily frequency. This down-sampling was done by using the mean () method. The data rows were reduced from 52774 to 2184 rows

DAILY CONSUMPTION DATA

DateTime	Consumption	Month	Year	Week
2016-01-04	12300.625000	1.0	2016.0	1.0
2016-01-05	12945.375000	1.0	2016.0	1.0
2016-01-06	13192.750000	1.0	2016.0	1.0
2016-01-07	14243.541667	1.0	2016.0	1.0
2016-01-08	14121.666667	1.0	2016.0	1.0

The LSTM model is sensitive to the scale of the input data, so it can be a good practice to rescale the data to the range of 0-to-1. This is called data normalization. In this case, the data was normalized using the MinMaxScaler function. MinMaxScaler scaled the output and the input in the range between 0 - 1 to match the scale of the LSTM layer. MinMaxScaler will subtract the minimum value of Consumption and then divide it by the range.

To train the model we split the data into a training set (80%) and a testing set (20%). Also, we split training data into 20% for the validation set and 80% for training. The validation set is used to validate the model performance during training.

The input layer of the LSTM model requires 3D input, Where the first dimension represents the sample size, and the second represents the number of time steps. And the third dimension represents the number of features used to train the model [number of samples, time steps, and number of features]. The dataset was reshaped to look like a 3- dimensional array, where our time steps were 100 and our feature is the Consumption

RESHAPE INPUTS TO FIT LSTM LAYERS

```
X_train shape:  (1292, 100, 1)
X_test shape:  (335, 100, 1)
X_val shape:  (247, 100, 1)
```

5.2 Model Structure

Stacked LSTM will be built in this project, which contains four hidden LSTM layers one on top of another. The model building starts by creating a sequential model using Keras. Then we added the hidden LSTM layers to the model. Each layer had 50 units. A dropout layer was added to help in preventing overfitting. The last layer was the dense layer, which does the below operation on the input and returns the output

To update network weights iterative based on training data we used the Adam optimization algorithm which was used famously instead of the classical gradient descent procedure. Also, we used root mean squared error (RMSE) to test the model performance. A smaller RMSE means that our model is performing better.

LSTM MODEL SUMMARY

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 100, 50)           10400

 dropout (Dropout)           (None, 100, 50)           0

 lstm_1 (LSTM)               (None, 100, 50)           20200

 lstm_2 (LSTM)               (None, 100, 50)           20200

 lstm_3 (LSTM)               (None, 50)                20200

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 71,051
Trainable params: 71,051
Non-trainable params: 0
_________________________________________________________________
```

5.3 Model Training

Training the LSTM model was done using the training set and the validation dataset for testing the results through the training process. The learning algorithm worked through the entire training dataset 20 times (Epoch), and the model weights were updated after each batch where the batch size is 20

TRAINING LSTM MODEL

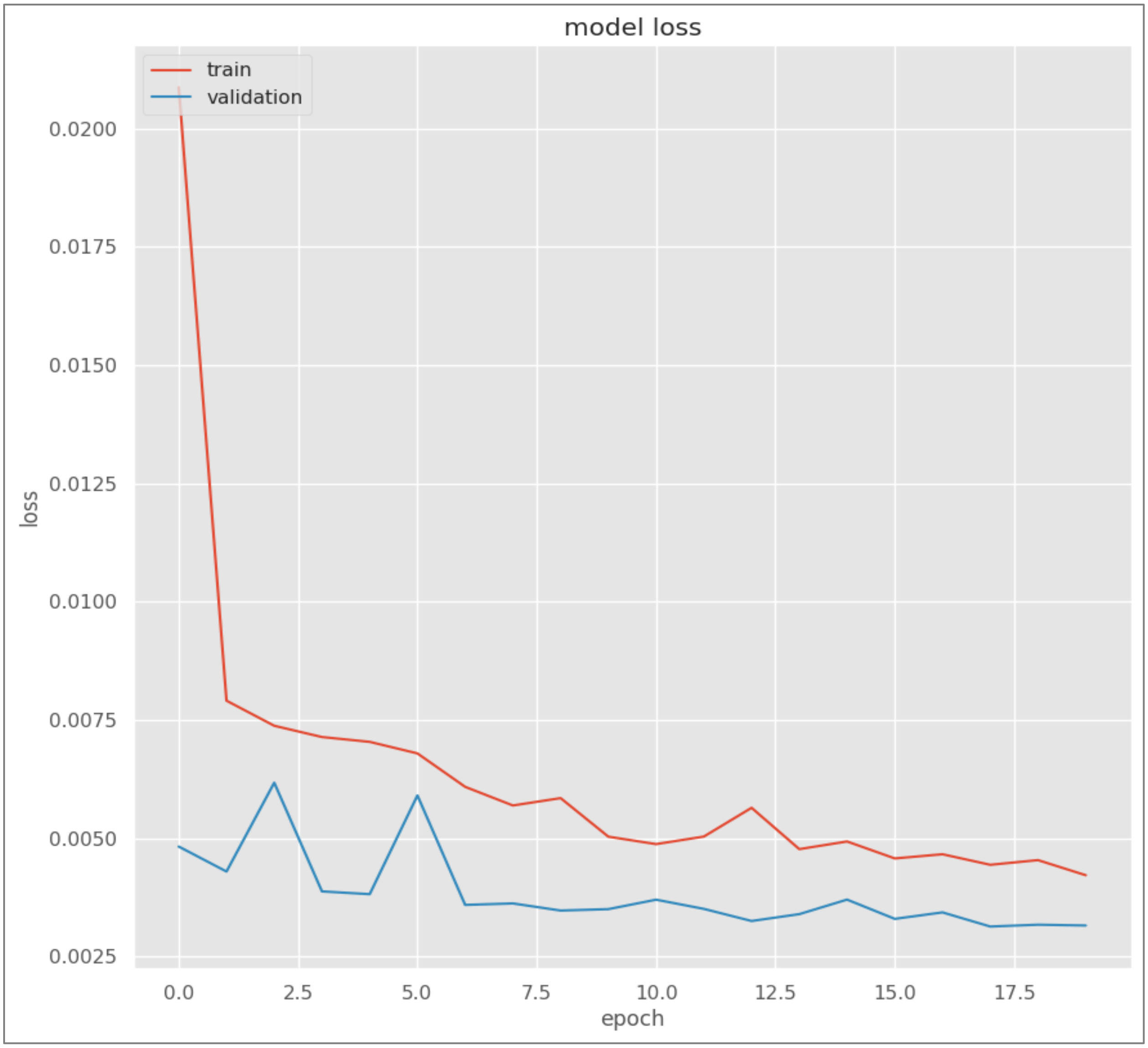
```
Epoch 1/20
64/64 [==============================] - 13s 206ms/step - loss: 0.0058 - val_loss: 0.0042
Epoch 2/20
64/64 [==============================] - 13s 207ms/step - loss: 0.0056 - val_loss: 0.0041
Epoch 3/20
64/64 [==============================] - 13s 204ms/step - loss: 0.0057 - val_loss: 0.0065
Epoch 4/20
64/64 [==============================] - 14s 215ms/step - loss: 0.0048 - val_loss: 0.0045
Epoch 5/20
64/64 [==============================] - 13s 207ms/step - loss: 0.0049 - val loss: 0.0052
Epoch 6/20
64/64 [==============================] - 13s 206ms/step - loss: 0.0046 - val_loss: 0.0040
Epoch 7/20
64/64 [==============================] - 13s 205ms/step - loss: 0.0046 - val_loss: 0.0038
Epoch 8/20
64/64 [==============================] - 13s 205ms/step - loss: 0.0050 - val loss: 0.0039
Epoch 9/20
64/64 [==============================] - 13s 204ms/step - loss: 0.0046 - val_loss: 0.0039
Epoch 10/20
64/64 [==============================] - 13s 207ms/step - loss: 0.0046 - val_loss: 0.0039
Epoch 11/20
64/64 [==============================] - 13s 207ms/step - loss: 0.0049 - val loss: 0.0041
Epoch 12/20
64/64 [==============================] - 13s 205ms/step - loss: 0.0050 - val_loss: 0.0043
Epoch 13/20
64/64 [==============================] - 13s 208ms/step - loss: 0.0044 - val_loss: 0.0045
Epoch 14/20
64/64 [==============================] - 13s 205ms/step - loss: 0.0044 - val_loss: 0.0061
Epoch 15/20
64/64 [==============================] - 13s 205ms/step - loss: 0.0050 - val_loss: 0.0036
Epoch 16/20
64/64 [==============================] - 13s 208ms/step - loss: 0.0045 - val_loss: 0.0041
Epoch 17/20
64/64 [==============================] - 13s 205ms/step - loss: 0.0043 - val_loss: 0.0036
Epoch 18/20
64/64 [==============================] - 14s 216ms/step - loss: 0.0044 - val_loss: 0.0034
Epoch 19/20
64/64 [==============================] - 14s 215ms/step - loss: 0.0040 - val_loss: 0.0037
Epoch 20/20

64/64 [==============================] - 13s 208ms/step - loss: 0.0040 - val_loss: 0.0038
```

6 Model evaluation

The check after training was to compare the training loss against the validation loss. The result shows that the two values were low, and no overfitting was detected.

Comparison Curve of Training and validation Accuracy of 20 epochs



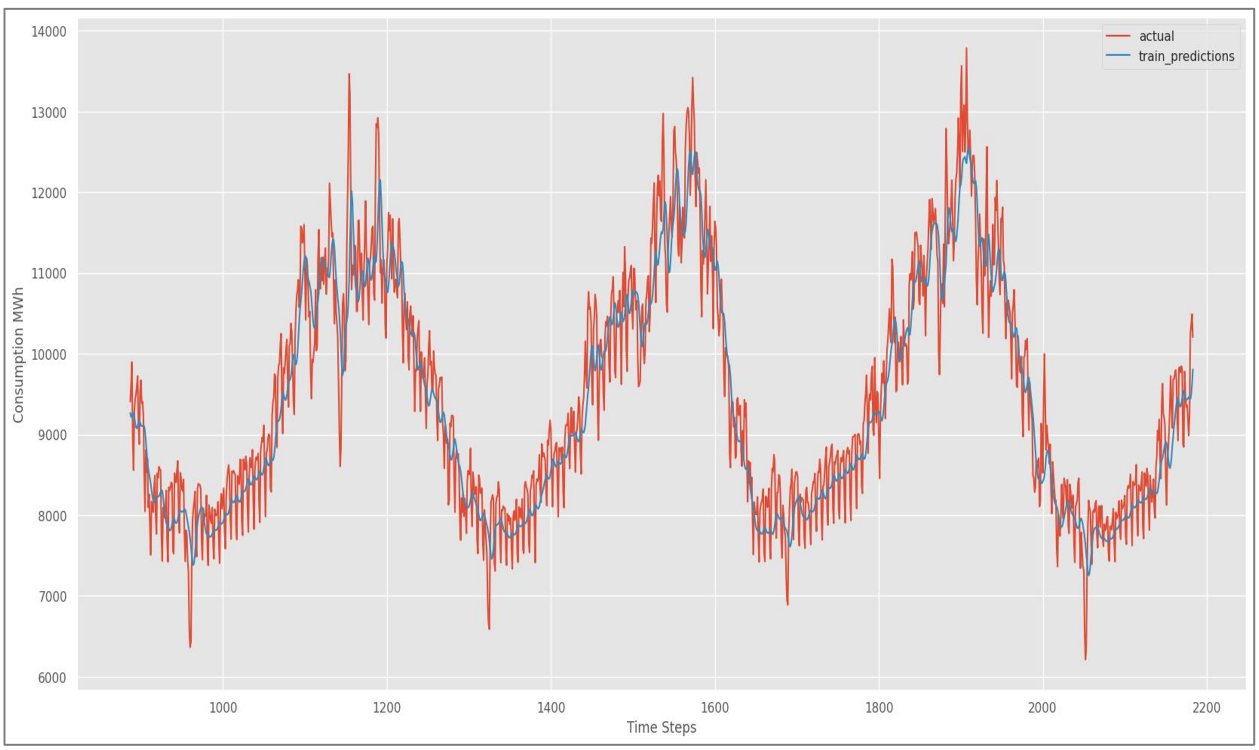
- ➔ `train_predict.shape  :  (1297, 1)`
- ➔ `test_predict.shape   :  (336, 1)`
- ➔ `val_predict.shape    :  (248, 1)`
- ➔ `train_predict[0]     :  [9495.356]`
- ➔ `y_train.shape        :  (1297,)`

6.1 Consumption predictions of the LSTM model were done using the training dataset.

```
41/41 [==============================] - 2s 53ms/step
```

	Train Predictions	Actuals
0	[9264.8671875]	[9406.708333333334]
1	[9219.64453125]	[9614.791666666666]
2	[9214.4677734375]	[9894.708333333334]
3	[9256.0810546875]	[8933.708333333334]
4	[9269.232421875]	[8557.208333333334]
...	...	...
1292	[9451.04296875]	[9259.666666666666]
1293	[9440.0625]	[10248.5]
1294	[9490.6181640625]	[10360.333333333334]
1295	[9618.0263671875]	[10489.833333333334]
1296	[9804.474609375]	[10204.5]

1297 rows × 2 columns

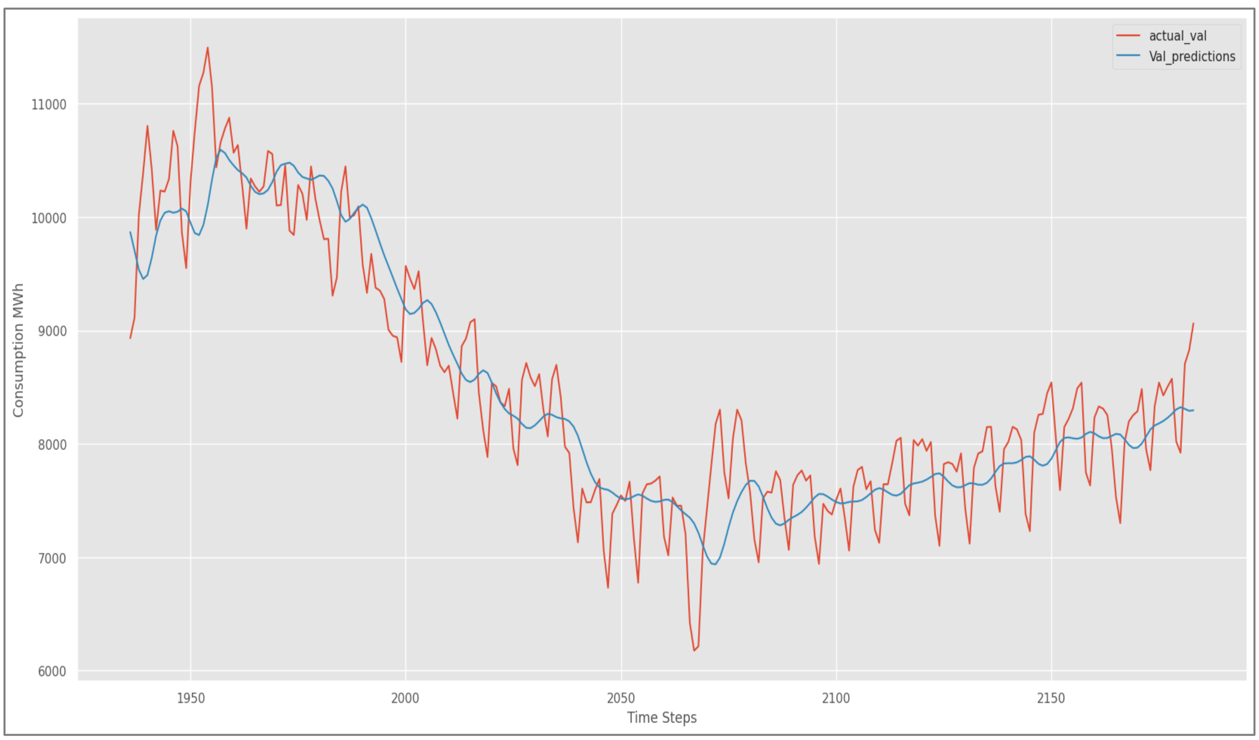


6.2 Consumption predictions of the LSTM model were done using the validation dataset.

```
8/8 [==============================] - 0s 50ms/step
```

	Val Predictions	Actuals_val
0	[9866.6240234375]	[8931.875]
1	[9703.4189453125]	[9112.916666666666]
2	[9535.7138671875]	[10019.416666666666]
3	[9453.4560546875]	[10390.375]
4	[9488.783203125]	[10802.916666666666]
...	...	...
243	[8303.322265625]	[8020.541666666667]
244	[8322.609375]	[7921.75]
245	[8307.9443359375]	[8705.041666666666]
246	[8290.7236328125]	[8824.708333333334]
247	[8295.7470703125]	[9062.375]

248 rows × 2 columns

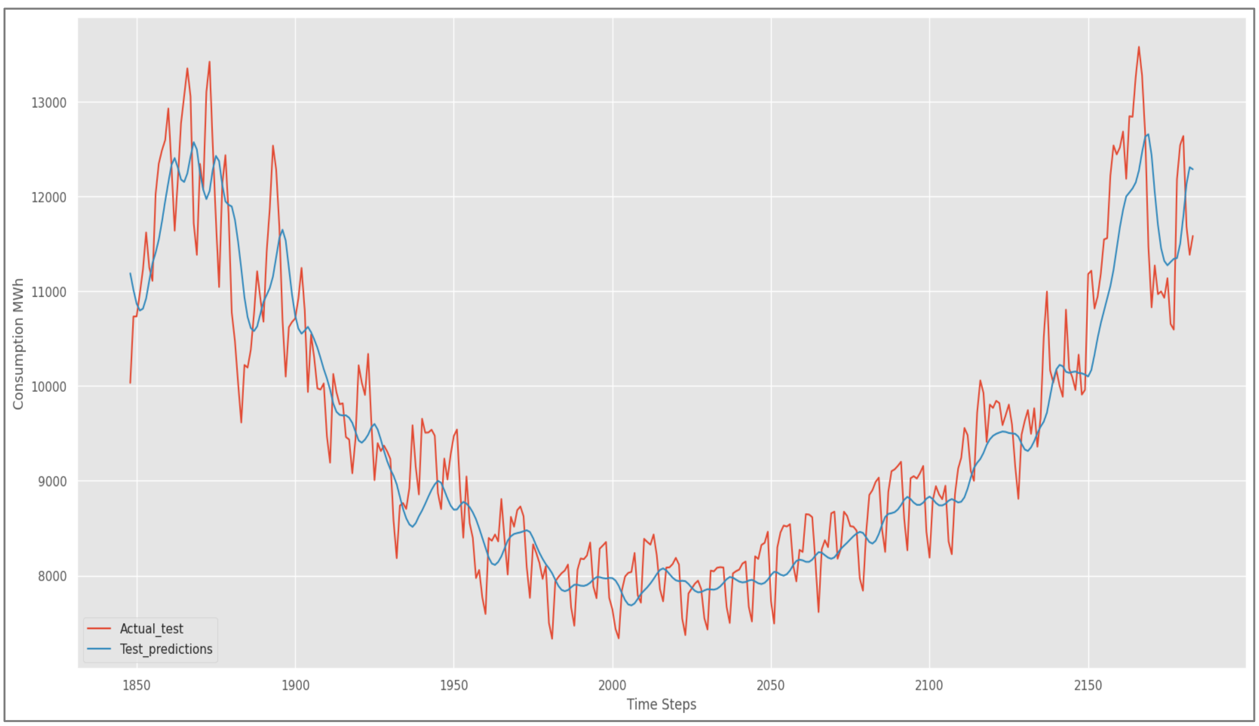


6.3 Consumption predictions of the LSTM model were done using the test datasets.

```
11/11 [==============================] - 1s 50ms/step
```

	test Predictions	Actuals_test
0	[11187.2333984375]	[10032.541666666666]
1	[11014.048828125]	[10732.125]
2	[10868.849609375]	[10733.583333333334]
3	[10795.6708984375]	[10971.875]
4	[10814.8095703125]	[11227.791666666666]
...	...	...
331	[11497.7705078125]	[12540.25]
332	[11787.5009765625]	[12635.958333333334]
333	[12127.8740234375]	[11684.333333333334]
334	[12306.8525390625]	[11384.166666666666]
335	[12285.5107421875]	[11581.625]

336 rows × 2 columns



Conclusion

The task of predicting electricity hourly consumption is a crucial challenge for power companies as it enables them to optimize their power generation and distribution processes, reduce their operating costs, and improve the overall reliability of the power grid. Through the use of data mining techniques, such as regression analysis, clustering, and feature selection, we can extract valuable insights from historical electricity consumption data to accurately forecast future electricity demand. With the increasing availability of data and advancements in data mining algorithms, the accuracy and reliability of electricity consumption prediction models continue to improve. By implementing these models, power companies can make informed decisions about their power generation and distribution processes, ultimately leading to a more efficient and sustainable energy system.

References

1. Allik, A., & Annuk, A. (2017, November). Interpolation of intra-hourly electricity consumption and production data. In 2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA) (pp. 131-136). IEEE

2. Miller, G. J., Novan, K., & Jenn, A. (2022). Hourly accounting of carbon emissions from electricity consumption. Environmental Research Letters, 17(4), 044073

3. de Chalendar, J. A., Taggart, J., & Benson, S. M. (2019). Tracking emissions in the US electricity system. Proceedings of the National Academy of Sciences, 116(51), 25497-25502

4. Andersen, F. M., Baldini, M., Hansen, L. G., & Jensen, C. L. (2017). Households' hourly electricity consumption and peak demand in Denmark. Applied energy, 208, 607-6

5. Edwards, R. E., New, J., & Parker, L. E. (2012). Predicting future hourly residential electrical consumption: A machine learning case study. Energy and

6. Noussan, M., & Neirotti, F. (2020). Cross-country comparison of hourly electricity mixes for EV charging profiles. Energies, 13(10), 2527

7. Gellert, A., Florea, A., Fiore, U., Palmieri, F., & Zanetti, P. (2019). A study on forecasting electricity production and consumption in smart cities and factories. International Journal of Information Management, 49,

8. Rodrigues, F., Cardeira, C., & Calado, J. M. F. (2014). The daily and hourly energy consumption and load forecasting using artificial neural network method: a case study using a set of 93 households in Portugal. Energy Procedia, 62, 220-22

9. Salian, I. (2018, August 2). SuperVize Me: What's the Difference Between Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning?

10. Salian, I. (2018, August 2). SuperVize Me: What's the Difference Between Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning?