**T.C.**

**BAHÇEŞEHİR UNIVERSITY**

**FACULTY OF ENGINEERING AND NATURAL SCIENCES**

**DEPARTMENT OF COMPUTER ENGINEERING**

**FINDING THE ENERGY DEFICIT IN THE SYSTEM**

**Capstone Project**

**Burkay Pehlivan**

**Advisor: Prof. Dr. Taşkın Koçak**

**İSTANBUL, January 2018**

**STUDENT DECLARATION**

By submitting this report, as partial fulfillment of the requirements of CMP4999, the student promises on penalty of failure of CMP4999 that

* they have given credit to and declared (by citation), any work that is not their own (e.g. parts of the report that is copied/pasted from the Internet, design or construction performed by another person, etc.);
* they have not receive unpermitted aid for the project design, construction, report or presentation;
* they have not falsely assigned credit for work to another student in the group, and not taken credit for work done by another student in the group.

# ABSTRACT

FINDING ENERGY DEFICIT IN THE SYSTEM

Burkay Pehlivan

Faculty of Engineering and Natural Sciences

Department of Computer Engineering

Advisor: Prof. Dr. Taşkın Koçak

January, 2018

This project is about creating a trading helper for energy market using neural network algorithms.

There is an energy trading market in Turkey which helps companies who own powerplants to reduce their imbalance penalty. The aim of the penalty is tho keep grid stable and minimalize the unexpected imbalance in the grid.

Purpose of the project is providing a forecast about deficit in grid using machine learning neural network algorirhms. It will be a guide to a broker who makes transactions in intraday market.

The data prepared from various sources but the most important one was transperancy platform of epiaş(enerji piyasaları işletme a.ş.) which company also the provider of the energy market system.There is a message system which shows powerplant failures and maintenance. This messages parsed and acquired hourly deficit in the grid. Added some other datas such as hour and weekday and wheather data etc.

For predicting, neural network tecniquesis being used and calculated its accuracy with the open-source “scikit-learn” library. The development was made in python 3.

**Key Words**: machine learning, neural networks, energy market

# TABLE OF CONTENTS

ABSTRACT iii

TABLE OF CONTENTS iv

LIST OF ABBREVIATIONS v

1. OVERVIEW 1

1.1. Description of the Project 1

1.2. Literature Review 2

1.3. Goals 2

2. TECHNICAL SOLUTIONS 3

2.1. Technical Limitations 3

2.2. Facilities and Components 3

2.3. Algorithm Design and Safety 3

2.4. Technical Problems 4

2.5. Product Specification and Verification 4

2.6. Results 4

3. WORK PLAN 4

3.1 Deliverables and Division of Tasks 4

3.2 Tasks and Time Line 5

3.3 Cost 5

4. CONCLUSION 7

ACKNOWLEDGEMENT 8

REFERENCES 9

APPENDIX A 10

# LIST OF ABBREVIATIONS

BPM Balancing Power Market

DAM Day Ahead Market

IDM Intraday Market

MCP Market Clearing Price

SMP System Marginal Price

EPİAŞ Enerji Piyasaları İşletme Anonim Şirketi

ML Machine Learning

MLP Multilayer Parceptron

# 1. OVERVIEW

This project aims to predict the energy deficit in the grid (for whole country) by using various kind of data about grid and predict the outcome via using machine learning neural network algorithms. These predictions will be helpful for the brokers who are making trades in the energy market.

## 1.1. Description of the Project

The key value the project aims to find is SMP (System Marginal Price).

Because of the regulations in the turkey, Power Plants which are connected to grid must give their predictions of the production quantity to Epiaş one day before the production. This process acquires the price which state will pay for 100 Kw/h electricity, which name is Market Clearing Price (MCP).

SMP acquires according to the deficit in the grid. The base value is that hour’s MCP price. If deficit exists in the grid, SMP moves higher values than MCP. If production is more than the consumption in the grid, SMP price becomes lower than the MCP price. The imported thing is, the exact value of SMP for each hour acquires five hours later. To explain it different way, we can learn current hour’s SMP price five hour later.

If a plant would not be able to reach the estimates which was given one day before, government punishes that plant according to the missed value of kilowatts. The penalty is, SMP value with some calculations for each kilowatt.

Aim of the project is to predict if SMP will be bigger or not than MCP two hours before consumption happened.

## 1.2. Literature Review

Machine learning applications are being used in many fields of computer and data science.The closest projects are predicting the stock prices with machine learning algorithms.

The brokers who makes trades in this market usually estimates the consumption by themselves. Even though there are companies who makes forecasts for production of powerplants, there is not much company which makes prediction about consumption side.

## 1.3. Goals

The goal is to predict if SMP will be bigger or not than MCP two hours later as accurate as possible. This is requiring require selection of the best algorithm and best data and getting ac much as accuracy.

# 2. TECHNICAL SOLUTIONS

## 2.1. Technical Limitations

Possible limitations are lack of data or useless data. The quality of data affects the prediction. Even though there is a transperancy platform hich is provided by EPİAŞ itself, it is really hard to find any wheather record without paying wheather forcest companies a lot of money and which is a key factor of consumption country-wide. The only data avaible about wheather record was in american government website which keeps record of airport wheathers world wide: NCEI (National Centers For Enviromental Information).

In the NCEI’s website, the wheather data is providing in daily basis but energy market is hourly-based. Due to lack of data about wheather records generally, the daily wheather data in NCEI transformed into hourly data, which is a good input for classification algorithm but not the best.

## 2.2. Facilities and Components

For construction the project, “Scikit Learn” library is used. Scikit Learn(Sklearn) is an open-source project for machine learning applications which is developed by software community. It is easy to develop an machne learning algorithm from scratch.

Sklearn is developed in python language. Python is also a fast and easy-to-use programming language. For these reason python 3.6 is used for building the model.

## 2.3. Algorithm Design and Safety

In the first step, data prepared in myql server for fitting the classifier. Tables created and required sql codes has written for prepration.After pprepration the desired imported in “.csv” format. (The reason behind is python modules are more succesful when reading data from csv file instead of fetching from database).

Then sklearn library imported. With the help of this library, it is possible to many things besides generating algorithm such as testing, scoring and arranging data for fitting the model.

In next step, the data should be prepared for fitting. The input consists of six different input and these are;

- Month

- Hour

- Maintenance (country-wide powerplants)

- Failure (country-wide powerplants)

- Weekday

- Temprature

Note: Temprature data calculated by collecting seven cities’ tempratures which shows the geographic characteristics of the geographic area.

And the desired output is;

- Position (either 1 or 0)

The input data should be scaled, this process helps the algorithm for better classification.

In the next step, the data must be splittedd in training and testing parts. For better learning algorithm, training data should be 20% of the whole rows and this data should be selected randomly. Sklearn has a method for this solution and it is used.

Next step is preparing the classifier object before fitting the data. The hidden layers set in [100 : 1000 : 100] format.

To verify that algorithm is learning correctly, the mean squared error should find during learning and it should be observed that the error is decreasing during backpropagation. The mean squared error calculated by Substracting the the squares of predicted outputs from correct output and taking avarage of it.

The risk of this algorithm is, by relying on the prediction of the algorithm it is possible to loss serious amount of money in intraday energy market. The users of this model should be aware that this algorithm is for guidance, there is no guarantee that the forecast is ablsolutely true. The responsibilty is on the broker who uses help of this algorithm.

## 2.4. Technical Problems

There was implementing problems due to the data type incompatiblities. These problems are overcomed.

## 2.5. Product Specification and Verification

After backpropagation, the fitted model tested by “score” method provided by sklearn library with the testing data. The result was in between 70% and 75%.

## 2.6. Results

According to the high accuracy score, the created model is acceptanble for the future possible users of the algorithm.

# 3. WORK PLAN AND COSTS

## 3.1 Deliverables and Division of Tasks

Due to the lack of technical information, before starting implementation the code, a pre-research was required.

Second step was the implementation part. The lack of technical information caused lot of problem but with the help of advisor Mr. Koçak these problems overcomed.

Last part, the model acquiring and report preperation.

## 3.2 Tasks and Time Line

1. Research about ML (first 3-4 week)
2. Research about useful data (between 1. And 8th week)
3. Determining most suitable algorithm (5th week)
4. Implementation (between 6th and 12th week)
5. Testing Data (between 9th and 12th week)
6. Preparing poster (between 12th and 14th week)

## 3.3 Cost

Thanks to the open-source community of both scikit-learn and python, there was no cost at all.

# 4. CONCLUSION

Even though lack of tachnical information, problems about finding and preparing data, inadequate time the project succeeded with the score around %70 truth. The design and implementing of this project was quite helpful on learning base information for diving into machine learning world.

# ACKNOWLEDGEMENT

Special thanks to Mr. Emre Demirel who contributed this project with his deep knowledge about energy market, The Energy Pool Turkey team for training me during my internship.

And I would like to thank Mr. Taşkın Koçak for sharing, guiding and support on this project even though inadequate time and my insufficent information about machine learning and neural networks.

# REFERENCES

<https://seffaflik.epias.com.tr/transparency/>

[https://pythonprogramming.net](https://pythonprogramming.net/)

<http://scikit-learn.org/>

http://www.noaa.gov

# APPENDIX A

*#importing necessary libraries*

**import** numpy **as** np

**from** numpy **import** genfromtxt

**from** sklearn.neural\_network **import** MLPClassifier

**from** sklearn **import** preprocessing, model\_selection

*# for mean squared error calculation*

**def** mse( y\_true, y\_pred):

y\_err = y\_true - y\_pred

y\_sqr = y\_err \* y\_err

y\_sum = np.sum(y\_sqr)

y\_mse = y\_sum / y\_sqr.size

**return** y\_mse

*#this is for showing real data*

np.set\_printoptions(suppress=**True**)

*# this part is for readind .csv files and turning them into numpy arrays*

data = genfromtxt(**'simple\_pos.csv'**, delimiter=**','**, skip\_header=1)

*#this part slices arrays into two parts*

*#labels*

*#month(0),hour(1),maintenance(2),failure(3),weekday(4),temprature(5),position(6)*

X = data[:, 0:6]

*#feature*

y = data[:, 6]

*#scaling the label data for effective clustering*

X = preprocessing.scale(X)

print(X.shape)

*#spliting data as training data and testing data*

X\_train, X\_test, y\_train, y\_test = model\_selection.train\_test\_split(X, y, test\_size=0.8)

*#preparing classification object*

mlp = MLPClassifier(random\_state=0, hidden\_layer\_sizes=[100, 1000, 100], max\_iter=1)

*# the part where training and observing error happens*

i = 1

cont = **True**

min\_error = 1

**while**(i <= 1000):

mlp.partial\_fit(X\_train, y\_train, np.unique(y\_train))

y\_pred = mlp.predict(X\_train)

mse\_err = mse(y\_train, y\_pred)

print(**"iteration: "** + str(i) + **" Mean Squared Error: "** + str(mse\_err))

i = i + 1

*#to see the accuracy of the prepared model*

accuracy = mlp.score(X\_test, y\_test)

print(accuracy)