KADİR HAS UNIVERSITY
SCHOOL OF GRADUATE STUDIES
PROGRAM OF MANAGEMENT INFORMATION SYSTEMS

# SHORT-TERM SOLAR IRRADIANCE FORECASTING WITH DEEP NEURAL NETWORKS

CANER VATANSEVER

MASTER'S THESIS

ISTANBUL, AUGUST, 2019

# SHORT-TERM SOLAR IRRADIANCE FORECASTING WITH DEEP NEURAL NETWORKS

CANER VATANSEVER

MASTER'S THESIS

Submitted to the School of Graduate Studies of Kadir Has University in partial fulfillment of the requirements for the degree of Master's in the Program of Management Information Systems

ISTANBUL, AUGUST, 2019

DECLARATION OF RESEARCH ETHICS /
METHODS OF DISSEMINATION

I, CANER VATANSEVER, hereby declare that;

- this master's thesis is my own original work and that due references have been appropriately provided on all supporting literature and resources;
- this master's thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- I have followed *Kadir Has University Academic Ethics Principles prepared in accordance with The Council of Higher Education's Ethical Conduct Principles.*

In addition, I understand that any false claim in respect of this work will result in disciplinary action in accordance with University regulations.

Furthermore, both printed and electronic copies of my work will be kept in Kadir Has Information Center under the following condition as indicated below:

☐ The full content of my thesis will be accessible from everywhere by all means. thesis will be automatically accessible from everywhere by all means.

CANER VATANSEVER

22/08/2019

KADİR HAS UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

# ACCEPTANCE AND APPROVAL

This work entitled SHORT-TERM SOLAR IRRADIANCE FORECASTING WITH DEEP NEURAL NETWORKS prepared by CANER VATANSEVER has been judged to be successful at the defense exam on 22.08.2019 and accepted by our jury as

APPROVED BY:

Dr. Öğr. Üyesi Oğuzhan Ceylan (Advisor)
Kadir Has Unversity .....

Dr. Öğr. Üyesi Kerem Altun
Işık University .....

Prof. Dr. Hasan Dağ
Kadir Has Unversity .....

I certify that the above signatures belong to the faculty members named above.

.....

Dean Prof. Dr. Sinem AKGÜL AÇIKMEŞE

Dean of Graduate School of Science and Engineering

DATE OF APPROVAL:

# TABLE OF CONTENTS

SHORT-TERM SOLAR IRRADIANCE FORECASTING WITH DEEP NEURAL NETWORKS

# ABSTRACT

Usage of solar energy has increased through the last decades, and they are being integrated into main grid systems since the recent years. In order to fully benefit from solar panels, predicting irradiance is essential. By knowing 15-minute ahead values of solar irradiance, resistance of the cells inside the solar panels can be measured to analyze production output. This study focuses on 15-minute ahead forecasting of irradiance by using sliding windows method on the feature set. ANN, K-NN, SVM and RF models are optimized in this study. As the result of the study, around 6% MAPE is achieved.

**Keywords: Solar Irradiance Forecasting, Artificial Neural Networks, Random Forest, K-Nearest Neighbor, Short Term Forecasting**

DERİN SİNİR AĞLARI KULLANIMIYLA KISA SÜRELİ GÜNEŞ IŞIMASI TAHMİNLEMESİ

# ÖZET

Güneş enerjisinin kullanımı son 10 yıl içerisinde artş göstermektedir. Ek olarak bu kullanım, son yıllarda, şebeke sistemleri ile entegre edilmeye başlanmştır. Güneş panellerinden tamamıyla yararlanabilmek için, ışımayı tahmin edebilmek çok önemlidir. 15 dakika sonrasındaki güneş ışıması değerlerini bilerek, güneş paneli içerisindeki direnci tahmin edebilir ve üretimi analiz edebiliriz. Bu çalışma sürgülü pencere yöntemini kullanarak 15 dakika sonrasındaki ışıma tahminlemesine odaklanmıştır. Yapay sinir ağları, k-en yakın komşu ve rassal orman modelleri bu çalışmada optimize edilmiştir. Bu çalışmanın sonucunda, yaklaşık olarak 6% mutlak yüzde hataya ulaşılmıştır

**Anahtar Sözcükler: Güneş Işıma Tahminlemesi, Yapay Sinir Ağları, Rassal Orman, K-En Yakın Komşu, Kısa Dönemli Tahminleme**

# ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Professor Hasan Dağ and Dr. Oğuzhan Ceylan, my research supervisors, for their patient guidance, enthusiastic encouragement and useful critiques of this research work. I would also like to thank Dr. Oğuzhan Ceylan, for her advice and assistance in keeping my progress on schedule. My grateful thanks are also extended to Mr. Yunus Kologlu for his help in doing the raw data analysis.

Nobody has been more important to me in the pursuit of this project than the members of my family. I would like to thank my parents, whose love and guidance are with me in whatever I pursue. They are the ultimate role models. Most importantly, I wish to thank my loving and supportive wife, Semanur, and my little wonderful daughter, Yaren, who provide unending inspiration.

To my little family

# LIST OF TABLES

# LIST OF SYMBOLS/ABBREVIATIONS

| | |
|---|---|
| ANN | Artificial Neural Network |
| AR | Autoregressive |
| ARIMA | Autoregressive Integrated Moving Average |
| ARMA | Autoregressive Moving Average |
| DT | Decision Tree |
| ELU | Exponential Linear Unit |
| XGBR | Extreme Gradient Boosted Regression |
| FASE | Forecast Aided State Estimator |
| GBR | Gradient Boosted Regression |
| IRLS | Iteratively Reweighted Least Squares |
| MAD | Mean Absolute Deviation |
| MAE | Mean Absolute Error |
| MSE | Mean Squared Error |
| MAPE | Mean Absolute Percentage Error |
| MRMR | Minimum Redundancy Maximum Relevance |
| MARS | Multilinear Adaptive Regression Splines |
| MIDC | Measurement and Instrumentation Data Center |
| MLP | Multi Layer Perceptron |
| NWS | National Weather Service |
| PV | Photovoltaic |
| RF | Random Forest |
| RSM | Response Surface Methodology |
| ReLU | Rectified Linear Unit |
| RMS | Root Mean Square |
| SVM | Support Vector Machine |

# 1. INTRODUCTION

By the increase in the usage of renewable energies, these systems became more integrated into the grid systems. Solar energy is the most abundant renewable energy exists in the world (Szuromi et al, 2007). There are two ways to convert solar energy into electricity. Solar Thermal Power Plants (STPP) convert the direct normal solar irradiance to electricity by using the heat of beams whereas Photovoltaic plants directly convert energy into the electricity (Lara-Fanego et al. 2012). Worldwide photovoltaic production increases continuously and it is estimated by International Energy Agency (IEA) that 2% of the electricity demand of the world would be satisfied by solar production by 2030 (IEA, 2006a). Thus it became a necessity predict output of these systems as they became integrated into the large scale power systems (Espinar et al, 2010). As energy production from other conventional sources can easily be calculated, due to the high variability in the weather conditions, it is difficult to predict precisely. It is necessary to predict output for following days and hours in order to accomplish a successful integration into the power systems.

Due to the increase in the integration and application of solar energy, requirement for solar data has increased dramatically in the recent years (Mellit Pavan, 2010). Solar irradiance is forecasted within the usage of different approaches; including machine learning models, meta-heuristic algorithms and empirical relations. Nevertheless, three factors are taken into the account during the selection of optimal model for irradiance prediction.

Firstly, data resolution and forecasting horizon play a vital role in the model selection. Data resolution refers to time intervals between the successive data points which are used in the model. Depending on the data resolution, forecasting horizon

is determined accordingly. There are mainly three forecasting horizons which can be used for solar irradiance prediction. Long-term forecasts are used mainly for the deciding the installations of the solar plants. By measuring the solar radiation over the long term, it is possible to increase utilization of solar plants by deciding the best location. Long term forecasts cover the predictions that are 1-year ahead or more. Data resolution belonging to the long-term forecasts have monthly or yearly intervals in general. Medium-term forecasts are used mainly for arranging the deals between energy companies, institutions and customers. By determining the approximate output gained from the solar plants, electricity retailing companies can decide the amount and price in the electricity supply and demand prices within free consumers and solar plant owners. Data resolution belonging to the medium- term forecasts have weekly or monthly intervals in general. Short-term forecasts are used mainly for supporting decision making processes during the planning of electricity production. By determining the amount of electricity gained from the solar plants and other renewable energy sources, production amount from fossil fuels (natural gas, coal, oil) can be adjusted. This will help within the over and underproduction situations. Overproduction of electricity may result in the waste of resources whereas underproduction of electricity may potentially cause city or countrywide electricity blackouts. Alternatively, very short- term forecasts can be used for momentarily adjustments in the solar panels. Angle of the solar panel and resistance of solar panels can be determined using the very short-term forecast data. Data resolution belonging to short-term forecasts include 1-minute, 15-minute, 45-minute and 1-hour.

Secondly, availability of atmospheric variables plays a vital role in the determination of optimal model for solar radiance forecasting. Basically, Global Horizontal Irradiance (GHI) amount determines the output of solar plants (Remund, Perez  Lorenz, 2008). GHI is the total amount of radiation which is obtained from above through to the surface that is horizontal to the ground. GHI is determined based on the equation 1 below. Diffuse Horizontal Irradiance (DHI) in the formula refers to the amount of radiation that is received from a surface that arrives through the scat-

tered molecules and particles in the atmosphere, rather than the directly arriving radiation from the sun. In other words, it is the radiation which arrives from the blue sky and clouds. Direct Normal Irradiance (DNI) in the equation refers for the quantity of solar radiation that received from surface by an angle. Maximum solar radiation is achieved when the surface is perpendicular to the incoming radiation. Since the sun is continuously shifting during the day and solar beams refract when they enter into the atmosphere, determining the angle of solar panels is another possible adjustment to fully utilize the solar panels. In that case, cosine of the angle between the surface and beams are used in the equation in order to calculate the amount of DHI.

$$GHI = DHI + DNI * cos(z) \tag{1.1}$$

There are various factors that affect the DHI. Solar elevation, which is the distance between the sun and horizon directly affects the amount of DHI (Reno, Hansen Stein, 2012). Maximum irradiance potentially occurs when the sun is at the most perpendicular position. Cloudiness is another factor that greatly affects the amount DHI because of the fact that clouds may partially block sun beams from reaching to the surface of earth. Weather condition on wind on the other hand, affect the power output of the solar panels. Because of the fact that utilization of the solar panels is directly affected from heat, wind and rain factors are included in the solar radiation forecast models. They are also indicators of the cloudiness in one sense. Alternatively, there are several other factors that are considered in order to choose the optimal model for solar radiance forecasting. Climate of the forecasting region plays a vital factor because of the fact that the significance of the atmospheric variables may change depending on the region. Also, additional variables are taken into the account in specific climates. For instance, sand variable is used in the forecasting models which are built for desert regions.

Thirdly, accuracy of the models plays a vital role in the selection of solar radiation

forecasting. Generally speaking, the frequently used factors are Root Mean Square Error (RMSE), Mean Bias Error (MBE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). The used accuracy metric affects the applicability of the models. The solar radiance amounts during the day start from low, increase in the following hours and decrease afternoon in general. For that reason, difference of amount of the solar radiation between different hours of the day may be large. Selection of RMSE as the accuracy metric in the optimization models would cause in the avoidance of high forecasting errors in where radiation is an extreme outlier within the day. Selection of MAE would cause in treating all of the data points equally, which generates an approximate forecasting error for any interval. Selection of MAPE aims for percentage error, which results in giving importance primarily for the intervals where the interval is lowest. Nevertheless, selection of MAPE causes MAE for the interval where the solar radiation is highest. For that reason, models should be optimized and compared in terms of different metrics in order to gain a better insight. For most of the cases, MAE is the most frequently used metric for the comparison. However, when models which are built for different regions are compared, accuracy metrics may lead into the faulty decisions, as the amount of solar radiation differs between different regions and climates. Additionally, seasons have a drastic effect on the models. In literature, there are generally specific models which focus on seasonal forecasts, such as summer and winter. As irradiance duration during the winter months are shorter and beam angles are lower, the amount of solar radiation is lower.

In the literature, Artificial Neural Network (ANN), K-Nearest Neighbor (K-NN) Support Vector Machines (SVM) and Random Forest (RF) models are frequently used for solar radiation forecasting tasks because of several reasons. Especially in short-term forecasting tasks, thousands of data tuples which include GHI amount and features exist and simple models are not capable of handling this data. There is also no linear relation between the features and the output, and determining the nonlinear relations require advanced models. Hence, Deep Neural Networks (DNN) are frequently used for this purpose. In DNN cases, there are different

factors that are taken into the account. Universal approximation theorem states that neural networks with a single hidden layer and a nonlinear activation function can approximate any continuous function with zero error; meaning that any optimization problem can be solved within the neural networks. However, there are several factors which must be taken into the account in ANN models. First of all, settings of hyper parameters directly affect the performance of the model; both in terms of accuracy and algorithm efficiency. Loss function type, number of hidden layers and nodes, optimization function and batch size are the hyper parameters that are included in the neural networks. In SVM and RF models, there are less types of hyper parameters which make it easier for these models to find the best accuracy.

There are theoretically infinite number of settings that can be applied in order to reach the best results. Nevertheless, it is not achievable in practice (Yager Kreinovich, 2003). For that purpose, hyper parameter optimization techniques are used in order to reach the best model. Grid search, random search and metaheuristic algorithms are frequently used for hyper parameter tuning. Grid search method is basically a brute force approach which tries all the possible settings in a grid of hyper parameters and finds the best model according to the defined accuracy metric. Although it guarantees finding the best model in the given parameter choices, it is computationally inefficient and also it limits user between a discrete set of hyper parameters. Random search on the other hand, is a faster version of grid search which only considers the random combination of given parameters in the defined iterations. It generally results in a worse models compared to the grid search. Nevertheless, it is computationally more efficient than the grid search method. Alternatively, metaheuristic algorithms (Bat Algorithm, Firefly Algorithm, Particle Swarm Optimization) can be used in order to optimize the hyper parameters. Metaheuristic algorithms can find a superior point compared to the initial point in a given subspace, without providing the optimal solution. Nevertheless, hyper parameter optimization is still a challenge to overcome in ANN models.

Purpose of this thesis is to develop and compare the forecasting models for GHI.

Mainly, machine learning models will be used for this purpose. Dataset belonging to the Oak Ridge National Lab, belonging to the May, June, July and August months of 2016, 2017, 2018 and 2019 will be used in this study. As for machine learning models, DNN, K-NN, SVM and RF will be used for that purpose. There are two main contributions which will be provided within this methodology. First contribution is the forecasting horizon from the given data resolution. Belonging data to the Oak Ridge National Lab has 1-minute data intervals and from this data resolution, predictions for 15 minutes ahead will be made, without knowing any of the data for the next 15 minutes. Second contribution is about the features that will be used in this study. Wind direction is integrated into the feature set and sliding windows method is applied in all of the features.

# 2. MACHINE LEARNING METHODS

## 2.1 Artificial Neural Networks

Artificial neural network is a methodology that is developed by inspiring from biological neural networks which human brains have. The reason of this inspiration is because brain is an incredible information processor. People focused on brains learning or processing structure in order to develop similar structures or algorithms for computers. Solely, human brains are very different from computers. Man (1982) discussed three levels of understanding an information processing; computational theory, representation and algorithm and hardware implementation. According to Gnen and Alpaydn (2011), people are at level of algorithm and representation level in understanding the brain and neural networks. There are many applications of neural networks such as picture and speech recognition, signature verification, and forecasting.

Briefly, the aim of the methodology is to work many machine learning algorithms together with data inputs to produce outputs. To understand how neural network is processing the data, some of the basic elements need to be defined such as perceptron, multilayer perceptrons, back propagation algorithm, and gradient descent algorithm.

### 2.1.1 Perceptron

In 1950s, Rosenblatt developed perceptrons. A perceptron may have inputs and its outputs may be inputs for other perceptrons. Basically, a perceptron takes inputs and produces outputs. For each connection, a weight w is defined which is called

connection weight. The z is defined as weighted summation of the inputs. The output y is defined as the function value of z, where $[w = [w_0, w_1, w_2, ...w_n]^T$ vector of weights and $X = [1, x_1, x_2...x_n]$ vector of inputs.

$$z = W^T X \tag{2.1}$$

For each node, an activation function G(z) is defined. The output of the function will give activation value (a) of that node.

$$G(z) = a \tag{2.2}$$

### 2.1.2 Multilayer perceptron

A single perceptron cannot be used for nonlinear functions of inputs. As a result of that, multilayer perceptron methodology is applied in such cases. This method includes hidden layer(s) between input and output layers. For each node, activation value and activation function is defined similar equations defined above. $X_i$ is taken to input layer and weighted summation (z) is calculated. Later the activation function to weighted summation is calculated in order to find the activation value a. For output layer, the activation value "a" is equal to y, output.

### 2.1.3 Back propagation algorithm

Back propagation algorithm was developed in 1970s but its importance became noticeable in 1986 by Rumelhart et al. It was an important development in terms of learning in neural networks. Basically, the aim of this algorithm is to see which weights are more significant on the error and assign new values to those weights using gradient descent algorithm. Thus, the error can be decreased.

$$\sigma E/\sigma W h_j = (\sigma E/\sigma Y_i) * (\sigma Y_i/\sigma Z_h) * (\sigma Z_h/\sigma W h_j) \tag{2.3}$$

If a nonlinear function with a single output is taken into account, Output y is

calculated as:

$$Y^t = \sum_h V_h * Z_h^t + V_0 \tag{2.4}$$

The error function is as follows:

$$E(W, v|X) = 1/2 \sum_t (r^t - y^t)^2 \tag{2.5}$$

In the given equation, $r_t$ stands for real values. The error function depends on the problem type. If the analysis is regression, the least-squared is used. If the analysis is classification, then the cost function of logistic regression is valid. In order to update hidden layer weights, least-squared rule is used.

$$\nabla V_h = n \sum_t (r^t - y^t) * Z_h^t \tag{2.6}$$

The terms $(r^t - y^t)$ play the role of error for , the hidden unit. The error in the output is $(r^t - y^t)$. Additionally, $Z_h^t * (1 - Z_h^t)$ is the the derivative of activation function in the equation. $X_j^t$ is the derivative of $Z_h$. In order to obtain $\nabla W_h j$ $V_h$ , we need to use . At the beginning of the algorithm, the weights are given as random. Later, using this algorithm the weights effect is calculated. Then, using gradient descent the new weights will be assigned.

### 2.1.4 Gradient descent

Gradient descent is an iterative algorithm to minimize a function by moving in a direction that is negative of the gradient. This algorithm is used in machine learning to adjust parameters. $\alpha$ is learning rate and $W_i$ is a neighbor point on function.

$$W_(i + 1) = W_i - a * \nabla F(W_i) \tag{2.7}$$

If F(x) is convex, the convergence to global maximum is guaranteed. For artificial neural networks, given a learning rate of *alpha*, $W_i^+$ stands for new value of weight w $W_i$ stands for existing weight$\nabla Wi$ stands for the effect of $W_i$ on cost function. Thus, the weights can be assigned and cost function can be improved.

### 2.1.5 Neural network hyper parameters

Optimizers shape and mold neural networks into its most accurate possible form by updating the model in response to the output of the loss function. The type of optimizers used that are generally used in researches are Stochastic Gradient Descent, RMSprop, Adam and Adadelta.

Adam is an adaptive learning rate method which means, it computes individual learning rates for different parameters. Adam optimizer uses estimation of first and second moments of gradient to adapt the learning rate for each weight of the neural network. The method is computationally efficient. It has little memory requirements and is well suited for problems that are large in terms of data and parameters.

Adadelta is an optimization method that uses the magnitude of recent gradient and steps to obtain an adaptive step rate. An exponential moving average over the gradients and steps is kept; a scale of the learning rate is then obtained.

RMSprop uses a moving average of squared gradients to normalized the gradient itself. It balances the step size. It decreases the step for large gradient to avoid exploding and increase the step for small gradient to avoid vanishing.

In the working process of neural networks, each layer feeds the next one with its outputs and those outputs become next layers inputs. What makes an input into output in a node is the activation function on that node. Basically, when a signal comes to a node it has its X value. This X value gets into the activation function and takes an output value. If activation functions were not in use in a neural network, inputs still vary when they get multiplied with weights. Yet the problem in this

situation is that the model would be linear. Neural networks are capable of handling complex models because with the activation functions they can achieve non-linear properties in the model. Non- linearity is important because it achieves better fitting in big complex datasets. One can create his/her own activation function yet there are some famous ones that can achieve the most out of the models. They are linear, ReLu, SeLu and ELu.

Linear activation function is a simple function that is simply . It is a line function where activation is proportional to input with c. Since c is constant, in back propagation the changes are constant. This way, changes are not dependent to . Linear activation function is not considered good in general.

Relu is the shortened version of rectified linear unit. It is the most used most general activation function in deep learning. Function of Relu is stated below.

$$R(x) = max(0, x) \tag{2.8}$$

Relu captures interactions and non-linearities very well. When more than one signal comes to the node with ReLu activation function, the activation of node is dependent to all coming signals, their inputs and weights. Also having non-constant slope, ReLu achieves non- linearity. Having ReLu in more than one layer, capturing non-linear fit gets better. ReLu avoids vanishing gradient.

Elu is exponential linear unit (Shat et al., 2016), it is so similar to ReLu. Elu also avoids vanishing gradient. The difference of ReLu and elu is that elu has negative values. Negative values push mean activations closer to zero. Zero means makes learning faster and it works like a regularization method. They bring gradient closer to natural gradient and prevent overfitting.

There are various pros and cons of each of these activation functions. Linear activation function reflects the range of activations, and linear relations can be represented

better this way. However, it is not capable of handling non-linear relations. Additionally, derivative of linear activation function is constant, which implies that there is no relation between the gradient and the values. Relu is capable of handling vanishing gradient problem and it is computationally efficient. Nevertheless, it can only be used inside the hidden layers rather than output layer. Additionally, since the range is between 0 and infinity, exploding gradient may possibly occur within this activation. Elu shares the similar properties with Relu. Additionally, negative values are also included in Elu.

A loss function is an objective function that machine learning models try to max or min through its learning steps. The output of a loss function measures the accuracy of models. There are several loss function options for the neural networks. Mean squared error is the sum of squared distances of data points to the regression line. Squaring has two goals, one to remove negativity and another is to increase the impact of distance. Mean absolute error is the average of sum of distances of data points to regression line. Mean absolute percentage error is the most used loss function in regression problems. It measures the prediction accuracy of forecasting methods.

## 2.2 K-Nearest Neighborhood Algorithm

The aim of the nearest neighbor algorithms is to determine the point in a dataset which is closest to a given query point (Beyer et al., 1999). It is mostly used in Geographical Information Systems as in these systems, points are associated with several geographical location and the closest city to a given point can be found by nearest neighbor algorithms. These algorithms request no preprocessing of the labeled sample set prior to their use. The crisp nearest-neighbor classification rule assigns an input sample vector, which is of unknown classification, to the class of its nearest neighbor.

By extending this idea, K-nearest neighbor algorithm can be constructed where

the aim is to find the class for vector by finding the most common class in K neighbors. Nevertheless, in a classification problem, there may be a tie if the number K is an even number. For binary classification problems, it can be prevented by choosing K as an odd number. The figure below shows the simple representation of K-NN algorithm. K-NN isnt a parametric method, which does not make any assumptions about the distribution of data. By choosing different distance metrics (Supremum, Manhattan, Euclidean, Minkowski), different results can be concluded. By considering $x_i$ as the x coordinate of the th point and as the y coordinate of the th point, Manhattan distance is found by the equation $Distance = |x_1 - x_2| + |y_1 - y_2|$. Euclidean distance can be found by the equation $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. Minkowski distance is the general name of these distance metrics and by changing the order of square root terms, different metrics can be found.

### 2.3 Decision Tree

The Decision Tree classifier is one of the possible approaches in multistage decision making, which involves breaking a complex decision into smaller and simple decisions to arrive at a final solution (Safavian  Landgrebe, 1991). A decision tree includes a root node which is formed from all of the data, a set of splits which are generated by smaller problems, and leaf nodes which represent the classes. The class of a new example can be identified by following the splits starting from the root and reaching to a leaf node. Figure below shows the scheme of a decision tree. Compared to the other supervised classification methods in the literature, decision tree has several benefits as it can handle both numeric and categorical inputs, allow missing values and handle nonlinear relations between features and classes. Additionally, decision trees significant intuitive appeal as the built classification framework is explicit and can be interpreted easily.

Decision trees are built within the consideration of entropy and information gain. Entropy controls the splitting scheme of the data, and effects the boundaries of a decision tree. Information gain must be calculated for each attribute in order

to execute splitting process in a decision tree. In a problem with classes of the target attribute, information gain for each attribute can be calculated by the formula $E(s) = \sum_i -p_i log_2 p_i$. $P_i$ refers to the number of occurrences of class I divided by the total number of instances.

There are several algorithms that are used in the decision tree method. ID3 algorithm, developed by Quinlan (1986) uses a top-down approach and deploys a greed search through the space of possible branches in the decision tree without the property of backtracking. Entropy and Information Gain are used in ID3 algorithm to build a decision tree. CART (Classification and Regression Trees) is another method to build decision trees. Decision trees built with CART are binary trees. Additionally, CART uses Gini Index metric instead of entropy and information gain for building decision tree. Equation Gini=$1 - \sum_i p_i^2$ shows the formula for Gini Index. In TT the perfect classification case, Gini Index is equal to zero.

Additionally, C4.5 is a popular decision tree algorithm which is an improved version of ID3 algorithm. Differently from ID3, it can also handle numerical values, missing values and it is suitable for error based pruning operation.

### 2.4 Ensemble Methods

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (such as weighted or unweighted average) in order to classify the new coming examples. Building good ensemble methods is an active research area in machine learning field, and empirical researches support the argument that ensembles are often more accurate when they are compared to individual classifiers which are part of the ensemble models.

In order to increase the accuracy of an ensemble model, the individual models which make the ensemble must be accurate and diverse, meaning that their structure should represent individual errors. The most famous ensemble methods are bagging, boosting, and random forest. Bagging (Bootstrap Aggregating) aims for generating

subsets from the original dataset randomly, training them and taking the average of their outputs. Boosting refers to the method which combines weak predictors with rules of thumb and forms a strong predictor. Lastly, random forest is a popular ensemble method which combines multiple decision tree models and finds the output (Breiman, 2001).

## 2.5 Feature Selection and Extraction Methods

### 2.5.1 Principal component analysis

Principal Component Analysis (PCA) is among the oldest and the most widely dimensionality reduction methods. Basically, model aims for finding variables that reduce the dimensionality while keeping the explained variance as high as possible. Let us consider that we have a dataset with p features and n samples. Let $X_j = X_1, X_2, ...X_n$ be the feature vector of the sample j. Finding a linear combination of the features that provide the maximum variance in data is aimed. These linear combinations are denoted as $\sum_i (a_j x_j = X a_i)$ , where a is considered as the vector of constants $a_j = a_1, a_2, ...a_n)$. This linear combinations variance can be found by the equation $var(X_a) = a' S_a$, as S is the sample covariance matrix and the transpose is denoted by . This problem is bounded by working on the unit vectors, and the limitation is provided by the equation $a'a = 1$. This problem is also the same problem as maximizing $a' S_a - \lambda(a'a - 1)$ , and $\lambda$ is considered to be the Lagrange multiplier in this equation. By taking derivative subject to a and equation to the null vector, we obtain the equation $Sa - \lambda a = 0 \iff Sa = \lambda a$. We are interested in largest $\lambda$ values because of the fact that eigenvalues are identify the variances of linear combinations. Returning to the original equation,$Xa_i$ are the principal components we aimed to find.

### 2.5.2 Factor analysis

Factor analysis is a method of reducing the dataset into a smaller size, while finding hidden patterns in the dataset while showing whether they overlap or not (Harman, 1960). Briefly, we can identify a factor as a set of observed variables which have the similar response patterns. The variables in these sets are bounded to each other by unknown noise variables.

Generally, two types of factor analysis are used; respectively exploratory and confirmatory factor analysis. Confirmatory Factor Analysis (CFA) aims for figuring the relationship between the set of observer variables and underlying the construction beneath them. On the other hand, Exploratory Factor Analysis (EFA) aims for underlying structure of large set of variables to a smaller set of summary variables (Brown, 2014).

### 2.5.3 Backward elimination

Backward elimination is a common sub feature set selection method in the machine learning. Consider that there are p features in a dataset. While deploying the backward elimination method, we begin by considering all p features in the time, and we eliminate one variable at a time according to the selection criterion, which can be found by the equation $PRESS = \sum_{i} (y_i - \hat{y}_i)^T (y_i - \hat{y}_i)$. In the given equation $\hat{y}_i$, refers to the predicted values from the equation $\hat{y}_i = X_m b_m$ ; where $X_m$ is the calibration / training set with the ith sample removed whereas $m$ refers to the set of regression parameters, and finally $\hat{y}_i$ is the true value of the removed sample from the set. In the backward elimination method, according to the given significance level and adjusted $R^2$ value, features are continuously eliminated until a subset of relevant features are reached.

## 2.6 Response Surface Methodology

In most of the cases, Grid Search and Random Search algorithms are the frequently used hyper parameter tuning techniques. Although metaheuristic methods such as Particle Swarm Optimization and Artificial Bee Colony Optimization are implemented into the machine learning in order to detect the optimal levels of the parameters, further developments are required within this field.

One method that can be used in Neural Networks in order to tune the hyper parameters is Response Surface Methodology. In ML perspective, there are examples of its usage on Random Forest models (Lujan-Moreno, Howard, Rojas  Montgomery, 2018), application of response surface methodology in ANN is not done in the literature. By adopting this methodology into the ANN models, performance and efficiency of the model could be further improved.

Response Surface Methodology (RSM) uses the variables obtained from the experiments, and tries to find the optimum point in a topological space (Khuri  Mukhopadhyay, 2010). Designing the experiment is the most important step in the application of RSM. Number of variables and interval of the variables directly affect the model output in RSM method. Additionally, degree and the curvature of the experimental results affect the outputs gained from an RSM model. If the response in an RSM model can be shown within a linear function of the independent variables, the approximating function in the RSM is called first-order model.

Additionally, if curvature exists in the RSM model and interactions between the variables of the RSM model affect the output, a polynomial of higher degree is used to show the response in the model, such as second-order model. In most of the RSM optimization cases, one or several approximating polynomials are tried to be utilized. Nevertheless, there is not a strict guarantee that obtained polynomial will be a reasonable approximation of the existing relation between the variables and their outputs in the entire space of the independent variables.

RSM can be identified as a sequential procedure (Bezerra et al., 2008). In the first step of the RSM methodology, initial operating conditions are identified as the starting point. A first order model is deployed in the beginning of experiments to move towards the optimality direction faster. However, after a certain threshold, a second order model is deployed after that threshold in order to increase the performance of the model as much as possible. RSM method can be identified as a problem of climbing hill, and only convergence to the local optimum can be guaranteed in an RSM model.

Experiments can be designed in several different ways. Factorial Designs, Composite Designs and Latin Hypercube Designs can be classified as several experiment designs in that sense. In the factorial design experiments, all combination of a number of parameters and their values in the domain are used (Montgomery, 1995). In that case, m factors with n possible values for each factor makes mn entries in a factorial design model. As the complexity of the model is exponential, Fractional Factorial Designs are used in order to overcome this increasing complexity. Fractional designs take some of the variables as the result of other variables, which decreases the number of factors used in a model. In general, although factorial designs usually provide better results, they are costly in terms of algorithm time.

Central Composite Designs on the other hand aim for placing the fractional or embedded factorial designs into the center points that is augmented with a group of star points which provide ease of estimation of the curvature (Ahmadi et al., 2005). Whenever the distance between a factorial point and the center of the design space is 1 for each factor, the distance between a star point and the center of the design space is $a > 1$, whereas the value of  depends on the desired certain properties in the experiment and the number of the involved factors. In the central composite designs, number of the star points are always twice the number of factors.

Additionally, Latin Hypercube Design is another type of experiment which takes randomized combinations of the factor values and collects them in an experiment

(Wang, 2003). In that sense, they can be related to the Monte Carlo simulations. Nevertheless, Latin hypercube designs dont take completely random combinations as Monte Carlo simulations do. Instead of this approach, feature space is divided into sub sample spaces and samples are drawn from these grids in the equal amounts. Instead of a square grid, samples are located in a hypercube. In order to sample from an experiment of N variables within M probable intervals in each of the variables, a uniform cube is formed.

The method of Steepest Ascent is used in RSM. Usually, initial operating conditions in a system are chosen to be farm from optimum. In these cases, primary task to accomplish is to move towards to the direction of optimum rapidly (Hill  Hunter, 1966). The method of steepest ascent is defined as the procedure of the sequential movements towards the path of the steepest ascent, which is the direction of the highest increase in the response. In a minimization case on the other hand, this method is called as the method of steepest descent.

Application of the steepest descent / ascent method involves several sequential steps. After the experiment is designed and initial operating conditions are determined, a first-order linear model which includes no quadratic terms or interactions between the factors is fitted into the data. According to the fitted first order model, the direction which provides the highest improvement is determined and tests are run on the path of steepest ascent until the response of the model improves no more. At that step, curvature of the response surface is examined. If the response surface does not have much curvature, first step is repeated and new experiment is built. However, in the case of a high curvature, a second order model which includes curvature or quadratic terms is deployed. According to the second order model, the path of the steepest ascend / descent is followed until the response no longer improves.

# 3. LITERATURE SURVEY

There are many examples of load forecasting models implemented by scientists. Azedah (2008) applied the fuzzy method that determines the type of ARMA models in load forecasting. Wang (2008) combined autoregressive models and moving average with exogenous variables (such as weather conditions) in electricity forecasting problem. Amjady (2007) employed a hybrid model that com- bines the multilayer perceptron (MLP) neural network and the forecast-aided state estimator (FASE) to indicate load of power systems. Additionally, many of these authors estimated the load by separating the days in 24-hour periods or days of week (Khadem, 1993).

Machine learning algorithms are commonly used for predicting the Photovoltaic energy prediction. In supervised learning algorithms, models try to find a mapping from given inputs to outputs (Inman, Pedro Coimbra, 2013). On contrary, unsupervised learning models seeks to find hidden structure between the input values without the introduction of output variables (Barlow, 1989). In that sense, it is similar to finding the distribution of the inputs. In addition to supervised and unsupervised methods, it is also efficient to combine several successful models in order to increase the overall model performance, and it is called ensemble learning (Gala, Fernandez, Diaz Dorronsoro, 2016).

Support Vector Machine (SVM) is a supervised learning method introduced by Vapnik in order to solve classification and regression problems (Vapnik, 2013). SVM aims to find the best hyperplane that separates the data into different categories in the most accurate way. Support Vectors are the name of data points have the closest distance to the defined hyperplane. Sharma, Sharma, Irwin and Shenoy (2011) applied machine learning models and compared the results within the forecasts of

National Weather Service (NWS). In their study, weather data between January 2010 and December 2010 was used. Weather metrics used in the data are temperature, dew point, wind speed, sky cover, probability of precipitation and relative humidity. Moreover, days and hours are implemented into the data. Training data was chosen between January and August, and as methods, Linear Regression and SVM with Radial Basis Function (RBF) kernel are used. By applying Principal Component Analysis, redundant information in data was eliminated, and as the result of the study, SVM with RBF kernel was found to be more successful within a lower RMS error. Shi et al. (2012) conducted a study in China to predict the photovoltaic production between January 2010 and October 2010. The data interval used in the study is fifteen minutes, and production values are normalized in order to increase the accuracy in the preprocessing phase. Before building the model, according to the weather conditions, data was separated into four categories: sunny, foggy, rainy and cloudy day. RBF kernel was selected for this purpose as stated in the study, it is the most frequently used kernel in these studies. In the result of the study, 12.42% error was obtained in cloudy days, 8.16% error was obtained in foggy days, 9.12% error was obtained in rainy days and 4.85% error was obtained in sunny days.

Random Forest (RF) is an ensemble method which is used for regression and classification tasks (Breiman, 2001). It essentially combines K number of decision trees and obtains an output. Study of Huertas Tato and Centeno Brito (2019) focuses on predicting the photovoltaic production of six solar panels at Faro (Portugal). In the study, data contains temperature, meteorological variables and radiation as attributes. Three years of data within the minute intervals were used. Number of trees is the most important hyper parameter in RF method, and it is tuned to 500 by brute force approach. In conclusion, it is stated that module type of solar plant affects the performance of RF method, and it is possible to improve the performance by conducting complex trend analysis, more relevant data and wider prediction intervals.

Nearest Neighborhood algorithms are instance-based supervised learning algorithms which aims to find an output by performing local approximations and identifying the closest data. By assigning weights to the determined number of closest neighbors, output is computed. Voyant, Paol, Muselli and Nivet (2013) conducted a study to assess the performance of forecasting methods for predicting photovoltaic production for different horizon. K-NN is used in the study as it is essential to use naive models to verify the relevance of complex models. In the result of the study, it is discovered that on daily basis, k-NN is a viable option for forecasting. However, k-NN is not the optimal model for hourly forecast according to the result of the study.

Furthermore, Panapakidis and Athanasios (2016) focuses on day-ahead electricity price forecasting within the usage of Artificial Neural Network (ANN) approach. The used data in the study belongs to South Italy Electricity Market, and seven different ANN models within their specific attributes. Attributes mainly includes day and hour labels, predicted loads, and sliding windows on electricity price. Logistic sigmoid, hyperbolic tangent sigmoid and linear activation functions are used in the neurons, the neural network includes single hidden layer with a varying number of nodes from 2 to 30 within the step size of 2, and epoch number is determined as 500. As results, lowest Mean Absolute Percentage Error (MAPE) was obtained from model which used the electricity price of other countries as an input variable, and the model which used a cas- caded structure in order to ensemble 2 models. In the cascaded structure, first model calculates the hourly price of the days, and the second model uses the output of the first model in addition to previously used parameters in order to smooth the initial prediction made by the first model. Lowest MAPE obtained in the study is around 18% by these models.

Tibshirani (1996) developed a new regression method which minimizes sum of squares within the consideration of sum of absolute values of coefficients being less than a determined constant. It gives a sparse solution with most of the coefficients being zero. This regression uses L1 norm in the algebra, and also called as LASSO (Least Absolute Shrinkage Selector Operator). Model is used on a prostate cancer data,

and in three different scenarios. First scenario is consisted of a data where a small number attributes contribute a large effect. The second scenario is consisted of a data where small to moderate number of attributes contribute moderate effect, and the third scenario is consisted of a data where large number of attributes contribute small effect. 3 methods are compared in these scenarios, namely LASSO, Ridge Regression (L2 Regression) and Subset Selection. In the first scenario, Subset Selection performs the best whereas other two models perform poorly. In the second scenario, LASSO performs the best, and in the third model, ridge regression and lasso perform the best. It can be concluded that in problems that are like the second case, LASSO regression is a viable option.

Luo, Hong and Fang (2018) present three new regression models in order to solve the data integrity problems existing in the proposed models in the literature. Data integrity problems are mostly caused by the under-forecasts, as they can possibly cause blackouts. Main motivation of the study was to solve sudden anomalies in electricity load data rather than solving Features in presented models include trend, time variables and temperature. Two of the models were Iteratively Reweighted Least Squares (IRLS) models, where the observations with the larger residuals were considered as anomalies, and forecasting these anomalies contributes to objective function less while accuracy on normal data points had a greater reward. First IRLS method assigns relatively small weights to the residuals with a large value whereas second IRLS method deletes the residuals above a threshold. Third model is an L1 regression model, and in the results, all models gave accurate results and especially L1 model was successful even when the 30% of data included anomalies, with an accuracy around 10%, as L1 model was less sensitive to outliers.

Ishik et al. (2015) implemented a feed-forward neural network for case study of short-term electricity load forecasting in Turkey. In this study, the authors focused on short term electricity demand forecasting. Ishik et al. trained a feed- forward neural network by using Levenberg- Marquardt algorithm to predict the next day load in the electricity market of Turkey. Their data file contains hour, day of week,

month, year, temperature of cities and the electricity load. There are six units of for input variable, and output variable of the study is hourly electricity load. The data samples of 2012 weekdays are randomly divided with 70% training set, 15% validation set and 15% test set. The authors separated the network for seasons of the year in Turkey and trained SVM to compare with their network. The results of their study show that final accuracy of the neural network is similar with SVM and the neural network gives better results for winter and spring data while Support Vector Machine predictions are better for summer and fall. MAPE for each season are between 2.0  3.7.

Moreover, adjusting the hyper parameters of the built models is another challenge in machine learning approaches. In ANN case the number of nodes, hidden layers, activation functions, optimizers, batch size and epoch numbers have in- finitely many combinations, thus trying all settings is computationally impossible. For that purpose, there are various approaches for hyper parameter tuning in the literature. Most common approach for optimizing the hyper parameters is using Grid Search method, which iteratively tries all of the possible combinations thus takes vast computational time. Bergstra and Bengio (2012) proposed a new methodology named Random Search which tries the random settings in the determined amount of iterations in order to find the optimal set of parameters, and it is demonstrated in this study that grid search is inferior to this methodology. Hinton (2012) shares the ideal values for several parameters such as batch size, learning rate, momentum and number of hidden units. Lujan-Moreno et al. (2018) propose a methodology which uses Response Sur- face Methodology and Design of Experiments methods in order to optimize number of trees, features, node sizes and number of maximum number of nodes in RF methodology.

Gala et al. (2016) conducted a study in Spain and applied SVR, Gradient Boosted Regression, Random Forest and a hybrid method which combines them in order to predict day-ahead and 3-hourly solar irradiance. The radiation data used in the study included hourly data for every day between October 2009 and July 2011. Their

main contribution to the literature is to develop a new method for downscaling solar irradiance, and decompose three-hour aggregated radiation into hourly values using the concept of a local empirical CS radiation curve. MAE (Mean Absolute Error) is used as the metric in the study, and built forecast models are compared with ECMWF (European Center for Medium Weather Forecast). Average irradiance value in the used data is 1370 W/m2. They evaluated values for several stations, but in general, SVR per- formed better on both 3-hour and daily models. Second best performing model in the study is the hybrid model, which is a combination of GBR, SVR and RF. Persson et al. conducted (2017) deployed Gradient Boosted Regression Tree (GBRT) method in their study. Data used in the study had a range between April 2014 and 2015. On hourly basis, weather forecasts and power observations exists in the used data. First 75% of the data is used training data and 5-fold cross validation is applied in order to optimize hyper parameters of the model. Power data includes hourly averages of 42 photovoltaic production plants in Nagoya Bay, Japan. Night hours are discarded from data as they mainly included zero production. In the result section, GBRT is compared within benchmark models, which are Persistence, Climatology and Recursive AR (Autoregressive). GBRT outperformed all other models, and additionally, it is noted that normalizing the data didnt contribute major benefits to the results.

Massidda and Marrocu (2017) deployed Multilinear Adaptive Regression Splines (MARS) method in order to predict photovoltaic production. MARS is a frequently used approach in data mining, and it doesnt take any assumption about the relationship between the features. A relationship between basis functions and a set of coefficients is constructed in MARS method, and divide and conquer approach is taken, which refers to splitting the input space into the regions and applying regression equations separately onto these regions. First phase of MARS method involves the calculation of intercept of the regression model and the deployment of the basis functions repeatedly. Second phase involves elimination of basis functions which provide the minimal increments in the accuracy of fit until the best sub model is found. Study is conducted in Borkum, a German island and the data which was

used included 15-minute interval data. 3-hourly forecasts are done. As benchmark model, Persistence is used. MARS method performed better in this study, and it is also observed that resolution of the data affects the performance of MARS method.

Bouzgou and Gueymard (2017) proposes a new methodology which combines Extreme Learning Machine (ELM) and mutual information measures, in order to predict global solar irradiation. The ensemble method in the study includes two steps; in the first steps dimensionality of the problem is reduced and in the second step, ELM is used in order to forecast. The method is tested in different time horizons, and it is observed that best possible dimensionality reduction strategy for the first step is Minimum Redundancy  Maximum Relevance (MRMR) method. Additionally, it is noted that accuracy of the ELM decreases inversely proportional within the cloud frequency.

Shakya et al. (2017) conducted a study by deploying a novel method, Markov Switching Model (MSM) for irradiance forecasting problem. Proposed method uses locally available data in order to forecast the solar irradiance amount for the next day. MSM was tested on data of 5 years, and the best model reached 31.8% MAPE. Marzo et al. (2017) deployed ANN models in order to predict the solar ir- radiation in desert areas. Features of the model are meters above sea level, daily angle, solar declination, zenith angle cosine, sunrise hourly angle, maximum temperature, minimum temperature and extraterrestrial solar radiation. Inputs and outputs are normalized in the study, and hidden nodes between 2 and 30 are tested. As the result of the study, 13% Root Mean Square Deviation (RRMSD) is reached.

Leva et al. (2017) deployed ANN model with 9 neurons in the first layer, 7 neurons in the second layer and 3000 iterations per trial. Forecasts in the study are done in hourly basis for next 30 days. As the result of the study, built model performed good in sunny days and slightly worse in cloudy and partially cloudy days. Torres-Barr an, Alonso and Dorronsoro (2019) applied Random Forest, Gradient Boosted Regression and Extreme Gradient Boosting in their study. Their study included forecasts with

daily horizon for 1 year, and for benchmarking, Multilayer Perceptron and Support Vector Regression are used. As the result of the study, lowest MAE is obtained from Gradient Boosted Regression and Extreme Gradient Boosting Regression models.

Pierro et al. (2017) conducted a study to measure the efficiency of deterministic and stochastic techniques in order to predict photovoltaic production. Weather and power data between January 2011 and December 2014, belonging to Airport Bolzano Dolomiti, Italy is used. Persistence model is used for benchmarking in this study. Data resolutions in the study were 15 minutes 1 hour. Daily forecast for a year is done in the study for testing the models. Stochastic models in the study performed remarkably better compared to deterministic models. Ibrahim and Khatib (2017) developed a novel-hybrid model for predicting hourly global solar radiation. Hourly prediction model uses three stages. In the first stage, RF method is trained by Bagger Algorithm, feature importances are measured, cluster analysis is executed and outliers are removed from the sample. In the second stage, after new variables are modified in dataset, number of trees and leaves in the RF model are optimized, Bagger algorithm for training is used and final evaluations are done. Number of trees and leaves are optimized by Firefly algorithm, and used metric in this process is chosen as RMSE. For benchmarking, conventional and optimized neural networks are used. Results of the study indicate that new hybrid model is superior to neural network models within 6.38% estimated MAPE and model execution speed.

Alfadda, Rahman and Pipattanasomporn (2018) conducted a study on Saudi Arabia and developed a model for photovoltaic production prediction in desert areas by including sand as a feature. MLP, KNN, SVR and DT models are developed in the study. It is determined in the study that MLP models are more suitable for desert areas, as it had achieved Mean Square Average Error of under 4%.

# 4. METHODOLOGY

For this thesis, models for solar radiance forecasting are built within the usage of Python language. Python is mainly used for data preprocessing phase and for the development of the machine learning models.

Main aim of the data prprocessing is to transform the features in the original dataset in order to use it mathematically. There are several steps which are required to follow in order to complete a KDD (Knowledge Discovery Process). Data cleaning is the initial step to be completed in a KDD process (Fayyad et al., 1996). Attribute values may include flaws in itself. Incomplete data refers to the blank cells in a column, and it causes neural networks to give not applicable results. Incomplete data can be fixed by replacing the values within the median, mean or mode of the values in the range. Alternatively, copying the values of the nearest neighbors is an alternative approach. Inconsistent and intentional errors may also occur in the data, and they may require manual handling. Alternatively, ignoring the tuples where errors occur can be applied to the data. Nevertheless, it results in fewer tuples in the data, which affects the model performance. Data integration is the next step in a KDD process, coming after the data cleaning. Clearing the redundancies and inconsistencies in a dataset covers the data integration process. Data redundancy refers to the unnecessary columns in a dataset. If a variable can be found as a function of other variables, it is referred a redundancy. In redundant datasets, multicollinearity occurs. Multicollinearity refers to the situation where affecting features in a model can be represented by different settings of parameters, which makes it difficult to determine the importance of the features. Additionally, correlation analysis helps to identify the relations between the variables and determine the related features. Both correlation analysis and redundancy clearing helps lowering the complexity of

the model.

Data reduction is the third step in the KDD process. It aims for achieving the same results in a model within a lower number of features and a smaller sample size. Dimensionality reduction is done by using feature selection and feature extraction methods which are mentioned in the chapter two. Smaller sample size is achieved by special sampling strategies. There are four frequently used sampling strategies: simple random sampling, sampling without replacement, sampling with replacement and stratified sampling. Simple random sampling is basically randomly selecting tuples within equal probability. Sampling without replacement is applicable for the datasets where identical tuples exist; and once a tuple is selected, its duplicates are removed from the population. In sampling without replacement, duplicates are not removed from the population. In stratified sampling, dataset is partitioned into the clusters, and sampling is done equally from each of these clusters.

Last step of the KDD process is the data transformation and discretization. Initially, type of each attribute must be determined in a dataset for this step. Attributes in a dataset can take values in four different types; which are Nominal, Binary, Ordinal and Numeric. Nominal values refer to the values which are simply labels and dont represent any mathematical values. Days of the weeks, uniform numbers of the sport players, marital status, occupation and zip codes are nominal valued attributes. Binary values refer to the values that take 0 and 1. Binary values can be symmetric and asymmetric. Symmetric values have equal importance; such as gender and coin sides. On the other hand, asymmetric values have different importance; such as medical test results. Ordinal values refer to the values which imply the rank of an attribute. Ranking and grades are examples for the ordinal values. Numeric values can be interval-scaled and ratio-scaled. Interval-scaled values are measured on a scale with equal intervals, such as calendar dates and temperature. Ratio-scaled values include a zero point and provides us the ability to establish a mathematical relation. Length, Kelvin temperature and counts are examples for the ratio-scaled values.

In order to process nominal values, a number for each element in the range is replaced within the original values. Because of the fact that there is no mathematical relation between the nominal values, encoding scheme is established on the values. One-Hot Encoding (OHE) and Binary Encoding (BE) are the most frequently used encoding methods. OHE generates columns for each of the specific values of an attribute. In other words, if there are different values for an attribute, columns are generated. Sum of all the values in these column for each of the tuples is equal to 1, referring that a vector cant take a value different than zero in the encoded columns. Differently from OHE, BE aims to generate encoded columns that are less in numbers compared to OHE. By using columns, up to different values can be represented by BE. Both of these methods have pros and cons. BE is less complex in terms of vector space, but feature selection and extraction methods dont work properly on variables that are encoded with BE. Although OHE overcomes this disadvantage, it is computationally inefficient and may lead to inaccurate results as it has mostly zeroes in its columns. Ordinal values can also be encoded these methods. Nevertheless, binary variables dont require encoding by nature.

Numeric values can be processed by using scaling methods. There are mainly two methods for the feature scaling. Normalization is done by finding the minimum and maximum values in the range of an attribute, and using the formula 4.1 below. In the equation, x refers to the numerical value of the attribute, and x' refers to the new value of the attribute. Essentially, normalization causes an attribute to take values between 0 and 1. Nevertheless, this interval can be arranged in any range for specific problems.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{4.1}$$

Standardization is another frequently used method for feature scaling. Aim of standardization is to replace the values with new values according to the distribution of the variable. Standard deviation and mean of the attribute is required in order to

complete standardization process on a variable. It is done according to the formula 4.2 below.

$$x' = \frac{x - \overline{x}}{\sigma} \tag{4.2}$$

In the formula, x' refers to the replaced value, $\overline{x}$ refers to the mean of the values of the attributes and $\sigma$ refers to the standard deviation of the attribute. Selection of the scaling method affects the outputs of the ML models. As nature, standardization allows negative values to be in range, whereas the length of the range cant be foreseen. On the other hand, although normalization takes the values in a predefined range, it is extremely sensitive to outliers because of the fact that it only considers minimum and maximum values in a range.

In order to choose the optimal scaling strategy, distribution of the data must be analyzed. Central tendency and dispersion of the data can be analyzed within the usage of several indicators. Indicators for the central tendency are mean, weighted arithmetic mean, trimmed mean, median, mode and midrange. Mean of the attribute can be determined by dividing the sum of the variables in a range by the number of the elements. Weighted arithmetic mean is determined by assigning weights to the individual data points. Trimmed mean is determined by excluding the outliers in an attribute range. Median refers to the middle value in the variable range. Mode refers to the most occurring value in an attribute range. Lastly, midrange is determined by taking the average of the highest and lowest value in a domain. Especially, mean and median can be used to determine the skewness of a distribution. In symmetric distributions, mean, mode and median are equal to each other. In positively skewed data, the mode is lower than the median and the median is lower than the mean. In negatively skewed data, the mean is lower than the median and the median is lower than the mean. Apart from the central tendency, dispersion of the data is also essential. Variance and quantiles are important indicators for data dispersion.

Proposed dataset includes approximately 350,000 rows which includes information about GHI for the June, July, August months of 2016, 2017, 2018 and May, June, July of 2019. Existing features in the dataset are listed below:

- Air Temperature
- Relative Humidity
- Wind Speed
- Peak Wind Speed
- Wind Direction
- Estimated Pressure
- Precipitation
- Accumulated Precipitation

In the raw dataset, test and training sets are generated in the first step. Initially, data belonging to 2016, 2017 and 2018 are used as training set whereas 2019 data are used as test set. In the next step, random sampling is applied to both test and training data independently. As a result, 25000 tuples for training and 5000 tuples for test set are chosen. Preprocessing steps are applied as a whole on the dataset.

Air temperature is taken from Kelvin type and it is normalized between 0 and 1. Additionally, 5 sliding windows are integrated into the data. These windows represent the temperature values up to 5 minutes prior to the data tuple. Relative humidity exists in dataset as percentage value. In a similar manner, it is normalized between 0 and 1, then 5 sliding windows are integrated into the dataset. Wind speed is given as feet/second in the original dataset. It is standardized differently from the previous features because of the fact that extreme outliers exist within the column. 5 sliding windows are integrated into the wind speed feature. Peak wind speed is processed in the same methodology. Wind direction is given in the original dataset as a direction between 0 and 360. Directions are classified according to 4 basic directions, and two feature columns are generated in order to represent wind direction in the dataset. Main reason of using two columns is to decrease redundancy

in the dataset. For instance, if north feature of a column takes 1, south feature of the column takes 0; hence it becomes possible to predict the value of a feature by using another feature; which eventually causes multicollinearity. Estimated pressure and precipitation features are normalized between 0 and 1, then sliding windows of 5 are integrated. CR800 temperature and RSR Battery features are related to the solar panel and dont affect the GHI amount, thus they are not included in the dataset.

After the sampling and preprocessing steps are completed, feature processing and extraction methods are applied on the dataset. Within the usage of Sci-kit learn library of the Python; PCA and Factor Analysis are applied as feature extraction methods and Extra Trees Regressor is applied as feature selection method. Additionally, a separate code is written in order to implement Backward Elimination as a feature selection method. Assessment criteria for these dimensionality reduction methods are unique. Factor analysis determines the selection variables. PCA aims for representing dataset with principal components, and each principal component possesses an explained variance ratio. Backward elimination determines the significant variables under given significance level, which is represented by $\alpha$. Extra trees regressor ranks the features from the most important to the least important, within their contributions to the output. Among these dimensionality reduction methods, the ones which gave more relevant features are chosen.

After the data preprocessing and dimensionality reduction techniques are applied, next step is to build ML models. Essentially, 4 different methods are used; which are SVM, RF, ANN and K-NN. Hyper parameter settings for each of these methods are unique.

SVM hyper parameters are error term, kernel, degree, gamma, zero coefficient, shrinking, probability and tolerance of stopping criterion. Error term, which is denoted by represents the penalty parameter of the error term. Kernel determines the type of the kernel to be used. Mainly, three types of kernels are used in SVM; which are radial basis function (rbf), polynomial and linear. Linear kernels aim

for separating the data points using a linear perceptron. Polynomial kernels aim for separating the data points by using a polynomial kernel of a specified degree. Rbf kernels use a radial basis function, which covers the area in a vector space and separate the data points. In general, there are trade-offs between these kernel types. Linear kernels are computationally less complex and they are better in generalization of the models. Polynomial kernels can be adjusted according to the degree hyper parameter in SVM. Kernels of lower degrees can generalize better whereas higher degrees may represent the data with higher accuracy. In general, the risk of underfitting is highest in linear kernels and lowest in rbf kernels. Nevertheless, the risk of overfitting is highest in rbf kernels and lowest in linear kernels. Gamma coefficient determines the influence of a single training example on a model. Intuitively, giving a high value to the gamma coefficient makes this influence high, and a low value makes this influence low. Default value for the gamma coefficient is 0,22. Shrinking parameter in an SVM is about using the shrinking heuristics. Probability parameter in SVM is about using the probability estimates in the built model. Tolerance of stopping criterion about determining the distance to the support vectors. In an ideal case, support vectors have a distance of 0 to the kernel. Nevertheless, it may be unreachable in many models, thus tolerance parameter determines this distance.

RF hyper parameters are number of estimators, maximum number of features, maximum depth of decision trees, minimum number of data points for splitting, minimum number of data points allowed in a leaf node and bootstrap. Number of estimators determines the used decision trees in an RF ensemble. Low number of decision trees may result in underfitting whereas high number of decision trees may result in overfitting. Maximum depth of a decision tree determines the total splitting operations done in each of the decision trees. Maximum number of features determines the amount of different features to be used in order to execute a splitting operation in nodes. Minimum number of data points for splitting determines the minimum amount of training samples for each node. Bootstrap parameter determines the sampling method used in decision trees (sampling with or without replacement).

K-NN hyper parameters are basically number of neighbors and used distance metric. Number of neighbors are usually selected from the set of odd numbers in classification models, because of the fact that majority of votes are counted in K-NN classifiers. Nevertheless, this limitation does not hold for regression problems as weighted average of the neighbors are taken into the account. Low number of neighbors may cause biased results whereas high number of neighbors may result in low variance. Used distance metric determines the type of Minkowski distance in the formula . N determines the order of the Minkowski distance, and results vary according to the distance metric. Giving the value of 1 results in having Manhattan distance, 2 results in Euclidean distance and results in Supremum distance.

In ANN case, the types of hyper parameters vary significantly. Number of layers, number of nodes, activation functions, number of epochs, optimizer type, loss function and batch size are hyper parameters to be optimized. Number of layers dramatically affect the performance of the model. Increased number of hidden layers reduce the overfitting in neural networks. Nevertheless, there is a computational trade-off because of the fact that it takes more time to fit neural networks with a high number of layers. In general, between 3 and 10 hidden layers are used generally in the literature. Number of nodes on the other hand, are directly related within the hidden layers. For each layer, different number of nodes can be specified. There is not a general number of nodes to be chosen. Nevertheless, in the most of the studies in the literature, number of nodes per layer is chosen as to be half of the input dimension as a rule of thumb. In the last layer, number of nodes is chosen to be 1 in regression models. For classification models, multiple nodes can be used in the output layer especially in multi-class classification problems. Activation functions in each layer dramatically affect the output in neural networks. In general, for hyperbolic tangent (tanh) and sigmoid functions are used in classification problems. For regression problems; linear, elu, selu and relu are used as activation functions. Although linear activation function is used in the output layer in general in regression problems, tanh and sigmoid functions can also be used in the output layer by scaling the output of the training set. Tanh and sigmoid functions significantly suffer

from the saturation effect in gradient descent algorithm, which is about the reduced effect of these activation functions when the value is between the boundaries. It can be solved by executing the scaling operation between a limited range, such as between 0,4 and 0,7. Relu is the most frequently used function in regression models. Additionally, usage of linear activation function in all of the layers does not make a difference in the outputs of the model. An epoch on the other hand, number of times that a neural network passes on the defined training sets. A whole iteration of back-propagation on these examples are called an epoch. Increasing the epoch number in a neural causes training set accuracy to be high. Nevertheless, it causes overfitting in the model because of the fact that the model becomes more dependent to the training set. There is not an absolute rule in the selection of number of epochs. However, early stopping method can be used in order to adjust number of epochs. In the early stopping method, accuracy on the validation set is monitored, and epoch numbers are increased until the accuracy on validation set no longer increases. A tolerance limit is defined in the early stopping method and stopping criteria is set based on this tolerance limit. For instance, if tolerance limit is set to , model stops after epochs if validation set accuracy doesnt increase. Optimizer type on the other hand are vital for running the neural networks. There are several frequently used optimizer types for neural network models in the literature. ADAM, Adagrad, Stochastic Gradient Descent (SGD), Nesterov Momentum are among these frequently used optimizers. Each of these optimizers have their distinct parameter types. A common parameter in these optimizers is learning rate hyper parameter. Learning rate is directly used in gradient descent formula and decides how fast the weights will change in each epoch. A high learning rate causes model to miss optimum level of weights, whereas a slow learning rate causes model to shift weights slowly. Thus, there is an explicit trade-off between accuracy and computation. Generally, 0,4 is taken as the default value for the learning rate. Related within the optimizers, adjusting batch size is essential for neural networks. Batch size refers for number of training examples to be selected in gradient descent algorithm. If there are number of tuples in a batch, gradient descent algorithm passes of these tuples and updates the weights according to that. Generally, there are three types of selecting the batch size. First approach

is taking all of the training examples as a batch. Second approach is selecting a single random tuple to use in gradient descent algorithm, which is also referred as Stochastic Gradient Descent. It generally produces more robust outputs compared to taking all of the examples inside a batch. Third approach is using a mini-batch for a gradient descent algorithm. As a rule of thumb, size of the mini-batch is chosen to be a power of 2; such as 16, 32, 64, 128 and 256. Lastly, selection of the loss function affects the performance of the models dramatically. Generally, MAPE, MAE and MSE are frequently used as loss functions; although it is possible to use a custom loss function. It is computationally easier to use a loss function which can be derivable, as gradient descent algorithm takes the derivative of the loss function. In addition to the computational complexity, outputs are directly influenced from the selection of loss function, as mentioned in chapter 2 of this thesis.

Main aim of this thesis is to improve the performance of neural networks through the novel feature usage and hyper parameter optimization, thus other mentioned models will be used for benchmarking. In order to make the calculations computationally efficient and reach a reasonable solution, several assumptions are made in the model settings. For SVM, only kernel types are adjusted and other hyper parameters are taken as default values. For RF, only number of estimators are adjusted. For K-NN, distance metric is chosen as Euclidean distance and number of neighbors are adjusted. Nevertheless, for neural network models, more type of hyper parameters is tuned. As optimizer type, ADAM is chosen and parameters of the ADAM are taken as default values. Models that are built include layers between 1 and 4. Number of epochs are chosen to be 100, 200, 300 and 400 initially, and after the optimization process is completed; number of epochs are adjusted through the usage of early stopping method. For activation functions, only Relu and Elu will be used and in all of the layers, activation function will be used as the same in order to reduce computational complexity. For 1-layer models however, linear activation function will also be used. Number of nodes and batch size will be optimized using hyper parameter tuning methods. As loss function, MAPE will be used.

For hyper parameter tuning, essentially two approaches will be taken; which are Random Search, Grid Search. In grid search method, a parameter grid is defined for the method and a k-fold cross validation is applied on training set in order to determine the best setting of hyper parameters. In k-fold cross validation, training set is divided into parts and by using parts, last part is predicted in training set. This process is done times and average score is taken as the accuracy. Grid search method tries all the combinations which are defined in a grid. Random search on the other hand, tries the random settings in the defined number of iterations. Although they are widely used for hyper parameter tuning process, they are incapable in reaching the optimum level, in addition to their computational inefficiency. RSM method on the other hand, although is not used for hyper parameter tuning in neural network models, it offers less computational complexity and a wider of hyper parameter grid. In adaptation of RSM method into the neural networks, an experimental design will be generated based on the outputs of the generated neural network models. There are essentially five types of experimental designs that can be used in RSM. Full factorial designs aim for using all of the possible combinations of the input features. Fractional factorial designs include the selection of a subset of all combinations, and Latin hypercube designs include selection of random features for each of the possible experimental objects. Central composite designs are used in RSM models for building a quadratic model for the response feature without the requirement of a complete three-level factorial design. Linear regression is used iteratively to reach a conclusion. Lastly, Box-Behnken design is a type of response surface design which does not include the embedded fractional or factorial designs. Instead, they possess treatment combinations which can be classified as the midpoints of the vertices of experimental space and must include at least three surface variables which are continuous. First and second-order coefficients can be estimated using the Box-Behnken designs. Additionally, since they include fewer design points, they are computationally less expensive compared to central composite designs which have the same number of factors. Nevertheless, due to the fact that they do not have an embedded factorial design, they are not suitable for sequential experiments.

Table below indicated the results of PCA applied on the dataset. Principal components are taken up to 15 because of the fact that their impact decreases after 15 components.As seen in the table, first two principal components explain more than the half of the variance in the data. Nevertheless, significance is reduced after the 4th component and after the 9th component, impact is dramatically reduced.

**Table 4.1** PCA Results

| P.Comp. | Sing. Val. | Exp. Var. | Exp. Var. R |
|---|---|---|---|
| 1 | 175,28 | 1,54 | 0.322 |
| 2 | 150,4 | 1,13 | 0.238 |
| 3 | 120,54 | 0,73 | 0.153 |
| 4 | 65,18 | 0,21 | 0.045 |
| 5 | 54,03 | 0.146 | 0,03 |
| 6 | 50,68 | 0.128 | 0.027 |
| 7 | 47,54 | 0.113 | 0.024 |
| 8 | 44,7 | 0,01 | 0.021 |
| 9 | 42,51 | 0,09 | 0.019 |
| 10 | 40,53 | 0.082 | 0.017 |
| 11 | 38,75 | 0.075 | 0.016 |
| 12 | 38,5 | 0.074 | 0.016 |
| 13 | 36,67 | 0.067 | 0.014 |
| 14 | 34,54 | 0,06 | 0.013 |
| 15 | 33,12 | 0.054 | 0.012 |

# 5. RESULTS

Grid search is run on the neural networks and other machine learning models in order to determine the optimum set of hyperparameters. Table 5.1, 5.2, 5.3 and 5.4 indicate the grid search results for 1-4 layer networks. Results of table 5.1 indicate that although a MAPE score of 0,23 is reached at the best, $R^2$ value - which indicated the explained variance in the regression model - is not significant. Generally, Elu is the best activation function to use in single layer networks. Ideal batch size is determined as 64, though 32 and 16 batch sizes give acceptable results on single layer.

Furthermore, table 5.2 indicates the grid search results on 2-layer networks. There is a significant increase in the MAPE compared to the single layer networks, and $R^2$ value increases up to 0,3 from 0,23. Thus, it can be interpreted that accuracy of the model increases within the explained variance in the model. Similar to single layer network, Elu is the most fitting activation function in the hidden layers. Although different batch sizes give reasonable results, best results are obtained from the batch size of 64. Table 5.3 indicates the grid search results on 3-layer networks. Lowest MAPE acquired from these networks are 0,09 and it is a significant drop compared to 2-layer networks. Additionally, explained variance of the model increases dramatically compared to the previous models. As previous networks, Elu is the most fitting function. Batch sizes of 32 and 64 give the best results. Lastly, table 5.4 shows the results for 4-layer networks. Although best MAPE and $R^2$ value is close to the ones obtained in 3-layer networks, these networks are still improved version of 3-layer networks. Relu activation functions perform better in the models.

In addition to the original dataset, optimum set of hyper parameters are also deter-

**Table 5.1** Single-Layer Grid Search Results

| Node | A.Func. | Batch | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|---|---|
| 100 | Relu | Whole | 122 | 12246 | 0,35 | 0,27 |
| 100 | Elu | Whole | 149 | 20936 | 0,43 | 0,33 |
| 100 | Relu | 16 | 157 | 22566 | 0,45 | 0,35 |
| 100 | Elu | 16 | 103 | 7619 | 0,29 | 0,23 |
| 100 | Relu | 32 | 120 | 12031 | 0,34 | 0,27 |
| 100 | Elu | 32 | 118 | 11334 | 0,34 | 0,26 |
| 100 | Relu | 64 | 148 | 19299 | 0,42 | 0,33 |
| 100 | Elu | 64 | 80 | 4980 | 0,23 | 0,18 |
| 100 | Relu | 128 | 144 | 20040 | 0,41 | 0,32 |
| 100 | Elu | 128 | 108 | 10785 | 0,31 | 0,24 |
| 200 | Relu | Whole | 105 | 9961 | 0,30 | 0,23 |
| 200 | Elu | Whole | 110 | 11621 | 0,31 | 0,24 |
| 200 | Relu | 16 | 120 | 12774 | 0,34 | 0,27 |
| 200 | Elu | 16 | 158 | 23723 | 0,45 | 0,35 |
| 200 | Relu | 32 | 139 | 18421 | 0,40 | 0,31 |
| 200 | Elu | 32 | 101 | 9269 | 0,29 | 0,22 |
| 200 | Relu | 64 | 89 | 5234 | 0,25 | 0,20 |
| 200 | Elu | 64 | 82 | 4072 | 0,23 | 0,18 |
| 200 | Relu | 128 | 148 | 19139 | 0,42 | 0,33 |
| 200 | Elu | 128 | 136 | 16718 | 0,39 | 0,30 |
| 300 | Relu | Whole | 131 | 14723 | 0,37 | 0,29 |
| 300 | Elu | Whole | 153 | 22456 | 0,44 | 0,34 |
| 300 | Relu | 16 | 168 | 27294 | 0,48 | 0,37 |
| 300 | Elu | 16 | 104 | 10216 | 0,30 | 0,23 |
| 300 | Relu | 32 | 125 | 13312 | 0,36 | 0,28 |
| 300 | Elu | 32 | 139 | 17804 | 0,40 | 0,31 |
| 300 | Relu | 64 | 101 | 9645 | 0,29 | 0,22 |
| 300 | Elu | 64 | 125 | 13896 | 0,36 | 0,28 |
| 300 | Relu | 128 | 128 | 13393 | 0,37 | 0,28 |
| 300 | Elu | 128 | 121 | 11742 | 0,35 | 0,27 |

**Table 5.2** 2-Layer Grid Search Results

| Node | A.Func. | Batch | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|---|---|
| 100 | Relu | Whole | 125 | 15250 | 0,36 | 0,50 |
| 100 | Elu | Whole | 89 | 6808 | 0,25 | 0,36 |
| 100 | Relu | 16 | 99 | 8226 | 0,28 | 0,40 |
| 100 | Elu | 16 | 50 | 1669 | 0,14 | 0,20 |
| 100 | Relu | 32 | 133 | 16129 | 0,38 | 0,53 |
| 100 | Elu | 32 | 89 | 7556 | 0,25 | 0,36 |
| 100 | Relu | 64 | 131 | 14920 | 0,37 | 0,52 |
| 100 | Elu | 64 | 74 | 5062 | 0,21 | 0,30 |
| 100 | Relu | 128 | 96 | 7096 | 0,27 | 0,38 |
| 100 | Elu | 128 | 121 | 14210 | 0,35 | 0,48 |
| 200 | Relu | Whole | 66 | 3752 | 0,19 | 0,26 |
| 200 | Elu | Whole | 88 | 5332 | 0,25 | 0,35 |
| 200 | Relu | 16 | 137 | 17840 | 0,39 | 0,55 |
| 200 | Elu | 16 | 82 | 5930 | 0,23 | 0,33 |
| 200 | Relu | 32 | 95 | 8668 | 0,27 | 0,38 |
| 200 | Elu | 32 | 63 | 2815 | 0,18 | 0,25 |
| 200 | Relu | 64 | 68 | 4184 | 0,19 | 0,27 |
| 200 | Elu | 64 | 107 | 10999 | 0,31 | 0,43 |
| 200 | Relu | 128 | 81 | 6101 | 0,23 | 0,32 |
| 200 | Elu | 128 | 84 | 5712 | 0,24 | 0,34 |
| 300 | Relu | Whole | 79 | 5600 | 0,23 | 0,32 |
| 300 | Elu | Whole | 80 | 5461 | 0,23 | 0,32 |
| 300 | Relu | 16 | 132 | 15212 | 0,38 | 0,53 |
| 300 | Elu | 16 | 137 | 16277 | 0,39 | 0,55 |
| 300 | Relu | 32 | 56 | 1143 | 0,16 | 0,22 |
| 300 | Elu | 32 | 91 | 5898 | 0,26 | 0,36 |
| 300 | Relu | 64 | 84 | 4802 | 0,24 | 0,34 |
| 300 | Elu | 64 | 52 | 2852 | 0,15 | 0,21 |
| 300 | Relu | 128 | 87 | 6949 | 0,25 | 0,35 |
| 300 | Elu | 128 | 81 | 4227 | 0,23 | 0,32 |

**Table 5.3** 3-Layer Greed Search Results

| Node | A.Func. | Batch | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|---|---|
| 100 | Relu | Whole | 118 | 13021 | 0,34 | 0,31 |
| 100 | Elu | Whole | 56 | 2008 | 0,16 | 0,62 |
| 100 | Relu | 16 | 105 | 10319 | 0,30 | 0,38 |
| 100 | Elu | 16 | 111 | 11994 | 0,32 | 0,35 |
| 100 | Relu | 32 | 108 | 11174 | 0,31 | 0,36 |
| 100 | Elu | 32 | 99 | 9470 | 0,28 | 0,41 |
| 100 | Relu | 64 | 91 | 7227 | 0,26 | 0,45 |
| 100 | Elu | 64 | 36 | 7169 | 0,10 | 0,72 |
| 100 | Relu | 128 | 85 | 6026 | 0,24 | 0,48 |
| 100 | Elu | 128 | 62 | 2375 | 0,18 | 0,59 |
| 200 | Relu | Whole | 30 | 1208 | 0,09 | 0,75 |
| 200 | Elu | Whole | 99 | 9626 | 0,28 | 0,41 |
| 200 | Relu | 16 | 64 | 2937 | 0,18 | 0,58 |
| 200 | Elu | 16 | 103 | 10518 | 0,29 | 0,39 |
| 200 | Relu | 32 | 99 | 8487 | 0,28 | 0,41 |
| 200 | Elu | 32 | 72 | 3757 | 0,21 | 0,54 |
| 200 | Relu | 64 | 75 | 5439 | 0,21 | 0,53 |
| 200 | Elu | 64 | 75 | 4176 | 0,21 | 0,53 |
| 200 | Relu | 128 | 63 | 3553 | 0,18 | 0,59 |
| 200 | Elu | 128 | 91 | 7867 | 0,26 | 0,45 |
| 300 | Relu | Whole | 54 | 2549 | 0,15 | 0,63 |
| 300 | Elu | Whole | 35 | 1422 | 0,10 | 0,73 |
| 300 | Relu | 16 | 110 | 11168 | 0,31 | 0,35 |
| 300 | Elu | 16 | 118 | 13031 | 0,34 | 0,31 |
| 300 | Relu | 32 | 56 | 1861 | 0,16 | 0,62 |
| 300 | Elu | 32 | 57 | 2870 | 0,16 | 0,62 |
| 300 | Relu | 64 | 83 | 5525 | 0,24 | 0,49 |
| 300 | Elu | 64 | 32 | 1087 | 0,09 | 0,74 |
| 300 | Relu | 12 | 8 100 | 9532 | 0,29 | 0,40 |
| 300 | Elu | 128 | 101 | 9474 | 0,29 | 0,40 |

**Table 5.4** 4-Layer Grid Search Results

| Node | A.Func. | Batch | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|---|---|
| 100 | Relu | Whole | 28 | 675 | 0,08 | 0,76 |
| 100 | Elu | Whole | 55 | 2847 | 0,16 | 0,63 |
| 100 | Relu | 16 | 30 | 724 | 0,09 | 0,75 |
| 100 | Elu | 16 | 30 | 508 | 0,09 | 0,75 |
| 100 | Relu | 32 | 76 | 4960 | 0,22 | 0,52 |
| 100 | Elu | 32 | 36 | 707 | 0,10 | 0,72 |
| 100 | Relu | 64 | 60 | 3024 | 0,17 | 0,60 |
| 100 | Elu | 64 | 26 | 712 | 0,07 | 0,77 |
| 100 | Relu | 128 | 89 | 7288 | 0,25 | 0,46 |
| 100 | Elu | 128 | 83 | 6007 | 0,24 | 0,49 |
| 200 | Relu | Whole | 63 | 2940 | 0,18 | 0,59 |
| 200 | Elu | Whole | 63 | 2847 | 0,18 | 0,59 |
| 200 | Relu | 16 | 61 | 2940 | 0,17 | 0,60 |
| 200 | Elu | 16 | 77 | 4729 | 0,22 | 0,52 |
| 200 | Relu | 32 | 57 | 2598 | 0,16 | 0,62 |
| 200 | Elu | 32 | 86 | 6828 | 0,25 | 0,47 |
| 200 | Relu | 64 | 25 | 652 | 0,07 | 0,78 |
| 200 | Elu | 64 | 82 | 6500 | 0,23 | 0,49 |
| 200 | Relu | 128 | 99 | 8795 | 0,28 | 0,41 |
| 200 | Relu | 128 | 99 | 8795 | 0,28 | 0,41 |
| 300 | Relu | Whole | 90 | 7695 | 0,26 | 0,45 |
| 300 | Elu | Whole | 49 | 2186 | 0,14 | 0,66 |
| 300 | Relu | 16 | 33 | 619 | 0,09 | 0,74 |
| 300 | Elu | 16 | 77 | 4918 | 0,22 | 0,52 |
| 300 | Relu | 32 | 69 | 3820 | 0,20 | 0,56 |
| 300 | Elu | 32 | 26 | 452 | 0,07 | 0,77 |
| 300 | Relu | 64 | 52 | 2101 | 0,15 | 0,64 |
| 300 | Elu | 64 | 67 | 4078 | 0,19 | 0,57 |
| 300 | Relu | 128 | 33 | 931 | 0,09 | 0,74 |
| 300 | Elu | 128 | 95 | 7978 | 0,27 | 0,43 |

mined for the PCA dataset. Table 5.5, 5.6, 5.7 and 5.8 show the grid search results obtained from the PCA dataset. It is noticed from the results that lower bound for achieved MAPE has decreased, whereas upper bound for achieved MAPE has increased.

Table 5.5 shows the grid search applied on single layer network. Differently from the original dataset, achieved MAPE is significantly high in single layer networks. MAPE of 0,19 is reached. Nevertheless, $R^2$ values are still lower in single layer networks. Elu is the best performing activation function. Optimal batch sizes for single layer networks are 32 and 64. Table 5.6 shows the grid search results for 2-layer networks. The lowest MAPE achieved is 0.13, which is an improvement compared to single layer networks. Differently from the single layer networks, Relu is the best activation function for these networks. The best batch sizes are whole, 16 and 64. Table 5.7 shows the grid search results for 3-layer networks. It is noticed that MAPE reaches as low as 0.07 in these models, while $R^2$ value is as high as 0.77. It is an improvement compared to 2-layer PCA networks and 3-layer original dataset networks. Relu is the best performing activation function in these models, and ideal batch size is 32.

Lastly, table 5.8 indicates the grid search results for 4-layer networks. The lowest MAPE in all networks are achieved in the 4-layer PCA datasets with 0.06 as well as the highest $R^2$ value with 0,8. 4-layer PCA networks are suitable in terms of both the accuracy and the explained variance. Relu is the best activation function to be used in 4-layer networks and ideal batch size is 64.

By using extra trees regressor method, also a feature selection dataset is generated. Table 5.9, 5.10, 5.11 and 5.12 indicate the results of grid search algorithm run on this dataset in order to determine the best operating conditions for the neural networks. It is noticed from these results that although the lower bound for MAPE has increased, the upper bound is decreased, which is a behavior in contrast within the PCA dataset. Table 5.9 shows the grid search results for single layer networks. The

**Table 5.5** Single Layer Grid Search Results on PCA Dataset

| Node | A.Func. | Batch | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|---|---|
| 100 | Relu | Whole | 211 | 43145 | 0,60 | 0,47 |
| 100 | Elu | Whole | 192 | 35762 | 0,55 | 0,43 |
| 100 | Relu | 16 | 136 | 15711 | 0,39 | 0,30 |
| 100 | Elu | 16 | 98 | 9043 | 0,28 | 0,22 |
| 100 | Relu | 32 | 65 | 3598 | 0,19 | 0,14 |
| 100 | Elu | 32 | 81 | 5155 | 0,23 | 0,18 |
| 100 | Relu | 64 | 67 | 3923 | 0,19 | 0,15 |
| 100 | Elu | 64 | 82 | 5115 | 0,23 | 0,18 |
| 100 | Relu | 128 | 115 | 12408 | 0,33 | 0,26 |
| 100 | Elu | 128 | 107 | 9261 | 0,31 | 0,24 |
| 200 | Relu | Whole | 112 | 10521 | 0,32 | 0,25 |
| 200 | Elu | Whole | 147 | 20312 | 0,42 | 0,33 |
| 200 | Relu | 16 | 204 | 40945 | 0,58 | 0,45 |
| 200 | Elu | 16 | 164 | 24669 | 0,47 | 0,36 |
| 200 | Relu | 32 | 206 | 39902 | 0,59 | 0,46 |
| 200 | Elu | 32 | 152 | 20130 | 0,43 | 0,34 |
| 200 | Relu | 64 | 197 | 36460 | 0,56 | 0,44 |
| 200 | Elu | 64 | 201 | 39799 | 0,57 | 0,45 |
| 200 | Relu | 128 | 188 | 34373 | 0,54 | 0,42 |
| 200 | Elu | 128 | 122 | 14430 | 0,35 | 0,27 |
| 300 | Relu | Whole | 224 | 4882 | 0 0,64 | 0,50 |
| 300 | Elu | Whole | 91 | 7566 | 0,26 | 0,20 |
| 300 | Relu | 16 | 207 | 41764 | 0,59 | 0,46 |
| 300 | Elu | 16 | 226 | 48389 | 0,65 | 0,50 |
| 300 | Relu | 32 | 208 | 42258 | 0,59 | 0,46 |
| 300 | Elu | 32 | 95 | 8171 | 0,27 | 0,21 |
| 300 | Relu | 64 | 187 | 33333 | 0,53 | 0,42 |
| 300 | Elu | 64 | 215 | 45447 | 0,61 | 0,48 |
| 300 | Relu | 128 | 139 | 17273 | 0,40 | 0,31 |
| 300 | Elu | 128 | 195 | 35288 | 0,56 | 0,43 |

**Table 5.6** 2-Layer Grid Search Results on PCA Dataset

| Node | A.Func. | Batch | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|---|---|
| 100 | Relu | Whole | 80 | 5272 | 0,23 | 0,32 |
| 100 | Elu | Whole | 68 | 3449 | 0,19 | 0,27 |
| 100 | Relu | 16 | 151 | 21244 | 0,43 | 0,60 |
| 100 | Elu | 16 | 198 | 36781 | 0,57 | 0,79 |
| 100 | Relu | 32 | 152 | 20886 | 0,43 | 0,61 |
| 100 | Elu | 32 | 71 | 4056 | 0,20 | 0,28 |
| 100 | Relu | 64 | 175 | 29127 | 0,50 | 0,70 |
| 100 | Elu | 64 | 97 | 9110 | 0,28 | 0,39 |
| 100 | Relu | 128 | 188 | 34012 | 0,54 | 0,75 |
| 100 | Elu | 128 | 159 | 24469 | 0,45 | 0,64 |
| 200 | Relu | Whole | 103 | 8609 | 0,29 | 0,41 |
| 200 | Elu | Whole | 129 | 16003 | 0,37 | 0,52 |
| 200 | Relu | 16 | 75 | 4596 | 0,21 | 0,30 |
| 200 | Elu | 16 | 101 | 9469 | 0,29 | 0,40 |
| 200 | Relu | 32 | 65 | 1871 | 0,19 | 0,26 |
| 200 | Elu | 32 | 89 | 7087 | 0,25 | 0,36 |
| 200 | Relu | 64 | 113 | 11757 | 0,32 | 0,45 |
| 200 | Elu | 64 | 64 | 3207 | 0,18 | 0,26 |
| 200 | Relu | 128 | 87 | 7123 | 0,25 | 0,35 |
| 200 | Elu | 128 | 164 | 24693 | 0,47 | 0,66 |
| 300 | Relu | Whole | 101 | 9019 | 0,29 | 0,40 |
| 300 | Elu | Whole | 139 | 16847 | 0,40 | 0,56 |
| 300 | Relu | 16 | 54 | 1080 | 0,15 | 0,22 |
| 300 | Elu | 16 | 78 | 3671 | 0,22 | 0,31 |
| 300 | Relu | 32 | 72 | 2709 | 0,21 | 0,29 |
| 300 | Elu | 32 | 80 | 6117 | 0,23 | 0,32 |
| 300 | Relu | 64 | 47 | 2720 | 0,13 | 0,19 |
| 300 | Elu | 64 | 185 | 33201 | 0,53 | 0,74 |
| 300 | Relu | 128 | 185 | 32916 | 0,53 | 0,74 |
| 300 | Elu | 128 | 154 | 22095 | 0,44 | 0,62 |

**Table 5.7** 3-Layer Grid Search Results on PCA Dataset

| Node | A.Func. | Batch | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|---|---|
| 100 | Relu | Whole | 113 | 11580 | 0,32 | 0,34 |
| 100 | Elu | Whole | 26 | 1481 | 0,07 | 0,77 |
| 100 | Relu | 16 | 116 | 12585 | 0,33 | 0,32 |
| 100 | Elu | 16 | 137 | 17437 | 0,39 | 0,22 |
| 100 | Relu | 32 | 61 | 3357 | 0,17 | 0,60 |
| 100 | Elu | 32 | 85 | 5757 | 0,24 | 0,48 |
| 100 | Relu | 64 | 78 | 5537 | 0,22 | 0,51 |
| 100 | Elu | 64 | 123 | 14031 | 0,35 | 0,29 |
| 100 | Relu | 128 | 69 | 3914 | 0,20 | 0,56 |
| 100 | Elu | 128 | 134 | 16586 | 0,38 | 0,23 |
| 200 | Relu | Whole | 129 | 15763 | 0,37 | 0,26 |
| 200 | Elu | Whole | 129 | 15661 | 0,37 | 0,26 |
| 200 | Relu | 16 | 84 | 6800 | 0,24 | 0,48 |
| 200 | Elu | 16 | 146 | 20766 | 0,42 | 0,17 |
| 200 | Relu | 32 | 155 | 23265 | 0,44 | 0,13 |
| 200 | Elu | 32 | 56 | 2599 | 0,16 | 0,62 |
| 200 | Relu | 64 | 43 | 859 | 0,12 | 0,69 |
| 200 | Elu | 64 | 49 | 1813 | 0,14 | 0,66 |
| 200 | Relu | 128 | 30 | 1492 | 0,09 | 0,75 |
| 200 | Elu | 128 | 55 | 2625 | 0,16 | 0,63 |
| 300 | Relu | Whole | 49 | 1349 | 0,14 | 0,66 |
| 300 | Elu | Whole | 128 | 15664 | 0,37 | 0,26 |
| 300 | Relu | 16 | 70 | 4426 | 0,20 | 0,55 |
| 300 | Elu | 16 | 135 | 17864 | 0,39 | 0,23 |
| 300 | Relu | 32 | 26 | 526 | 0,07 | 0,77 |
| 300 | Elu | 32 | 72 | 4640 | 0,21 | 0,54 |
| 300 | Relu | 64 | 106 | 10584 | 0,30 | 0,37 |
| 300 | Elu | 64 | 33 | 490 | 0,09 | 0,74 |
| 300 | Relu | 128 | 64 | 3727 | 0,18 | 0,58 |
| 300 | Elu | 128 | 74 | 4458 | 0,21 | 0,53 |

**Table 5.8** 4-Layer Grid Search Results on PCA Dataset

| Node | A.Func. | Batch | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|---|---|
| 100 | Relu | Whole | 61 | 3583 | 0,17 | 0,60 |
| 100 | Elu | Whole | 87 | 6514 | 0,25 | 0,47 |
| 100 | Relu | 16 | 63 | 3020 | 0,18 | 0,59 |
| 100 | Elu | 16 | 75 | 4873 | 0,21 | 0,53 |
| 100 | Relu | 32 | 90 | 7982 | 0,26 | 0,45 |
| 100 | Elu | 32 | 114 | 12171 | 0,33 | 0,33 |
| 100 | Relu | 64 | 57 | 2815 | 0,16 | 0,62 |
| 100 | Elu | 64 | 138 | 17930 | 0,39 | 0,21 |
| 100 | Relu | 128 | 90 | 7357 | 0,26 | 0,45 |
| 100 | Elu | 128 | 66 | 3894 | 0,19 | 0,57 |
| 200 | Relu | Whole | 76 | 5449 | 0,22 | 0,52 |
| 200 | Elu | Whole | 73 | 4181 | 0,21 | 0,54 |
| 200 | Relu | 16 | 68 | 3978 | 0,19 | 0,56 |
| 200 | Elu | 16 | 96 | 8898 | 0,27 | 0,42 |
| 200 | Relu | 32 | 20 | 452 | 0,06 | 0,80 |
| 200 | Elu | 32 | 126 | 15227 | 0,36 | 0,27 |
| 200 | Relu | 64 | 47 | 1281 | 0,13 | 0,67 |
| 200 | Elu | 64 | 38 | 961 | 0,11 | 0,71 |
| 200 | Relu | 128 | 35 | 651 | 0,10 | 0,73 |
| 200 | Elu | 128 | 50 | 1675 | 0,14 | 0,65 |
| 300 | Relu | Whole | 75 | 4684 | 0,21 | 0,53 |
| 300 | Elu | Whole | 61 | 2875 | 0,17 | 0,60 |
| 300 | Relu | 16 | 85 | 6454 | 0,24 | 0,48 |
| 300 | Elu | 16 | 108 | 11039 | 0,31 | 0,36 |
| 300 | Relu | 32 | 105 | 10108 | 0,30 | 0,38 |
| 300 | Elu | 32 | 73 | 5089 | 0,21 | 0,54 |
| 300 | Relu | 64 | 84 | 6350 | 0,24 | 0,48 |
| 300 | Elu | 64 | 20 | 403 | 0,06 | 0,80 |
| 300 | Relu | 128 | 66 | 3983 | 0,19 | 0,57 |
| 300 | Elu | 128 | 137 | 18678 | 0,39 | 0,22 |

lowest MAPE acquired is around 0.27 and $R^2$ value is low similar to the previously built single layer networks. Relu dominates the other activation functions in this dataset. Table 5.10 shows the grid search results for 2-layer networks. There is not a significant increase in MAPE, and $R^2$ value is around 0.3. Relu is still the most fitting activation function. Table 5.11 shows the grid search results for 3-layer networks, and MAPE decreases up to 0.14 in these networks. In a similar manner, 3-layer networks have better $R^2$ values in this dataset too. Differently, Elu is the best fitting function for this dataset. Finally, table 5.12 shows the grid search results for 4-layer networks. Results that are obtained from feature selection dataset are not succesful compared to the previous datasets. Nevertheless, it still manages to reach the MAPE of 0,1 and $R^2$ value of 0,72.

Table 5.13 is a collection of best models on layer-basis from the original dataset. It is observed that in general, Elu is more successful in the original dataset and 64 batch size is ideal. $R^2$ value increases significantly within the layer size. Additionally, MAPE is decreased within each layer. Table 5.14 and 5.15 are prepared in similar manner. Table 5.16 indicated the results of the other machine learning models. It shows that Random forest is better than SVM and KNN, and it has a MAPE around 0.26 at best.

**Table 5.9** 1-Layer Grid Search Results on Feature Selection Dataset

| Node | A.Func. | Batch | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|---|---|
| 100 | Relu | Whole | 137 | 17843 | 0,39 | 0,30 |
| 100 | Elu | Whole | 131 | 14622 | 0,37 | 0,29 |
| 100 | Relu | 16 | 114 | 10342 | 0,33 | 0,25 |
| 100 | Elu | 16 | 125 | 15044 | 0,36 | 0,28 |
| 100 | Relu | 32 | 102 | 9375 | 0,29 | 0,23 |
| 100 | Elu | 32 | 134 | 17381 | 0,38 | 0,30 |
| 100 | Relu | 64 | 142 | 19056 | 0,41 | 0,32 |
| 100 | Elu | 64 | 101 | 9101 | 0,29 | 0,22 |
| 100 | Relu | 128 | 126 | 13571 | 0,36 | 0,28 |
| 100 | Elu | 128 | 113 | 10454 | 0,32 | 0,25 |
| 200 | Relu | Whole | 132 | 14986 | 0,38 | 0,29 |
| 200 | Elu | Whole | 126 | 13444 | 0,36 | 0,28 |
| 200 | Relu | 16 | 130 | 14926 | 0,37 | 0,29 |
| 200 | Elu | 16 | 108 | 10127 | 0,31 | 0,24 |
| 200 | Relu | 32 | 130 | 16162 | 0,37 | 0,29 |
| 200 | Elu | 32 | 104 | 9310 | 0,30 | 0,23 |
| 200 | Relu | 64 | 114 | 12272 | 0,33 | 0,25 |
| 200 | Elu | 64 | 150 | 20655 | 0,43 | 0,33 |
| 200 | Relu | 128 | 133 | 16572 | 0,38 | 0,30 |
| 200 | Elu | 128 | 128 | 15121 | 0,37 | 0,28 |
| 300 | Relu | Whole | 100 | 7805 | 0,29 | 0,22 |
| 300 | Elu | Whole | 146 | 19347 | 0,42 | 0,32 |
| 300 | Relu | 16 | 139 | 16743 | 0,40 | 0,31 |
| 300 | Elu | 16 | 126 | 13332 | 0,36 | 0,28 |
| 300 | Relu | 32 | 101 | 7698 | 0,29 | 0,22 |
| 300 | Elu | 32 | 108 | 10375 | 0,31 | 0,24 |
| 300 | Relu | 64 | 145 | 20396 | 0,41 | 0,32 |
| 300 | Elu | 64 | 145 | 19767 | 0,41 | 0,32 |
| 300 | Relu | 128 | 128 | 15529 | 0,37 | 0,28 |
| 300 | Elu | 128 | 106 | 10579 | 0,30 | 0,24 |

**Table 5.10** 2-Layer Grid Search Results on Feature Selection Dataset

| Node | A.Func. | Batch | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|---|---|
| 100 | Relu | Whole | 81 | 4696 | 0,23 | 0,32 |
| 100 | Elu | Whole | 94 | 8573 | 0,27 | 0,38 |
| 100 | Relu | 16 | 76 | 3819 | 0,22 | 0,30 |
| 100 | Elu | 16 | 85 | 6082 | 0,24 | 0,34 |
| 100 | Relu | 32 | 91 | 7781 | 0,26 | 0,36 |
| 100 | Elu | 32 | 70 | 2595 | 0,20 | 0,28 |
| 100 | Relu | 64 | 90 | 6000 | 0,26 | 0,36 |
| 100 | Elu | 64 | 71 | 2916 | 0,20 | 0,28 |
| 100 | Relu | 128 | 93 | 7225 | 0,27 | 0,37 |
| 100 | Elu | 128 | 98 | 7602 | 0,28 | 0,39 |
| 200 | Relu | Whole | 75 | 3444 | 0,21 | 0,30 |
| 200 | Elu | Whole | 94 | 7601 | 0,27 | 0,38 |
| 200 | Relu | 16 | 78 | 4594 | 0,22 | 0,31 |
| 200 | Elu | 16 | 84 | 5485 | 0,24 | 0,34 |
| 200 | Relu | 32 | 79 | 5615 | 0,23 | 0,32 |
| 200 | Elu | 32 | 100 | 8844 | 0,29 | 0,40 |
| 200 | Relu | 64 | 87 | 7151 | 0,25 | 0,35 |
| 200 | Elu | 64 | 71 | 2954 | 0,20 | 0,28 |
| 200 | Relu | 128 | 74 | 3342 | 0,21 | 0,30 |
| 200 | Elu | 128 | 82 | 5558 | 0,23 | 0,33 |
| 300 | Relu | Whole | 100 | 8302 | 0,29 | 0,40 |
| 300 | Elu | Whole | 81 | 4294 | 0,23 | 0,32 |
| 300 | Relu | 16 | 99 | 7685 | 0,28 | 0,40 |
| 300 | Elu | 16 | 77 | 4982 | 0,22 | 0,31 |
| 300 | Relu | 32 | 81 | 4759 | 0,23 | 0,32 |
| 300 | Elu | 32 | 97 | 8304 | 0,28 | 0,39 |
| 300 | Relu | 64 | 97 | 7036 | 0,28 | 0,39 |
| 300 | Elu | 64 | 100 | 8210 | 0,29 | 0,40 |
| 300 | Relu | 128 | 77 | 4680 | 0,22 | 0,31 |
| 300 | Elu | 128 | 89 | 7562 | 0,25 | 0,36 |

**Table 5.11** 3-Layer Grid Search Results on Feature Selection Dataset

| Node | A.Func. | Batch | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|---|---|
| 100 | Relu | Whole | 69 | 3262 | 0,20 | 0,56 |
| 100 | Elu | Whole | 65 | 3715 | 0,19 | 0,58 |
| 100 | Relu | 16 | 52 | 2103 | 0,15 | 0,64 |
| 100 | Elu | 16 | 61 | 3536 | 0,17 | 0,60 |
| 100 | Relu | 32 | 58 | 2212 | 0,17 | 0,61 |
| 100 | Elu | 32 | 67 | 4244 | 0,19 | 0,57 |
| 100 | Relu | 64 | 66 | 4241 | 0,19 | 0,57 |
| 100 | Elu | 64 | 70 | 3536 | 0,20 | 0,55 |
| 100 | Relu | 128 | 60 | 2386 | 0,17 | 0,60 |
| 100 | Elu | 128 | 70 | 4156 | 0,20 | 0,55 |
| 200 | Relu | Whole | 66 | 3645 | 0,19 | 0,57 |
| 200 | Elu | Whole | 73 | 4589 | 0,21 | 0,54 |
| 200 | Relu | 16 | 56 | 2045 | 0,16 | 0,62 |
| 200 | Elu | 16 | 64 | 3780 | 0,18 | 0,58 |
| 200 | Relu | 32 | 69 | 3844 | 0,20 | 0,56 |
| 200 | Elu | 32 | 64 | 3964 | 0,18 | 0,58 |
| 200 | Relu | 64 | 54 | 2600 | 0,15 | 0,63 |
| 200 | Elu | 64 | 54 | 1590 | 0,15 | 0,63 |
| 200 | Relu | 128 | 65 | 3616 | 0,19 | 0,58 |
| 200 | Elu | 128 | 51 | 1987 | 0,15 | 0,65 |
| 300 | Relu | Whole | 56 | 1704 | 0,16 | 0,62 |
| 300 | Elu | Whole | 75 | 4454 | 0,21 | 0,53 |
| 300 | Relu | 16 | 72 | 4480 | 0,21 | 0,54 |
| 300 | Elu | 16 | 64 | 3341 | 0,18 | 0,58 |
| 300 | Relu | 32 | 59 | 2560 | 0,17 | 0,61 |
| 300 | Relu | 64 | 70 | 3976 | 0,20 | 0,55 |
| 300 | Elu | 64 | 60 | 3296 | 0,17 | 0,60 |
| 300 | Relu | 128 | 50 | 2178 | 0,14 | 0,65 |
| 300 | Elu | 128 | 60 | 3267 | 0,17 | 0,60 |

**Table 5.12** 4-Layer Grid Search Results on Feature Selection Dataset

| Node | A.Func. | Batch | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|---|---|
| 100 | Relu | Whole | 58 | 1961 | 0,17 | 0,61 |
| 100 | Elu | Whole | 36 | 1301 | 0,10 | 0,72 |
| 100 | Relu | 16 | 56 | 2990 | 0,16 | 0,62 |
| 100 | Elu | 16 | 41 | 1322 | 0,12 | 0,70 |
| 100 | Relu | 32 | 61 | 3077 | 0,17 | 0,60 |
| 100 | Elu | 32 | 49 | 1900 | 0,14 | 0,66 |
| 100 | Relu | 64 | 55 | 2703 | 0,16 | 0,63 |
| 100 | Elu | 64 | 54 | 2351 | 0,15 | 0,63 |
| 100 | Relu | 128 | 36 | 1076 | 0,10 | 0,72 |
| 100 | Elu | 128 | 55 | 2513 | 0,16 | 0,63 |
| 200 | Relu | Whole | 40 | 2249 | 0,11 | 0,70 |
| 200 | Elu | Whole | 55 | 1758 | 0,16 | 0,63 |
| 200 | Relu | 16 | 52 | 2236 | 0,15 | 0,64 |
| 200 | Elu | 16 | 55 | 2687 | 0,16 | 0,63 |
| 200 | Relu | 32 | 62 | 2501 | 0,18 | 0,59 |
| 200 | Elu | 32 | 42 | 1559 | 0,12 | 0,69 |
| 200 | Relu | 64 | 61 | 3457 | 0,17 | 0,60 |
| 200 | Elu | 64 | 37 | 655 | 0,11 | 0,72 |
| 200 | Relu | 128 | 40 | 1380 | 0,11 | 0,70 |
| 200 | Elu | 128 | 55 | 2154 | 0,16 | 0,63 |
| 300 | Relu | Whole | 62 | 2968 | 0,18 | 0,59 |
| 300 | Elu | Whole | 62 | 3326 | 0,18 | 0,59 |
| 300 | Relu | 16 | 62 | 3072 | 0,18 | 0,59 |
| 300 | Elu | 16 | 35 | 1010 | 0,10 | 0,73 |
| 300 | Relu | 32 | 45 | 1439 | 0,13 | 0,68 |
| 300 | Elu | 32 | 47 | 1433 | 0,13 | 0,67 |
| 300 | Relu | 64 | 36 | 740 | 0,10 | 0,72 |
| 300 | Relu | 128 | 53 | 2298 | 0,15 | 0,64 |
| 300 | Elu | 128 | 43 | 1550 | 0,12 | 0,69 |

**Table 5.13** Best Models in Original Dataset

| Layer | Node | A.Func. | Batch | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|---|---|---|
| 1 | 100 | Elu | 64 | 80 | 4980 | 0,23 | 0,18 |
| 1 | 200 | Elu | 64 | 82 | 4072 | 0,23 | 0,18 |
| 1 | 200 | Relu | 64 | 89 | 5234 | 0,25 | 0,20 |
| 1 | 100 | Elu | 16 | 103 | 7619 | 0,29 | 0,23 |
| 1 | 200 | Elu | 32 | 101 | 9269 | 0,29 | 0,22 |
| 2 | 100 | Elu | 16 | 50 | 1669 | 0,14 | 0,20 |
| 2 | 300 | Elu | 64 | 52 | 2852 | 0,15 | 0,21 |
| 2 | 300 | Elu | 32 | 56 | 1143 | 0,16 | 0,22 |
| 2 | 200 | Relu | Whole | 66 | 3752 | 0,19 | 0,26 |
| 2 | 100 | Elu | 64 | 74 | 5062 | 0,21 | 0,30 |
| 3 | 300 | Elu | 64 | 32 | 1087 | 0,09 | 0,74 |
| 3 | 100 | Elu | 64 | 36 | 7169 | 0,10 | 0,72 |
| 3 | 300 | Elu | 32 | 57 | 2870 | 0,16 | 0,62 |
| 3 | 300 | Relu | 32 | 56 | 1861 | 0,16 | 0,62 |
| 3 | 100 | Elu | Whole | 56 | 2008 | 0,16 | 0,62 |
| 4 | 100 | Relu | Whole | 28 | 675 | 0,08 | 0,76 |
| 4 | 300 | Elu | 32 | 26 | 452 | 0,07 | 0,77 |
| 4 | 200 | Relu | 64 | 25 | 652 | 0,07 | 0,78 |
| 4 | 100 | Relu | 16 | 30 | 724 | 0,09 | 0,75 |
| 4 | 100 | Elu | 16 | 30 | 508 | 0,09 | 0,75 |

**Table 5.14** Best Models in PCA Dataset

| Layer | Node | A.Func. | Batch | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|---|---|---|
| 1 | 100 | Relu | 32 | 65 | 3598 | 0,19 | 0,14 |
| 1 | 100 | Relu | 64 | 67 | 3923 | 0,19 | 0,15 |
| 1 | 100 | Elu | 32 | 81 | 5155 | 0,23 | 0,18 |
| 1 | 100 | Elu | 64 | 67 | 3923 | 0,19 | 0,15 |
| 1 | 300 | Elu | 32 | 95 | 817 | 0,27 | 0,21 |
| 2 | 300 | Relu | 16 | 54 | 1080 | 0,15 | 0,22 |
| 2 | 100 | Elu | Whole | 68 | 3449 | 0,19 | 0,27 |
| 2 | 200 | Relu | 32 | 65 | 1871 | 0,19 | 0,26 |
| 2 | 200 | Relu | 64 | 64 | 3207 | 0,18 | 0,26 |
| 2 | 300 | Relu | 64 | 47 | 2720 | 0,13 | 0,19 |
| 3 | 100 | Elu | Whole | 26 | 1481 | 0,07 | 0,77 |
| 3 | 300 | Relu | 64 | 33 | 490 | 0,09 | 0,74 |
| 3 | 300 | Relu | 32 | 26 | 526 | 0,07 | 0,77 |
| 3 | 200 | Relu | 64 | 43 | 859 | 0,12 | 0,69 |
| 3 | 200 | Elu | 32 | 56 | 2599 | 0,16 | 0,62 |
| 4 | 300 | Elu | 64 | 20 | 403 | 0,06 | 0,8 |
| 4 | 200 | Relu | 32 | 20 | 452 | 0,06 | 0,8 |
| 4 | 200 | Relu | 128 | 35 | 651 | 0,10 | 0,73 |
| 4 | 200 | Elu | 64 | 38 | 961 | 0,11 | 0,71 |
| 4 | 200 | Relu | 64 | 38 | 1281 | 0,13 | 0,67 |

**Table 5.15** Best Models in Feature Selection Dataset

| Layer | Node | A.Func. | Batch | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|---|---|---|
| 1 | 300 | Relu | Whole | 100 | 7805 | 0,29 | 0,22 |
| 1 | 100 | Relu | 32 | 102 | 9375 | 0.29 | 0.23 |
| 1 | 300 | Relu | 32 | 101 | 7698 | 0.29 | 0.22 |
| 1 | 300 | Relu | 128 | 128 | 15529 | 0,27 | 0,28 |
| 1 | 300 | Elu | 128 | 106 | 10579 | 0,30 | 0,24 |
| 2 | 100 | Elu | 32 | 70 | 2595 | 0,20 | 0,28 |
| 2 | 200 | Elu | 64 | 71 | 2954 | 0,20 | 0,28 |
| 2 | 200 | Relu | Whole | 75 | 3444 | 0,21 | 0,3 |
| 2 | 200 | Relu | 16 | 78 | 4594 | 0,22 | 0,31 |
| 2 | 200 | Relu | 32 | 79 | 5615 | 0,23 | 0,32 |
| 3 | 300 | Relu | 128 | 50 | 2178 | 0,14 | 0,65 |
| 3 | 300 | Elu | 128 | 60 | 3267 | 0,17 | 0,6 |
| 3 | 300 | Elu | 64 | 60 | 3296 | 0,17 | 0,6 |
| 3 | 200 | Relu | 64 | 54 | 2600 | 0,15 | 0,63 |
| 3 | 200 | Elu | 64 | 54 | 1590 | 0,15 | 0,63 |
| 4 | 100 | Relu | Whole | 36 | 1301 | 0,10 | 0,72 |
| 4 | 100 | Relu | 128 | 36 | 1076 | 0,10 | 0,72 |
| 4 | 200 | Relu | Whole | 40 | 2249 | 0,11 | 0,7 |
| 4 | 200 | Elu | 64 | 37 | 655 | 0,11 | 0,72 |
| 4 | 200 | Relu | 128 | 40 | 1380 | 0,16 | 0,63 |

**Table 5.16** Results of Other Machine Learning Models

| Model | MAE | MSE | MAPE | $R^2$ |
|---|---|---|---|---|
| RF - 100 | 140 | 11685 | 0,36 | 0,28 |
| RF - 200 | 135 | 10329 | 0,32 | 0,26 |
| RF - 400 | 110 | 9987 | 0,26 | 0,34 |
| RF - 800 | 130 | 14294 | 0,28 | 0,25 |
| RF - 1000 | 125 | 13284 | 0,28 | 0,22 |
| SVM - Linear | 198 | 32784 | 0,51 | 0,11 |
| SVM - RBF | 165 | 27888 | 0,39 | 0,25 |
| SVM - 2 | 188 | 35192 | 0,45 | 0,16 |
| SVM - 3 | 182 3 | 3782 | 0,42 | 0,15 |
| SVM - 4 | 175 | 31167 | 0,44 | 0,12 |
| SVM - 5 | 172 | 26710 | 0,46 | 0,15 |
| SVM - 6 | 176 | 29112 | 0,41 | 0,14 |
| KNN - 1 | 195 | 34901 | 0,53 | 0,12 |
| KNN - 3 | 179 | 29128 | 0,47 | 0,2 |
| KNN - 5 | 175 | 26721 | 0,44 | 0,19 |
| KNN - 7 | 162 | 24272 | 0,39 | 0,21 |
| KNN - 9 | 155 | 22156 | 0,35 | 0,24 |

# 6. CONCLUSIONS

In this study, a novel approach to very-short term solar irradiance forecasting is taken. Sliding windows methodology is integrated into model by applying all of the features, and wind direction is also used as a categorical measure. 15-minute ahead is directly forecasted from the data with 1-minute resolution. During the forecasting phase, no future features are used, as the main aim of the study is to built a regular time series method from Supervised Machine Learning methods. Feature selection and extraction methods are used in order to enhance the accuracy of the built models. As the result of the study, 0.06 MAPE is reached within the 4-layer networks combined within the Principal Component Analysis as the dimensionality reduction method. This thesis additionally indicates that integration of sliding windows method provides a robust framework to the solar irradiance prediction problems, as their performance are not negatively influenced by taking the lag number as 15.

Further studies can be accomplished in order to increase the performance of the mentioned models. One possible improvement is to select a better hyper parameter optimization method, rather than random search or grid search. As mentioned in chapter 2, response surface methodology can be integrated in order to overcome this problem.

# REFERENCES


Ahmadi, M., Vahabzadeh, F., Bonakdarpour, B., Mofarrah, E., Mehranian, M. (2005). Application of the central composite design and response surface methodology to the advanced treatment of olive oil processing wastewater using Fenton's peroxidation. Journal of Hazardous Materials, 123(1-3), 187-195.

Amjady, N. (2007). Short-term bus load forecasting of power systems by a new 21 hybrid method. IEEE Transactions on Power Systems, 22(1), 333-341.

Azadeh, A., Saberi, M., Ghaderi, S. F., Gitiforouz, A., Ebrahimipour, V. (2008). Improved estimation of elec- tricity demand function by integration of fuzzy system and data mining approach. Energy Conversion and Man- agement, 49(8), 2165-2177.

Barlow, H. B. (1989). Unsupervised learning. Neural computation, 1(3), 295- 311. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Bergstra, J., Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb), 281-305.

Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U. (1999, January). When is nearest neighbor meaningful?. In International conference on database theory (pp. 217-235). Springer, Berlin, Heidelberg.

Bezerra, M. A., Santelli, R. E., Oliveira, E. P., Villar, L. S., Escaleira, L. A. (2008). Response surface methodology (RSM) as a tool for optimization in analytical chemistry. Talanta, 76(5), 965-977.

Bouzgou, H., Gueymard, C. A. (2017). Minimum redundancymaximum rel- evance with extreme learning machines for global solar radiation forecasting: Toward an optimized dimensionality reduction for solar time series. Solar Energy, 158, 595-609.

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Brown, T. A. (2014). Confirmatory factor analysis for applied research. Guilford Publications.

Chen, B. J., Chang, M. W. (2004). Load forecasting using support vector machines: A study on EUNITE competition 2001. IEEE transactions on power systems, 19(4), 1821-1830.

Deo, R. C., Şahin, M. (2017). Forecasting long-term global solar radiation withan ANN algorithm coupled with satellite-derived (MODIS) land surface temperature (LST) for regional locations in Queensland. Renewable and Sustainable Energy Reviews, 72, 828-848.

Espinar, B., Aznarte, J. L., Girard, R., Moussa, A. M., Kariniotakis, G. (2010, April). Photovoltaic Forecasting: A state of the art. In 5th European PV-Hybrid and Mini-Grid Conference (pp. Pages-250). OTTI-Ostbayerisches Technologie-Transfer-Institut.

Fan, J., Wu, L., Zhang, F., Cai, H., Zeng, W., Wang, X., Zou, H. (2019). Empirical and machine learning models for predicting daily global solar radiation from sunshine duration: A review and case study in China. Renewable and Sustainable Energy Reviews, 100, 186-212.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P. (1996, August). Knowledge Discovery and Data Mining: Towards a Unifying Framework. In KDD (Vol. 96, pp. 82-88).

Gala, Y., Fernandez, A ., D az, J., Dorronsoro, J. R. (2016). Hybrid machine learning forecasting of solar radiation values. Neurocomputing, 176, 48-59.

Gardner, M. W., Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron)a review of applications in the atmospheric sciences. Atmospheric environment, 32(14-15), 2627-2636.

Gönen, M., Alpaydı n, E. (2011). Multiple kernel learning algorithms. Journal of machine learning research, 12(Jul), 2211-2268.

Gigoni, L., Betti, A., Crisostomi, E., Franco, A., Tucci, M., Bizzarri, F., Mucci, D. (2018). Day-ahead hourly forecasting of power generation from photovoltaic plants. IEEE Transactions on Sustainable Energy, 9(2), 831-842.

Harman, H. H. (1960). Modern factor analysis.

Hill, W. J., Hunter, W. G. (1966). A review of response surface methodology: a literature survey. Technometrics, 8(4), 571-590.

Homadi, A. M. (2016). Effect of elevation and wind direction on silicon solar panel efficiency. World Academy of Science, Engineering and Technology International Journal of Energy and Power Engineering, 10(9).

Hinton, G. E. (2012). A practical guide to training restricted Boltzmann machines. In Neural networks: Tricks of the trade (pp. 599-619). Springer,

Berlin, Heidelberg.

Huertas Tato, J., Centeno Brito, M. (2019). Using Smart Persistence and Random Forests to Predict Photovoltaic Energy Production. Energies, 12(1), 100.

Ibrahim, I. A., Khatib, T. (2017). A novel hybrid model for hourly global solar radiation prediction using random forests technique and firefly algorithm. Energy Conversion and Management, 138, 413-425.

IEA, 2006a. Global energy technology perspectives. International Energy Agency, OECD Publication Service, OECD, Paris.

Inman, R. H., Pedro, H. T., Coimbra, C. F. (2013). Solar forecasting methods for renewable energy integration. Progress in energy and combustion science, 39(6), 535-576.

Ishik, M. Y., Göze, T., Özcan, I., Güng or, V. C ., Aydın, Z. (2015, April). Short term electricity load forecasting: A case study of electric utility market in Turkey. In 2015 3rd International Istanbul Smart Grid Congress and Fair(ICSG) (pp. 1-5). IEEE.

Khadem, M. (1993). Application of kohonen neural network classifier to short term load forecasting. In Panel Session on Application of Neural Networks to Short-term Load Forecasting, 1993 IEEE Winter Meeting.

Khuri, A. I., Mukhopadhyay, S. (2010). Response surface methodology. Wiley Interdisciplinary Reviews: Computational Statistics, 2(2), 128-149.

Lara-Fanego, V., Ruiz-Arias, J. A., Pozo-V azquez, D., Santos-Alamillos, F. J., Tovar- Pescador, J. (2012). Evaluation of the WRF model solar irradiance fore- casts in Andalusia (southern Spain). Solar Energy, 86(8), 2200-2217.

Lago, J., De Brabandere, K., De Ridder, F., De Schutter, B. (2018). Short-term forecasting of solar irradiance without local telemetry: A generalized model us- ing satellite data. Solar Energy, 173, 566-577.

Leva, S., Dolara, A., Grimaccia, F., Mussetta, M., Ogliari, E. (2017). Analy- sis and validation of 24 hours ahead neural network forecasting of photovoltaic output power. Mathematics and computers in simulation, 131, 88-100.

Lujan-Moreno, G. A., Howard, P. R., Rojas, O. G., Montgomery, D. C. (2018). Design of experiments and response surface methodology to tune machine learn- ing hyperparameters, with a random forest case-study. Expert Systems with Applications, 109, 195-205.

Luo, J., Hong, T., Fang, S. C. (2018). Robust Regression Models for Load Forecasting. IEEE Transactions on Smart Grid.

Man, D., Vision, A. (1982). A computational investigation into the human representation and processing of visual information.

Marzo, A., Trigo-Gonzalez, M., Alonso-Montesinos, J., Mart nez-Durb an, M., L opez, G., Ferrada, P., ... Batlles, F. J. (2017). Daily global solar radiation estimation in desert areas using daily extreme temperatures and extraterrestrial radiation. Renewable Energy, 113, 303-311.

Massidda, L., Marrocu, M. (2017). Use of Multilinear Adaptive Regression Splines and numerical weather prediction to forecast the power output of a PV plant in Borkum, Germany. Solar Energy, 146, 141-149.

Measurement and Instrumentation Data Center (n.d.), Oak Ridge National Laboratory Irradiance Data, retrieved from; https://midcdmz.nrel.gov/apps/sitehome.pl?site=ORNL

Meenal, R., Selvakumar, A. I. (2018). Assessment of SVM, empirical and ANN based solar radiation prediction models with most influencing input parameters. Renewable Energy, 121, 324-343.

Mellit, A., Pavan, A. M. (2010). A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected PV plant at Trieste, Italy. Solar Energy, 84(5), 807-821.

Panapakidis, I. P., Dagoumas, A. S. (2016). Day-ahead electricity price forecasting via the application of artificial neural network based models. Applied Energy, 172, 132-151.

Pedro, H. T., Coimbra, C. F., David, M., Lauret, P. (2018). Assessment of machine learning techniques for deterministic and probabilistic intra-hour solar forecasts. Renewable Energy, 123, 191-203.

Persson, C., Bacher, P., Shiga, T., Madsen, H. (2017). Multi-site solar power forecasting using gradient boosted regression trees. Solar Energy, 150, 423-436.

Pierro, M., Bucci, F., De Felice, M., Maggioni, E., Perotto, A., Spada, F., ... Cornaro, C. (2017). Deterministic and stochastic approaches for day-ahead so- lar power forecasting. Journal of Solar Energy Engineering, 139(2), 021010.

Qian, N., Sejnowski, T. J. (1988). Predicting the secondary structure of globu- lar proteins using neural network models. Journal of molecular biology, 202(4), 865-884.

Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

Remund, J., Perez, R., Lorenz, E. (2008, September). Comparison of solar radiation forecasts for the USA. In Proc. of the 23rd European PV Conference (Vol. 14).

Reno, M. J., Hansen, C. W., Stein, J. S. (2012). Global horizontal irradiance clear sky models: Implementation and analysis. SANDIA report SAND2012-2389.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386.

Rumelhart, D. E., Durbin, R., Golden, R., Chauvin, Y. (1995). Backpropagation: The basic theory. Backpropagation: Theory, architectures and applications, 1-34.

Safavian, S. R., Landgrebe, D. (1991). A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 21(3), 660-674.

Sejnowski, T. J., Rosenberg, C. R. (1987). Parallel networks that learn to pronounce English text. Complex systems, 1(1), 145-168.

Shah, A., Kadam, E., Shah, H., Shinde, S., Shingade, S. (2016). Deep residual networks with exponential linear unit. arXiv preprint arXiv:1604.04112.

Shakya, A., Michael, S., Saunders, C., Armstrong, D., Pandey, P., Chalise, S., Tonkoski, R. (2017). Solar irradiance forecasting in remote microgrids using Markov switching model. IEEE Transactions on Sustainable Energy, 8(3), 895- 905.

Sharma, N., Sharma, P., Irwin, D., Shenoy, P. (2011, October). Predicting solar generation from weather forecasts using machine learning. In 2011 IEEE international conference on smart grid communications (SmartGridComm) (pp. 528-533). IEEE.

Shi, J., Lee, W. J., Liu, Y., Yang, Y., Wang, P. (2012). Forecasting power out- put of photovoltaic systems based on weather classification and support vector machines. IEEE Transactions on Industry Applications, 48(3), 1064-1069.

Srivastava, S., Lessmann, S. (2018). A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data. Solar Energy, 162, 232-247.

Szuromi, P., Jasny, B., Clery, D., Austin, J., Hanson, B. (2007). Energy for the long haul. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288.

Torgo, L., Gama, J. (1996, October). Regression by classification. In Brazil- ian symposium on artificial intelligence (pp. 51-60). Springer, Berlin, Heidelberg.

Torres,J.F.,Troncoso,A.,Koprinska,I.,Wang,Z.,Mart nez-A lvarez,F.(2018, June). Deep learning for big data time series forecasting applied to solar power. In The 13th International Conference on Soft Computing Models in Industrial and Environmental Applications (pp. 123-133). Springer, Cham.

Torres-Barr an, A., Alonso, A ., Dorronsoro, J. R. (2019). Regression tree ensembles for wind energy and solar radiation prediction. Neurocomputing, 326, 151-160.

Vapnik, V. (2013). The nature of statistical learning theory. Springer science business media.

Voyant, C., Paoli, C., Muselli, M., Nivet, M. L. (2013). Multi-horizon solar radiation forecasting for Mediterranean locations using time series models. Renewable and Sustainable Energy Reviews, 28, 44-52.

Wang B., Tai, N. L., Zhai, H. Q., Ye, J., Zhu, J. D., Qi, L. B. (2008). A new ARMAX model based on evolu- tionary algorithm and particle swarm optimization for short-term load forecasting. Electric Power Systems Research, 78(10), 1679-1685.

Yager, R. R., Kreinovich, V. (2003). Universal approximation theorem for uninorm-based fuzzy systems modeling. Fuzzy Sets and Systems, 140(2), 331-339.

# CURRICULUM VITAE

**Personal Information**

Name Surname : Caner Vatansever
Place and Date of Birth : İstanbul, 1990

**Education**

Undergraduate Education : Bachelor in Management Information Systems, Boğaziçi University
Graduate Education : Master in Management Information Systems, Kadir Has University
Foreign Language Skills : English

**Work Experience**

Turkish Airlines, 2014-Current
Ibtech (Finansbank), 2014
YapıKredi Bankası,2013-2014

**Contact**

Telephone : 00905378827633
E-mail Address : caner.vatansever@gmail.com