Pázmány Péter Catholic University
Faculty of Information Technology and Bionics
Computer Science Engineering M.Sc. Program

Data Processing of Measurements Collected in IoT Systems with Machine Learning

Övgü Özdemir

A thesis presented for the degree of Master of Science

May 2018

Advisor:
Dr. Kálmán Tornai

J/1773/2017

PÁZMÁNY PÉTER CATHOLIC UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY AND BIONICS
REGISTRAR'S OFFICE
H-1083 Budapest, Práter u. 50/a.
Tel: (36-1) 886-4711 FI: 79633 E-mail: tanulmanyi.osztaly@itk.ppke.hu

Diploma Thesis Proposal

Student

Name: Övgü Özdemir	**Neptun ID:** I2HGFJ
Study program: Computer Science Engineering MSc IMNI-AMI	

Supervisor

Name: Kálmán Tornai
Position: associate professor

Thesis

Title: Data processing of measurements collected in IoT systems with machine learning

Summary of the thesis:

Smart systems enable huge amount of data to be collected from power consumers. It is required to process this data in order to obtain information about the system and consumers' behaviors. During the thesis work, the objective is to investigate different neural networks based classification and clustering methods on measurement data provided by the Internet of Things networks, especially sensor data from intelligent home applications and smart electricity grids. Based on the behavioral patterns of the consumers they can be categorized in order to increase efficiency and provide better or new service.

Detailed task description:

The student will investigate the answers of the following questions in the thesis work.

- How can be the data transformed to eliminate the influence of the seasonality of the measurements?
- What are the minimum limits of the neural network based solutions? (How much data points are required in the training set, while the test performance of the solution remains acceptable?)
- How could be the number of classes further increased?

- How can be the neural networks based solutions used for clustering? (Is it possible at all to have good results?)
- What is the performance of the investigated methods on other type of application/data set?

In order to solve the tasks above, the student has to:

- survey the appropriate literature to seek algorithms, methods, and solutions concerning with time series classification, clustering and possibly segmentation.
- implement the solutions in Python or Matlab language.
- execute extensive tests on the implemented solutions and evaluate thoroughgoing the results.

Hereby I undertake to supervise the Diploma Thesis work of the student.

Signature of supervisor

Hereby I apply for the approval of the theme of my Diploma Thesis.

Budapest, 28 November 2017

Signature of student

The theme of the Diploma Thesis has been approved by the Faculty of Information Technology and Bionics.

Budapest, 14/12/2017

Dean

The student has regularly attended consultation sessions, and has met all the requirements of the task description.

The diploma thesis work meets the formal and substantive requirements of Annex No. 3 of the Code of Studies and Exams.

Budapest, 2018. 05. 14

Signature of supervisor

Declaration of Originality

I, the undersigned Övgü Özdemir, student of the Pázmány Péter Catholic University, Faculty of Information Technology and Bionics, declare that I have written this diploma thesis solely myself, without any unauthorized help, and I have only used the sources referenced. Every part quoted word by word or in a paraphrased manner is indicated clearly, with a reference made. I have not submitted this diploma thesis in any other training program.

Signature

Contents

Abstract

In the Internet of Things systems, smart meters gather a huge amount of data about the elements of the system. Processing of the measurements enables to obtain comprehensive information about the users taking part in the system and improve the service intelligently. In Smart Grids, it is a significant task to process the data collected by smart meters for the analysis of energy usage. Every consumer in a smart grid tends to use the energy different ways. Hence, it is an important task to achieve classification of different types of consumers in Smart Grids at a high accuracy and efficiency. To manage the capacity, pricing and distribution planning efficiently, electric power should be provided to the consumers having different consumption patterns different ways.

Related works are trying to solve the classification task using various machine learning methods. However, the thesis work focuses on solving the classification task with deep learning, using particularly convolutional neural networks, as well as considering the performance of the other related machine learning algorithms. Even though deep neural networks have come up with remarkable performance results in case of two-dimensional images for several image classification applications, the capabilities of the deep neural networks have not been comprehensively investigated in case of using time series for this type of classification task. Thereby, the thesis presents a deep learning approach to be able to solve the classification problem of power consumers in Smart Grid and demonstrates promising performance results as a result of different tests. The results are obtained using a publicly available database containing electricity consumption measurements collected from 16 types of consumers (buildings) in the United States.

As the consumption measurements are time series, data has been transformed to two-dimensional form. Implementation of convolutional neural networks and performance tests on the electricity consumption measurements have been carried out using Python and Keras deep learning environment. The purpose of the tests has been to observe the limit of the deep neural networks by evaluating different test scenarios regarding input length and seasonal impact on the data set and to obtain a detailed analysis for the solution of the classification of the measurements. For this purpose, implemented deep neural networks have been evaluated using different seasons to find how much deep neural networks are influenced by the seasonality. The model has also been evaluated using different input lengths to find the minimum limit. Besides, the performance has been tested when the number of the classes were increased based on the seasons. To demonstrate the performance of the model for another classification application, tests have been performed on the data set of measurements collected by smart sensors containing time series of human activities. Both results demonstrate that the convolutional neural networks are capable of solving the classification task on time series data at high

level of accuracy. The performance results on the electricity consumption data set have been compared with different classification algorithms as well, and the comparison indicated that the convolutional neural networks had the most remarkable performance results among the other machine learning algorithms.

Chapter 1

Introduction

Data processing is described as manipulating and transforming the data to obtain meaningful information. In the digital era, different types of data can be collected different ways, and the amount of data collected is rapidly increasing day by day. As well as the amount of data, improvements in software and hardware are also important to be able to process the data efficiently. Particularly, developments in parallel computing and machine learning algorithms have enabled that the data processing has become more popular in many fields over the past decade.

Data processing of measurements collected in the Internet of Things systems is a significant task, to obtain efficient and automatized systems controlling the elements of the system intelligently. One of the important applications of the Internet of Things can be seen in increasingly developed smart grids. While different size and type of power plants and consumers take part in a smart grid, the system provides a two-way information flow between them. Traditional energy grids have to produce more energy to meet the demand needed. However, in smart grids it is possible to use the resources in a more efficient way thanks to the two directions of communication flow allowed. Smart sensors are the important elements of smart grids, that are capable of collecting a vast amount of data from the energy consumers. Hence, data processing of measurements collected by the smart sensors plays a critical role in smart grids to achieve a more efficient and smarter energy distribution and transportation, using the information about the energy usage of the consumers. For this purpose, classification of the energy consumers in smart grids emerges as an essential problem that has to be solved.

This chapter provides the introduction about the specific problem approached in the thesis work, a summary description of the data set used in the major part of the work, analysis of the detailed tasks declared in the thesis proposal and the structure of the thesis document respectively.

1.1 Problem Statement

Distribution of the electric power should be provided in different ways to the consumers in smart grids as every consumer has its necessities, profile and consumption behavior. Information that will be gathered on energy usage of the consumers can be used for planning of the balance between supply and consumption, better organizing for the distribution of the energy, storing and pricing. For this reason, it is significant to achieve the classification of the consumers at high accuracy level. There are several statistical and machine learning methods applied for the solution of the problem. However, convolutional neural networks have not been investigated comprehensively on time series. The purpose of this work is to process the measurements collected in smart grid, more specifically to observe the classification capabilities of the convolutional neural networks on electricity consumption time series belonging to the consumers in a smart grid. The thesis work tries to obtain a broad analysis of data processing of measurements using a deep learning method by considering different evaluation scenarios regarding the input length and seasonal impact as well.

1.2 Summary Description About the Data Set

The data set contains hourly load profile data, that was collected in US, belonging to 16 different consumers (or building types): Full-Service Restaurant, Hospital, Large Office, Medium Office, Small Office, Large Hotel, Small Hotel, Midsize Apartment, Primary School, Secondary School, Quick Service Restaurant, Warehouse, Supermarket, Outpatient Health Care, Stand-alone Retail and Strip Mall. Every measurement belonging to the consumers is a time series measured throughout one year. The consumers are located in several sites which exist in different states and climate zones in the US.

1.3 Analysis of the Tasks in Thesis Proposal

Detailed task description contains five questions that have been declared to be investigated in the proposal. Firstly, it has been investigated the solution of how the data can be transformed to eliminate the influence of the seasonality. This problem has been solved by converting the data into two-dimensional form and processing the transformed data using deep neural networks. Secondly, limits of the deep networks have been examined regarding the input length. Different amounts of input data have been evaluated and performance results have been compared. Thirdly, to observe the case when the number of the classes have been increased, the existing 16 classes have been divided based on the seasons, and a performance evaluation has been done. For the fourth question that is about how the neural networks can be used for clustering, some of the clustering methods have been reviewed as well as neural networks based classification algorithms, and it is determined that detailed research about clustering can be done in the future work. Lastly, to observe the performance of the deep learning method for another application, some tests have been conducted on another measurement data for the classification of human activities. To achieve these tasks, literature has been reviewed extensively to seek existing machine learning algorithms and methods used for clustering and classification. Afterward, solutions and tests have been performed on Python, and the performance analysis has been done.

1.4 Structure of the Thesis Document

The main body of this thesis document is split into five chapters. The first chapter gives an introduction to the specific task as well as the problem tried to be solved in this work, and provides short information about the data set used and the analysis of the detailed tasks declared in the thesis proposal. The second chapter presents the theoretical information about technological concepts and related machine learning algorithms, and training techniques regarding the thesis work. The third chapter provides details about the implementation of the deep neural networks and design choices behind the implementation, as well as the data preparation steps applied. The fourth chapter presents the performance results obtained as a result of several tests evaluating seasonal impact, input length and comparison of the implemented structure with other algorithms. Finally, the fifth chapter provides a summary of the overall work and results achieved, conclusion and future work.

Chapter 2

Related Work

This chapter introduces the related work behind the thesis topic. First, the background about the relevant technology is given. Afterward, several clustering and classification algorithms regarding the task, and some training methods used in machine learning are presented.

2.1 Technological Background

This section describes the theoretical information about the Internet of Things and Smart Grid technologies with reference to the main problem approached in the thesis work, to provide a general background to the reader.

2.1.1 The Internet of Things

The Internet of Things (IoT) describes a wireless network of interconnected devices which can contain sensors, physical devices, smart objects, home appliances, RFID and other embedded electronics items. In other words, the IoT is a system that allows the items existing in the network to communicate each other. The term 'IoT' was first introduced by Kevin Ashton in 2002, though the notion was used by Neil Gershenfeld in 1998 [1]. Then the IoT concept has increasingly spread in both academia and industry. Smart objects are the important elements of the IoT as they are capable of interacting with both people and other smart objects, accessing the Internet services. Smart sensors enable monitoring of the system environment and interactions, as well as collecting and processing the data transmitted in the network.

The Internet of Things enables numerous applications for intelligent systems used for smart buildings, transportation, and energy, and has the potential of developing many more applications in the future. One can observe one of the important applications of the IoT in a smart grid. Facilities of interconnection, automation, and monitoring of the devices are achieved using the IoT in a smart grid.

2.1.2 Smart Grid

A smart grid is described as a network which consists of electricity generators, consumers, and the intelligent interactions between them [2]. Due to increased power demand, power systems arrived at a new level through the developments of smart grid networks.

A traditional power grid carries out three basic operations that are: generation, transmission, and distribution of electrical power and it flows only in one direction, which is from the provider to the consumers [3]. In opposition to the traditional power grid, a smart grid is capable of having two-way flows of communication between providers and consumers by intelligently collecting measurements [4]. In a smart grid, information belonging to consumers about their electricity consumption is transmitted to the providers. Measuring of consumers electricity consumption is achieved by smart meters existing in the smart grid. Smart meters can collect a huge amount of data from the consumers. Thereby, the system enables the electric power to be distributed and transported more efficiently. Also it can manage the balance between supply and demand appropriately, compared to the traditional power systems.

The measurements system of the smart grid enables new applications to be implemented as well as to use the data gathered in the network to optimize the energy flow, and also to provide new features to the consumers. According to the demand for electrical energy, important decisions such as pricing and scheduling can also be made.

2.2 Clustering

Clustering is an unsupervised learning method in machine learning. Clustering methods state creating groups of objects based on their common characteristics. Clustering is used in the case that there is no knowledge about the target classes of the objects. It is needed to have more similarity between the objects in a group or more differences between groups for a better clustering [5]. Generally, the measure of the similarity is stated by a distance measure (or similarity measure) such as Euclidean, Manhattan, Jaccard distance. Cluster analysis is used for many different applications of biology, marketing, statistics, pattern recognition. It can also be applied to the consumption measurements for the clustering of the power consumers. There are many clustering algorithms used for different applications, however, in this section K-means algorithm, Mean Shift Algorithm and Hierarchical Clustering which are the basic and widely used clustering methods, will be reviewed.

2.2.1 K-means Algorithm

K-means is one of the most straightforward algorithms for clustering. The term 'k-means' was first expressed by James Macqueen in 1967 [6]. However, the standard algorithm was first presented by Stuart Lloyd in 1957 (his paper was published in 1982) [7]. The standard algorithm, also known as Lloyd's algorithm, is reviewed in this section, though there are many variations of it.

The algorithms first choose K random centroids such that K is the parameter that states the number of the desired clusters set by a user. After specifying the centroids, it assigns each point to the nearest centroid. The points that are nearer to the centroid become elements of a group or cluster. The algorithm updates the centroid of each cluster, assign the points considering the new centroid. This process continues until the centroids remain same. There have been some works, such as Kwac et al. [8], using K-means algorithms for the cluster analysis on power consumption data. K-means algorithm has the advantage of converging fast and this can be feasible in case of large amounts of data. However, the performance of the results considerably depend on the starting points, and the performance can be influenced by the noise.

2.2.2 Mean Shift Algorithm

Mean shift clustering is based on the density function. It uses kernel density estimation method to estimate the probability function of a given data set [9]. It works using kernels sliding on the data set and finds the densest region of the data points. There are many types of kernels. However, Gaussian kernel [10] is one of the most popular ones. The algorithm try to find local maximum of the density, which is called 'mode', within the window sliding and shifts the window to the densest region [9]. This process is described mean-shift iteration and repeated until all the data points in the window converge to the same mode defining the cluster. Mean shift method is used in several applications of smoothing, tracking as well as the clustering. It is a simple algorithm and convenient for real time data analysis. However, a disadvantage of the algorithm is that it can take very long time to execute when applied to large amounts of data.

2.2.3 Hierarchical Clustering

Hierarchical clustering methods tries to build a hierarchy of clusters based on the similarity between groups of data. There are two main approaches to hierarchical clustering that are agglomerative and divisive. Agglomerative clustering algorithm first puts each data point into its cluster and then merges two nearest clusters as one iteratively until all the points are merged [11]. In contrast, divisive clustering algorithm puts all the points into one cluster and then divide them into groups using a second clustering algorithm such as K-means. Since it needs another algorithm, divisive clustering has more complexity than agglomerative method.

2.3 Classification Algorithms

In machine learning, classification is a supervised learning task to categorize a new observation by a training set which is already labeled. Classification can be achieved on several types of data such as image, text, audio or any time series. There are many classification algorithms developed including decision trees, regression trees, statistical models. However, in this section four frequently used algorithms, namely k-Nearest Neighbor, Feed Forward Neural Networks, Support Vector Machine and Hidden Markov Model will be reviewed as they are usable algorithms for classification of power consumption measurements. Besides, Convolutional Neural Networks, which has been used in the practical part of this work, will be introduced with more details.

2.3.1 k-Nearest Neighbor

k-Nearest Neighbor (k-NN) is one of the most basic and simple classification algorithms. K-NN assigns unlabeled observations to the class of the nearest (or most similar) labeled samples. k-Nearest Neighbor and other distance-based classification methods base on a metric, i.e., a distance function, to decide the class of an unlabelled new sample [12]. To do that, a distance between new unlabeled sample and elements of the other classes are calculated and stored, and the 'closest' class is found.

Several distance functions can be used as a similarity measure, such as Euclidean distance, Manhattan distance, Hamming distance [13]. The Euclidean distance function is used in most of the problems, though it could not be appropriate for all cases [14, 15].

The k-NN method has advantages since it is simple to use, and practical when a large amount of training data is used. However, it has the disadvantages of high computational cost and memory requirement.

The k-NN method can be efficiently applied to power consumption time series. However, it is very sensitive to noise in the data, and the performance of k-NN method highly depends on the similarity measure [16].

The Fuzzy k-NN classification method is similar to the basic k-NN. Fuzzy logic describes a membership degree for every classification object, with respect to this, in Fuzzy k-NN a new object has a membership degree for the different classes. Thereby, the method does not assign a new unlabelled sample to a specific class, instead, estimates the similarity degree for different classes. When the membership degree of a new object exists between two or several classes, this situation may cause creating a new class [17]. This algorithm can achieve better performance than simple k-NN. However, it may happen that new objects cannot be assigned to existing classes in the case that several small classes are created.

2.3.2 Feed Forward Neural Networks

A Feed Forward Neural Network (FFNN) is one of the most frequently used artificial neural networks architectures for solving classification problems. The simplest form of FFNNs, which is called single-layer feedforward networks, consists of only input and output layers. Multilayer FFNNs structure mainly consists of an input, one or more hidden layers and an output layer. All connections between neurons are in a feedforward type, which means neurons in one layer are connected to the neurons in the next layer, and there is no feedback. Figure 2.1 indicates an FFNN structure having one input, hidden and output layers.

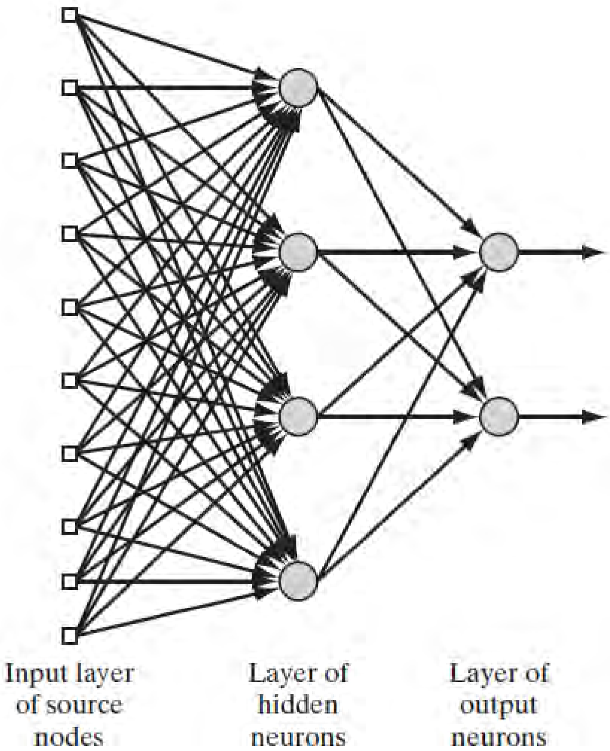


Figure 2.1: An FFNN structure having one hidden layer [18]

The FFNNs can be trained using back-propagation algorithms. Input and output values are given to the network for many iterations so that the network learns the relationship between the input and output by calculating and optimizing weights. When an input vector of a training sample is found, the output vector is compared to the target value by calculating the error, and the aim is to minimize the error to approach the target.

The FFNNs and different types of artificial neural network have been applied to solve the classification problem of consumers in a smart grid [19]. However, the performance of the network highly depends on the choice of the training algorithms, initialization of the weights, the order of the samples in the training set.

2.3.3 Support Vector Machine

Support Vector Machine (SVM) is a type of artificial neural networks and one of the popular machine learning methods used for both pattern classification and non-linear regression. Simon Haykin summarizes the notion behind support vector machine in his book 'Neural Networks and Learning Machine' as follows: “Given a training sample, the support vector machine constructs a hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized.” [18]. Support vectors are the samples which are closest to the optimal hyperplane. Figure 2.2 indicates the optimal hyperplane for linearly separable patterns in 2D space. Also, data points can be represented in multidimensional space, and SVM can be used for multiclass classification problems by constructing several hyperplanes.

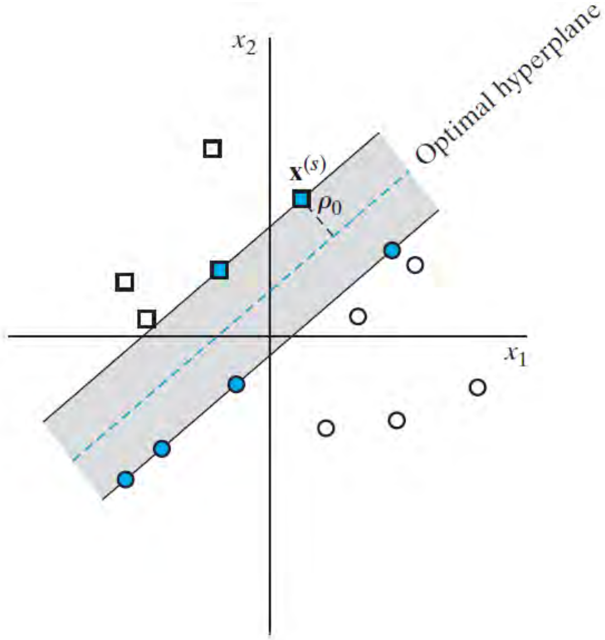


Figure 2.2: Optimal hyperplane for linearly separable patterns in SVM [18]

2.3.4 Hidden Markov Model

A Hidden Markov Model (HMM) is a probabilistic Markov model where for a given sequence it finds the probability distribution over the possible sequences of classes and selects the best class sequence [20]. A Markov Model is used in the case that there is need to compute the probability for a sequence of events that are observable. However, some events may not be observable; this is why those are described as 'hidden' in HMM [20]. In other words, HMM is a Markov model having unobserved states. HMM can be used for time series prediction and classification. Natural language processing and bioinformatics are the fields in which HMM is applied frequently. HMM requires prior knowledge to select the particular features from the input and proability assumption on the states [21].

2.3.5 Convolutional Neural Networks

In recent years, deep learning methods have dramatically developed and become hot topics particularly in computer vision, language and speech processing. Convolutional neural networks are one of those methods as they offer a powerful approach for learning problem-specific features. The convolutional neural networks (CNNs) has a long history, as CNNs were constructed and demonstrated by Fukushima Kunihiko in 1980 [22]. Also, Yan LeCun demonstrated that CNNs are successful in many different problems such as document and handwritten recognition in 1998 [23]. However, these networks have not been applied in practical applications until 2011, due to the memory and hardware constraints. Also, it has not been accessible to have a significant amount of data for the training of the networks.

Improvement in computer hardware, particularly with GPUs, and availability of large-scale data sets allow that CNNs become more potent for classification problems in computer vision [24]. It has also been demonstrated that in several specific problems they are capable of the solving the classification problem at a very high level of accuracy which is comparable to the human performance. Also, there are problems, where CNNs can outperform the humans as well. However, CNNs are one of the most popular techniques, because of the strong capability of representing the features of the data, used for the problems in computer vision such as facial detection, object recognition, and natural language processing applications.

The CNN can also be suitable for the classification of the power consumption measurements by extracting the specific features in the data belonging to the consumers. However, there has not been a broad analysis for the usage of the CNNs on time series data containing consumption measurements. In this work, the implementation and performance of the CNNs will be tried to investigate.

CNNs focus on learning unique features for solving the classification problem. CNNs use a set of convolutional filters for learning and train filters as feature identifiers. This operation is fulfilled with hidden layers, which can be convolutional layers, non-linearity layers, pooling layers, fully connected layers and softmax layers. According to the necessities, several layers may be added, and more features may be extracted. In addition to the layers, to keep the distribution of the input batch normalization is mostly used between the layers in CNNs.

2.3.5.1 Convolutional Layer

A convolutional layer applies a convolution operation with a set of filters to the input of the layer. A filter having same weights is applied across the spatial structure, which is two-dimensional generally in the case that input data is an image. The output of the convolutional layer will be the input of the next hidden layer. The convolution operation causes reducing the number of free parameters, allowing that the network can have many hidden layers, while the number of free parameters is few [25].

During the training phase, a convolutional layer creates a map for unique features in the input by computing element-wise multiplications. This map is called as a feature map. By applying a series of convolutional layers, the principal features can be identified and recognized, while the complexity of the input data is reduced. Figure 2.3 represents these operations.

2.3.5.2 Non-linearity Layer

In CNNs, each convolutional layer is usually followed by a non-linearity layer. This layer is also called as an activation layer. The purpose of the non-linearity layer is

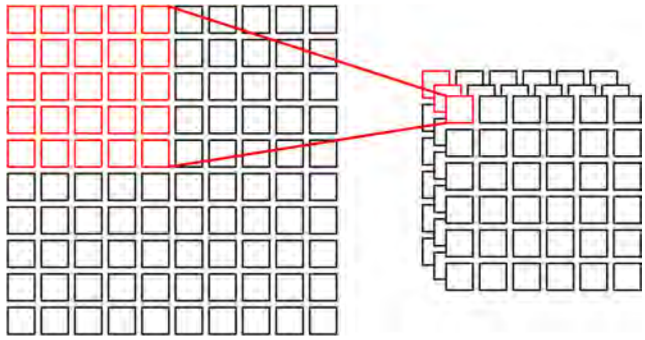


Figure 2.3: Convolution operation

to introduce non-linearity to the system that has created the feature map by linear operations during the convolutional layers. A non-linearity layer takes the feature map created by convolutional layer and brings an activation map as an output. In case of artificial neural networks the saturating hyperbolic tangent $f(x) = \tanh(x)$ or $f(x) = |\tanh(x)|$ and the sigmoid function $f(x) = (1 + e^{-x})^{-1}$ are commonly used, however, in CNNs it is found that rectified linear units (ReLu) are the best choice for several problems [26]. The output of a rectified unit is 0 when the input is 0, otherwise the output is equal to the input:

$$f(x) = max(x, 0) \tag{2.1}$$

2.3.5.3 Pooling Layer

An important layer in CNNs is the pooling layer, which is responsible for down-sampling the data in a nonlinear way. Several non-linear functions can be used to implement pooling. However, the max pooling and average pooling are most commonly used pooling techniques. There are other pooling methods, such as mean pooling and L2-norm pooling.

An average pooling divides the 2D input data into a set of disjoint subsets and outputs the average of each subset, while a max pooling outputs the maximum of each subset. The pooling operation reduces the size of the representation of the data and also expresses the most important features. It helps to reduce the computation cost and the risk of overfitting. Figure 2.4 shows an example of average pooling.

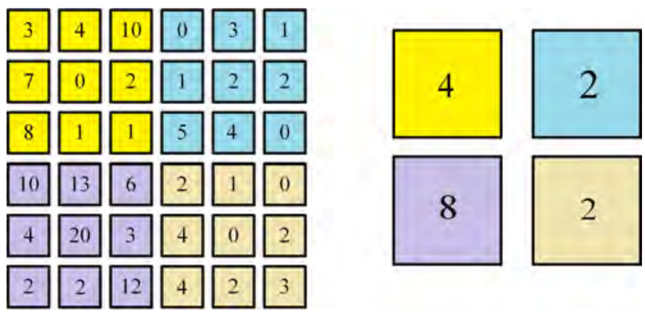


Figure 2.4: Example for average pooling

2.3.5.4 Fully Connected Layer

After several convolutional, non-linearity and pooling layers, one or more fully connected layers are usually deployed in CNNs. A fully connected layer is generally a regular feed-forward neural network which consists of input, output, and several hidden layers, and has full connections between all the neurons in the consecutive layers. The purpose of the fully connected layer is to provide a high-level representation of the knowledge learned by the network.

2.3.5.5 Softmax Layer

In this layer a softmax function is applied to the artificial neurons as the activation function (or normalized exponential):

$$P\left(\mathcal{C}_r|u\right) = \frac{e^{a_r}}{\sum_{j=1}^{k} e^{a_j}}, \tag{2.2}$$

where $0 \leq P\left(\mathcal{C}_r|u\right) \leq 1$ and

$$\sum_{j=1}^{k} P\left(\mathcal{C}_j|u\right) = 1.$$

In previous equation $P\left(u|\mathcal{C}_j\right)$ denotes the conditional probability of the sample given class r and $P\left(\mathcal{C}_j\right)$ is the prior probability of the class, and

$$a_r = \ln\left(P\left(u|\mathcal{C}_j\right) P\left(\mathcal{C}_r\right)\right).$$

The softmax function can be considered the multi-class generalization of logistic sigmoid function.

2.3.5.6 Batch Normalization

In deep networks as CNNs, distribution of the input of each layer alters during the training as the parameters of the previous layer change [27]. Ioffe and Szegedy say as follows “The change in the distributions of layers’ inputs presents a problem because the layers need to continuously adapt to the new distribution.” in their article [27], where they introduced the notion of batch normalization, published in 2015. This problem causes a deceleration in the training and sensitivity in the initial parameter changes. To deal with this problem, a normalization step is applied so that the means and variances of layer inputs can be fixed, and this is called Batch Normalization.

Batch Normalization helps to accelerate the training. Besides, it reduces the dependence of the gradients on the initial parameters and enables using higher learning rates without divergence [27].

2.4 Training Techniques

In this section, some techniques used in the training process of machine learning will be reviewed. In machine learning, the choice of the optimizer, loss function and regularization method play a critical role in the training to find the optimal results by having less time cost. To have a better understanding of the training process, some of the frequently used loss functions and optimization methods and the regularization approach will be introduced respectively.

2.4.1 Loss Functions

In classification problems, a loss function represents the cost in the case of inaccurate classification. In other words, the loss function states the loss between calculated output and desired output. The aim of training is minimizing the loss, to approach the optimal solution. Two main loss functions, Mean Square Error and Cross Entropy, will be introduced in this section.

2.4.1.1 Mean Squared Error

Mean Squared Error (MSE) is a loss function usually used to train neural networks. It states the expected value of squared error loss.

$$MSE = \frac{1}{n}\sum_{i}^{n}|e_i|^2 \tag{2.3}$$

such that e_i is the error between calculated output and desired output for n predictions. It can be used for binary and multiclass classification problems.

2.4.1.2 Categorical Cross Entropy

Categorical Cross Entropy is a loss function which states the loss as negative log likelihood. The equation is given as

$$H(y, \hat{y}) = -\sum_{i} y_i \log \hat{y}_i \tag{2.4}$$

where y and $\hat{y}$ are the vectors representing the distribution of the desired outputs and calculated outputs by the classifier respectively. Categorical cross entropy is frequently used for multiclass classification problems.

2.4.2 Optimization Algorithms

Optimization is an important step in the learning process of training. To obtain optimal results in a shorter training duration, it is significant to choose the appropriate optimization method. Although there are several optimization algorithms, four of those methods, stochastic gradient descent, adagrad, adadelta, and adam, which are widely used and related to the neural network architecture implemented in this work will be reviewed.

2.4.2.1 Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is a stochastic optimization technique which computes the parameters θ with respect to the objective function $J(\theta)$ for each training examples $x^{(i)}, y^{(i)}$ [28].

$$\theta = \theta - \alpha \nabla_{\theta} J(\theta; x^{(i)}, y^{(i)}) \tag{2.5}$$

Stochastic Gradient descent tends to reach good convergence for a local optima [28]. For many problems, SGD is the main optimization method used in machine learning.

2.4.2.2 Adagrad

Adagrad [29] is a gradient-based optimization method that changes the learning rate and adapts them to the parameters [30]. In Equation 2.5, stochastic gradient descent algorithm uses the same learning rate α while updating the parameters θ at each iteration. However, Adagrad changes the learning rate accordingly for every parameter at each iteration.

2.4.2.3 Adadelta

Adadelta is an extension of the Adagrad optimization method. MD Zeiler states Adadelta to be improved for two drawback that Adagrad has, which are constantly decreasing learning rate and a global learning rate that has to be selected manually [31]. Adadelta solves these two drawbacks by bounding the window of past gradients rather than storing the sum of all the squared gradients [31].

2.4.2.4 Adam

Adaptive Moment Estimation (Adam) [32] is another optimization method which has the same notion of adaptive learning rates, similar to Adagrad and Adadelta. Kingma and Ba states in their paper that Adam fits better with the problems that require large data and parameters, and fewer memory [32].

2.4.3 Regularization

Deep networks are capable of learning the relationships, even the very complicated ones, between input and output. However, due to some reasons, such as inadequate amount of training data or too much noise in the data set, the network can memorize the training set and be unsuccessful in the classification of new samples that do not exist in the training set [33]. In this case, the network memorizes the training set instead of learning, and this is called overfitting. Overfitting is a serious problem in machine learning.

Although there are several regularization techniques such as L1 and L2 regularization, dropout method will be reviewed in this subsection as it is used in the practical part of this work. The technique was first introduced by Srivastava et al. in 2014, and they explain the dropout technique as dropping the units (with all forward and backward connections) out from the network by choosing randomly [33]. Each unit is either kept in or removed from the network with a fixed probability p. Figure 2.5 indicates the comparison between a standard neural network containing two hidden layers and a neural network that has lost some of the connections after applying the dropout technique.

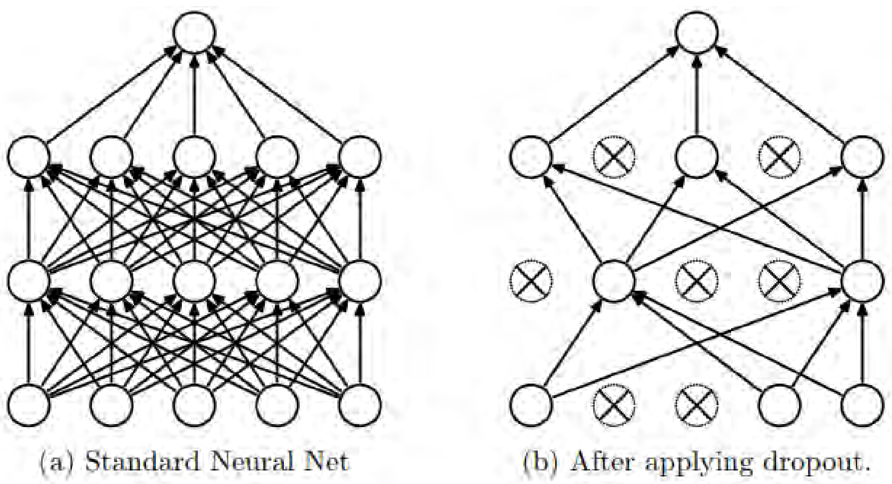


Figure 2.5: A standard neural net and a neural net after applying dropout [33]

Dropout technique is generally used between the layers in convolutional neural networks to robust the network against to the overfitting, and give successful results.

Chapter 3

Implementation

The convolutional neural networks have been implemented for the classification task approached in this work. This chapter describes the preparation of the data set, the explanation of the training setup and the implemented structure of the convolutional neural networks. In Section 3.1, the details of the data set is given, as well as explaining how the data set has been used before the training. In Section 3.2, the training setup and the structure implemented on Python are introduced, and the design choices behind the implementation are given.

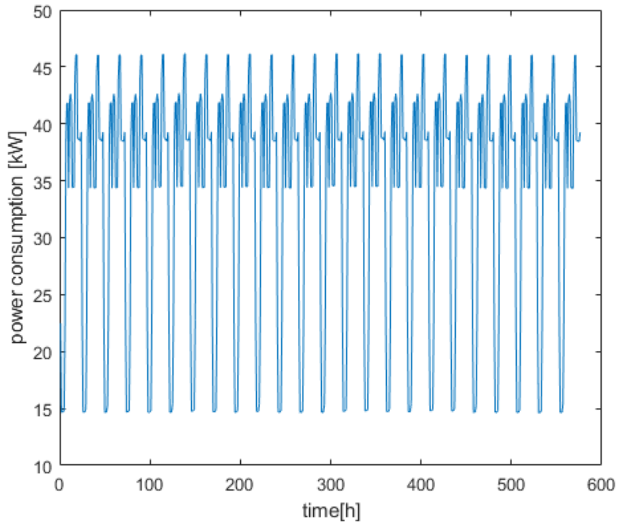
3.1 Data Preparation

This section describes the detailed information about the electricity consumption data set, explains the method to transform the data into two-dimensional form to use it with CNNs, and impact of the different seasons on electricity consumption measurements.

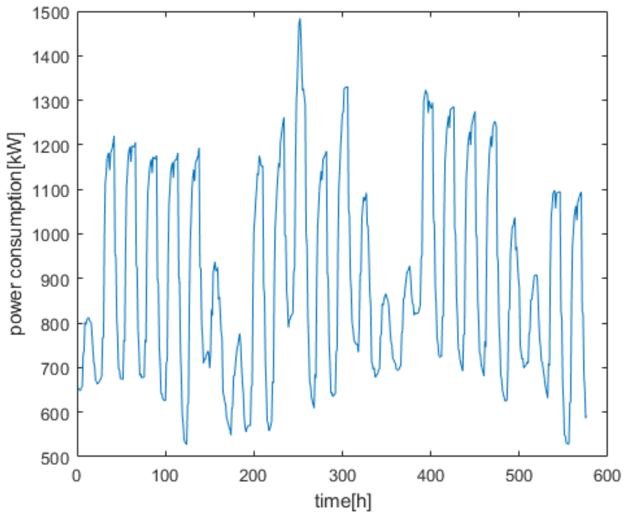
3.1.1 Electricity Consumption Data Set

Data set were constructed using a publicly available database, from 'Commercial and Residential Hourly Load Profiles for all TMY3 Locations in the United States' [34]. This database contains hourly load profile data measured throughout 365 days in a year. It contains electricity consumption measurements, which are time series, collected my smart meters from 16 different commercial (based on the DOE commercial reference building models) and residential building types (based off the Building America House Simulation Protocols): Full-Service Restaurant, Hospital, Large Office, Medium Office, Small Office, Large Hotel, Small Hotel, Midsize Apartment, Primary School, Secondary School, Quick Service Restaurant, Warehouse, Supermarket, Outpatient Health Care, Stand-alone Retail and Strip Mall. The hourly load profiles are available for overall TMY3 locations in the United States. This publicly available database consists of electricity consumption measurements of commercial and residential buildings being from 936 sites located in 50 states in the United States. Figure 3.1 indicates examples from time series of the measurements belonging to two different consumers (a full-service restaurant and hospital) located in the same site.

Because of the size of the database, the number of the sites have been reduced from 936 to 160 by randomly selecting from 12 states. Considering that usage profiles of the power consumers may change according to climate conditions, the states in which



(a) 24 hours measurement of a restaurant



(b) 24 hours measurement of a hospital

Figure 3.1: Measurements from two different buildings

consumers are located have been randomly selected from different pre-defined climate zones. The states where the consumers have been selected from have been Ohio (OH), Kentucky (KY), Tennessee (TN), Alabama (AL), Arkansas (AR), Texas (TX), Oklahoma (OK), Nebraska (NE), Kansas (KS), North Dakota (ND), South Dakota (SD) and California (CA). Figure 3.2 indicates the map of the US states based on the climate regions indicated in different colors.

3.1.2 Transformation of Time Series

As mentioned in Section 3.1.1, the data set contains hourly based electricity consumption time series measured throughout 365 days. Since each time series is one-dimensional data, it is not suitable to be used with two-dimensional convolutional layers in CNNs which are generally applied to 2D images for image classification applications. Hence, to transform

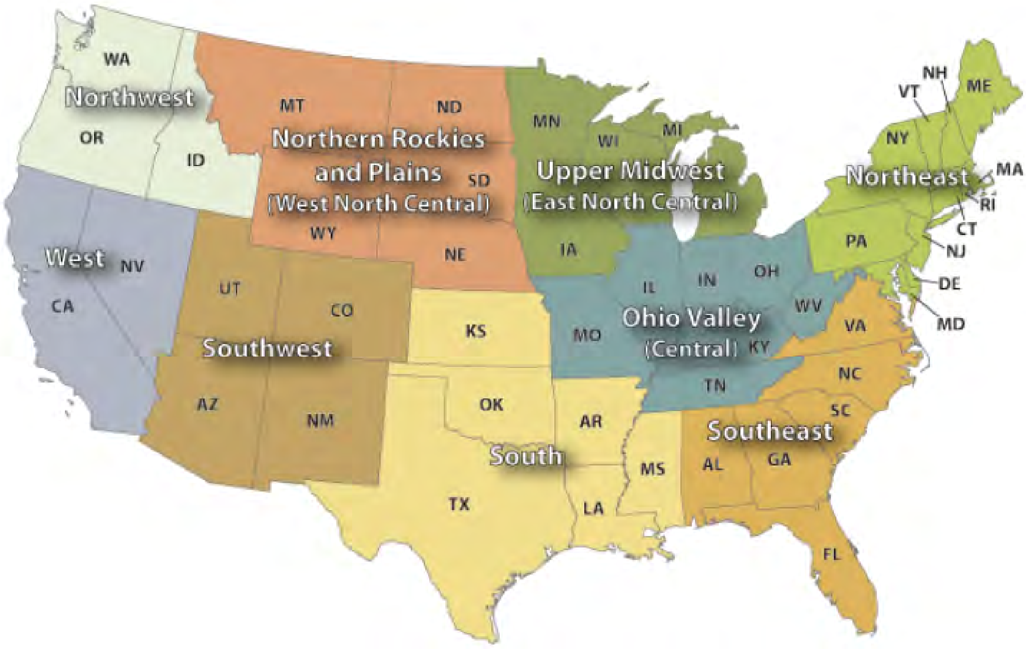


Figure 3.2: Climate regions map of US [35]

the one-dimensional time series into two-dimensional form some data preparation steps have to be done.

In the previous work, the auto-correlation matrix of each time series has been tried to construct to obtain a two-dimensional representation of the time series data. However, this way of transformation did not represent the data appropriately, and the evaluations did not demonstrate good performance results.

A more suitable way of transformation has been applied by representing each time series in a similar way to two-dimensional images. For this purpose, the data is represented with (x, y) coordinates, where x represents the day, and y represents the hour. From each time series, 576 hours long data, that cover 24 days, have been selected and 24×24 rectangular matrices have been constructed. Figure 3.3 represents the transformation operation.

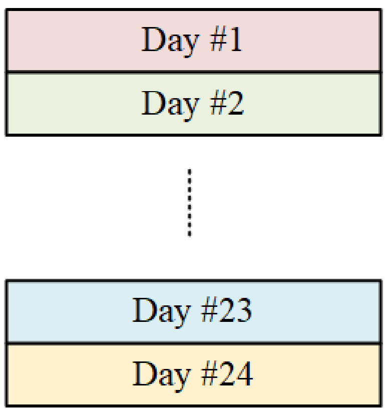


Figure 3.3: Transformation of time series vector to matrix

For different performance evaluation scenarios using less input (that will be mentioned in Section 4.1.3), each of 24 hours data has been re-sampled by merging two hours as one, so that the frequency has become 2 hours. Thereby, one day has been represented with 12 hours and 144 hours long data has been used to construct 12×12 rectangular matrices. The same logic has also been applied to construct 8×8 rectangular matrices using 64 hours (8 days) long input.

3.1.3 Different Seasons

There may be many factors that have an impact on power consumption of the users (smart buildings). The consumption behavior can change depending on the number of the people living or working in a building if the day is a workday or holiday, or meteorological conditions. However, seasonal impact on weather conditions has been taken into account as the most significant factor among the all possible factors affecting the behaviors of the consumers changing throughout a year.

Electricity consumption is highly correlated with the outdoor temperature. Even though the seasons do not have the same characteristics in every climate zone, one can assume that the weather conditions affect consumers' consumption behaviors. For example, hot weather may cause high energy consumption in a supermarket or a strip-mall because of the air conditioners working all the day, while the consumption in a midsize apartment may not be too much affected because of this situation.

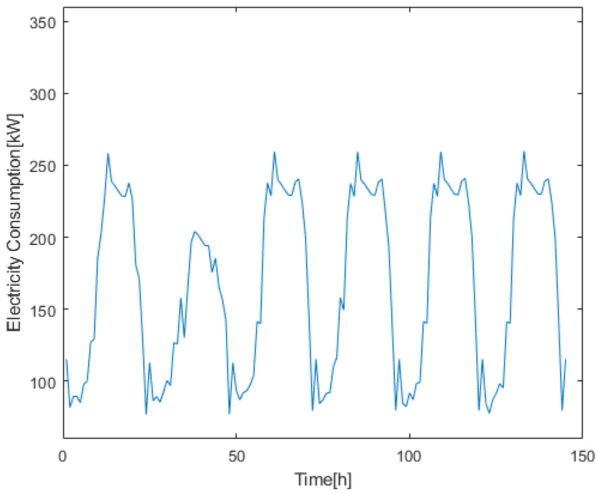
As an example of seasonal differences, six days of electricity consumption data which belongs to the same supermarket in winter and summer are indicated in Figure 3.4a and Figure 3.4b. It can be observed that minimum values and peak points measured during these six days in summer are much higher than the values measured in winter.

To understand the impact of the seasonal changes on the data set and to test how much convolutional neural networks are affected by seasonality, based on the time stamps the data set has been divided into four subsets according to the four seasons. The seasons are determined by using the meteorological definitions. These subsets have been used for different test scenarios that will be explained in Section 4.1.2.

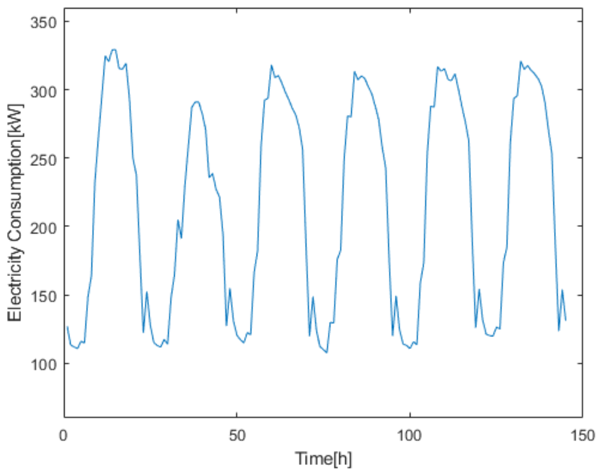
3.2 Implementation of Deep Networks

Implementation of the deep networks has been performed using CNNs on Python using the Keras [36] deep learning framework. Keras allows using many functions, layers, optimizers that exist in the library to construct deep neural networks.

In the implementation step, it was aimed to build a small but efficient structure having a low computation cost and complexity, to implicitly extract features of the power consumers from the data set. Implemented deep network structure consists of four 2D convolutional layers, three average pooling layers, two fully connected layers and a softmax layer. It was preferred to construct a relatively less deep network to avoid



(a) Time series of a supermarket's consumption in winter



(b) Time series of a supermarket's consumption in summer

Figure 3.4: 6 days electricity consumption of a supermarket in two different seasons

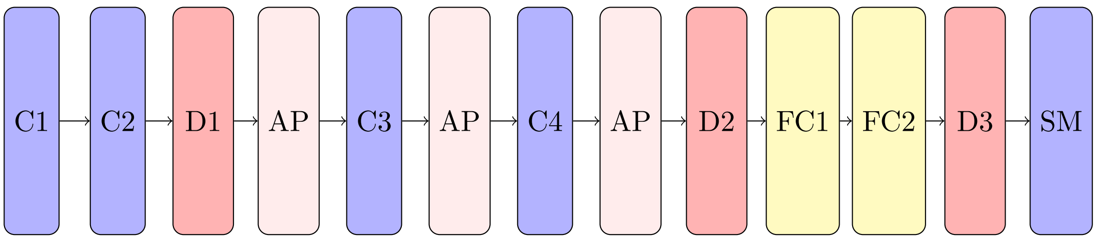


Figure 3.5: Implemented CNN structure

the possibility of overfitting as the input size was not significant when compared to the image classification problems. The pooling layers have helped to decrease computation cost by reducing the matrix size. Also, rectified linear unit (ReLu) has been used as the activation function as it improves the generalization of the network, and speeds up the training. Also, to robust the network three dropout layers have been added using the probability of 0.1, 0.3 and 0.5. Figure 3.5 introduces the overall structure.

The structure starts with applying the first 2D convolutional layer (C1), which includes 24 output channels, to the input. The kernel size which convolves around the input has been specified as 3×3. After each convolutional layer, batch normalization and the activation function ReLu have been applied respectively. Batch normalization has enabled to keep the distribution of the input same. After creating the activation map, the second convolutional layer (C2) having 48 output channels has been applied. After extracting the main features of the input, first two convolutional layers have been followed by a dropout layer (D1) which has 0.1 probability. It dropped some connections of the network out to robust the network. After that, 2D average pooling layer (AP) where the size of the pooling in the x and y directions was $(2, 2)$ has been added. Last two convolutional layers (C3 and C4) having 96 and 100 output channels, have been added afterward and followed by average pooling layers. Before adding two fully connected layers (FC1 and FC2) having 75 and 150 neurons respectively, dropout (D2) with 0.3 probability has been applied. After fully connected layers, a dropout layer(D3) with 0.5 has been applied to improve the robustness, and it has been followed by a softmax layer (SM) to determine the probability of the classes.

For the optimization of the network, the optimizers Adadelta and Adam, that have been mentioned in Section 2.4.2.3 and Section 2.4.2.4, have been used instead of Stochastic Gradient Descent, due to the advantages of adaptive learning methods regarding the memory and computational efficiency. Also, categorical cross entropy which has been mentioned in Section 2.4.1.2 has been taken into consideration as a loss function.

Chapter 4

Performance Analysis

This chapter introduces the results of the performance analysis obtained as a result of several tests that have been performed using convolutional neural networks on Python. In Section 4.1, experiments and test results of the consumer classification in smart grid, carried out on the electricity consumption data set, is presented. In Section 4.2, results of the performance tests conducted on another data set which contains time series, for another application which is the classification of human activities, are introduced to indicate the performance of the deep neural networks on different measurements. In Section 4.3, performance results obtained of the consumer classification with convolutional neural networks are compared with the other classification algorithms.

4.1 Experiments and Test Results

This section explains the description of the experiments executed on the electricity consumption data set and their results to characterize the performance of the network, and to explore the limits of the network. All evaluations have been carried out on Python using Keras deep learning framework. The performance tests have been repeated for several times to have averaged results, to eliminate extreme good or bad results if there were. The available data set has been randomly split into training, test and validation sets in the ratio of 0.6, 0.2 and 0.2 respectively. Training and test sets have contained consumption measurements belonging to the smart buildings (consumers) from different sites, and they do not overlap. The sites randomly chosen from the database can be located in different states and climate zones as well.

The performance of the classification can be characterized by comparing the decision of the deep network to the known information provided in the database. As a result, accuracy has been considered as the only performance metric that describes the number of correctly classified time series divided by the total number of time series.

According to Section 3.1.3, the seasonal impact has been taken into consideration as the most significant impact in the data set and the experiments have been done mostly to observe the model's performance against seasonal changes. In previous work, some other methods have been tried for time series classification using artificial neural networks, and the features of the consumers have been extracted using statistical parameters as well. However, the performance of these methods have been highly influenced by seasonal impacts and demonstrated very low accuracy values in cases of different seasons.

For the performance analysis of convolutional neural network implemented, first, the model has been evaluated for one single season. Second, it has been investigated whether the classification performance was still high when training and test sets were taken from different seasons. Third, length of the input data has been taken into consideration as a limit in the network and it has been investigated that how much input length could be sufficient to classify consumers at a high accuracy level. Besides, the performance has been measured when the number of the classes has been increased based on the seasonal timestamps on the data.

4.1.1 Single Season Performance

Since seasonal conditions affect the behaviors of consumers' electricity consumption, tests have been carried out by distinguishing the data set based on seasonal ranges. 24 days long data has been taken from each of four seasons and the deep networks implemented in Section 3.2 has been trained and evaluated using the data from the same period of the year. The network has been trained with 30 epochs and tested for several times to observe the averaged accuracy. Obtained test accuracy values can be found in Table 4.1.

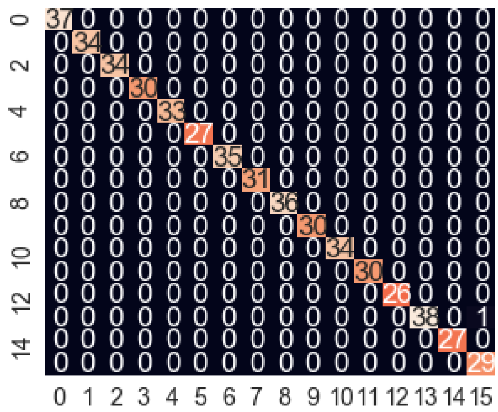
Training & Test Sets	**Test Accuracy**
Winter	99.65%
Spring	99.77%
Summer	98.45%
Autumn	99.45%

Table 4.1: Test results for single season case

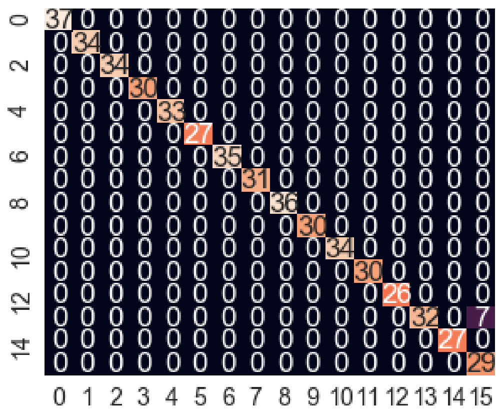
According to the table, it can be observed that the network is capable of solving the classification task at a very high level of accuracy when the test and training sets are taken from the same season. Figure 4.1a, Figure 4.1b, Figure 4.1c and Figure 4.1d summarize confusion matrices in case that both training and test sets have been taken from winter, summer, spring and autumn period respectively (Note that the number of the samples are not equal among the classes according to the database). Only a few of samples belonging to the Outpatient Health Care and Strip Mall classes have been classified incorrectly in this case.

4.1.2 Inter-seasonal Performance

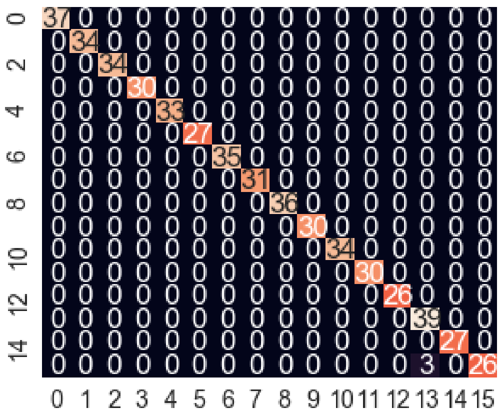
Seasonal changes affect the consumption behavior of the consumers. However, to answer whether it is possible to have a good classification performance using convolutional neural networks independent from seasonal influences, the model has been trained and tested with the data from different seasons. For each season, the month which is in the middle of the season (e.g., April) has been taken into account since it is considered that these months represent better the characteristics of corresponding seasons. In this way, 24 days long data, that covers 576 hours, has been taken from each season. While the CNN



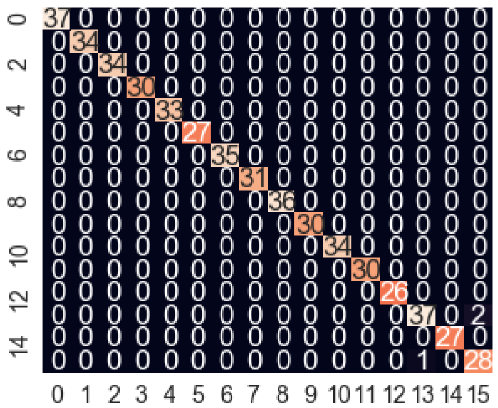
(a) Confusion matrix of classification achieved on 24 days long winter data



(b) Confusion matrix of classification achieved on 24 days long summer data



(c) Confusion matrix of classification achieved on 24 days long spring data



(d) Confusion matrix of classification achieved on 24 days long autumn data

Figure 4.1: Confusion matrices for single season case

model has been trained using the data from each season, the tests have been performed using the samples taken from the other seasons. Performance results are given in Table 4.2a, Table 4.2b, Table 4.2c and Table 4.2d. Tables indicate the accuracy values where the training input has been taken from spring, winter, summer, and autumn respectively.

According to the values in Table 4.2, one can observe that the capability of the network for solving the classification task is still high, even though the accuracy values have dropped when compared to the performance in single-season case. The worst accuracy value belongs to (winter, summer) case, which is 84.78%. It may be expected the performance, in this case, to be lower than others since behavioral patterns of consumers in winter and summer can have significant differences. The results which belong to winter training case are also shown in confusion matrices in Figure 4.2. The matrices indicate that Outpatient Health Care consumer has been misclassified completely.

The confusion matrices for all possible season combinations are given in Figure 4.3, Figure 4.4, Figure 4.5 when training set has been taken from spring, summer and autumn respectively.

Test Sets	Test Accuracy
Spring	92.63%
Summer	84.78%
Autumn	92.16%

(a) Test results in the case that training set is from winter

Test Sets	Test Accuracy
Winter	92.06%
Summer	92.93%
Autumn	97.16%

(b) Test results in the case that training set is from spring

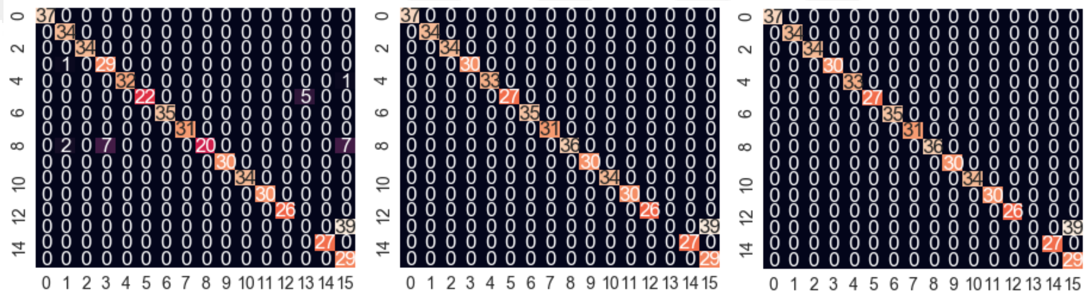
Test Sets	Test Accuracy
Winter	92.39%
Spring	95.85%
Autumn	95.83%

(c) Test results in the case that training set is from summer

Test Sets	Test Accuracy
Winter	91.42%
Spring	99.35%
Summer	94.43%

(d) Test results in the case that training set is from autumn

Table 4.2: Test results for inter-seasonal case

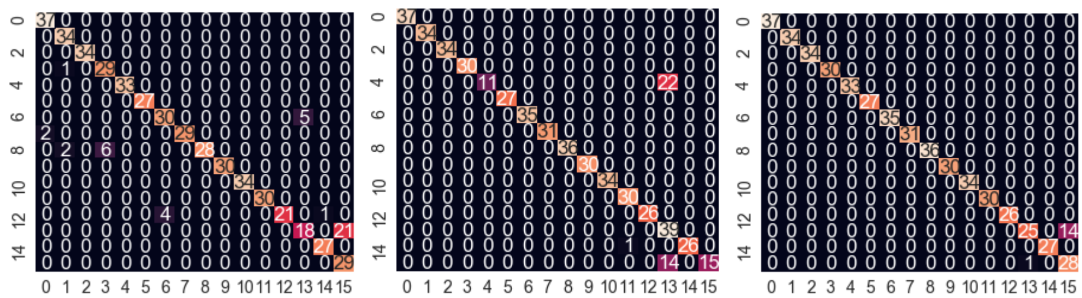


(a) Confusion matrix when the tests have been conducted on summer data

(b) Confusion matrix when the tests have been conducted on spring data

(c) Confusion matrix when the tests have been conducted on autumn data

Figure 4.2: Test results when the training set has been taken from winter



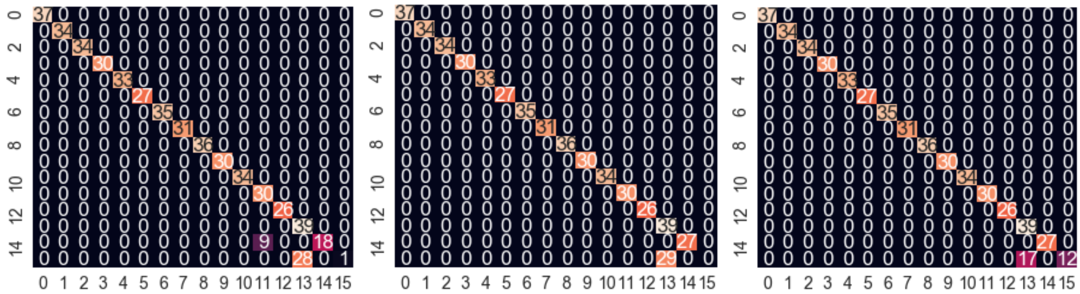
(a) Confusion matrix when the tests have been conducted on summer data

(b) Confusion matrix when the tests have been conducted on winter data

(c) Confusion matrix when the tests have been conducted on autumn data

Figure 4.3: Test results when the training set has been taken from spring

In most of the cases, the network has accomplished the classification task at a high accuracy level, which is higher than 90% except for (winter, summer) case. However, there are classes which can be confused such as strip mall and stand-alone retail, primary and secondary schools. These result demonstrate that consumption behavior of

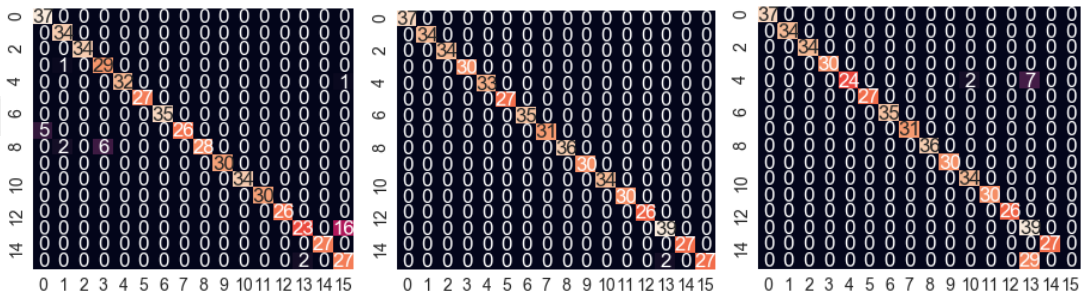


(a) Confusion matrix when the tests have been conducted on summer data

(b) Confusion matrix when the tests have been conducted on spring data

(c) Confusion matrix when the tests have been conducted on autumn data

Figure 4.4: Test results when the training set has been taken from summer



(a) Confusion matrix when the tests have been conducted on summer data

(b) Confusion matrix when the tests have been conducted on spring data

(c) Confusion matrix when the tests have been conducted on autumn data

Figure 4.5: Test results when the training set has been taken from autumn

particularly these consumers have been highly changing according to the seasons.

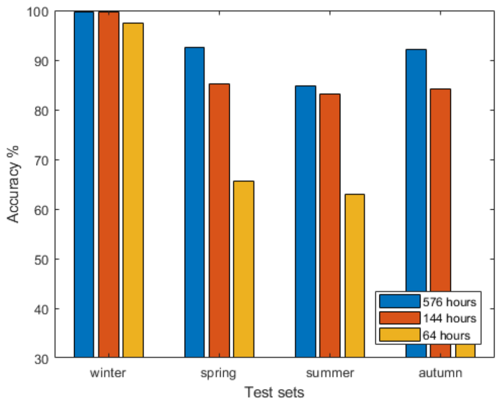
4.1.3 Impact of the Input Length

Data amount given to the network for training has a significant impact on the classification performance. When the time series is considered, it may be estimated that input length covering several months would provide better results. However, it is important to achieve the classification using the minimum amount of input length for the applicability and efficiency.

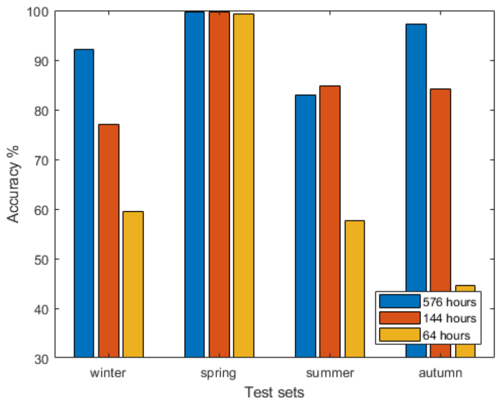
In this section, it has been investigated that how the length of the input data influences the performance of the used deep neural network structure. To observe the limits of the structure, i.e., how many data points are required when the test performance of the solution remains acceptable, the model has been trained using less amount of data by reducing the size of input matrices.

As mentioned in Section 3.1.2, to reduce the input length, time series of the consumers have been resampled in the range of two hours and three hours respectively. After resampling, by taking 144 and 64 hours data from each season, 12×12 and 8×8 rectangular matrices have been created where the rows represent the days and the columns represent two-hour, and three-hour time slot.

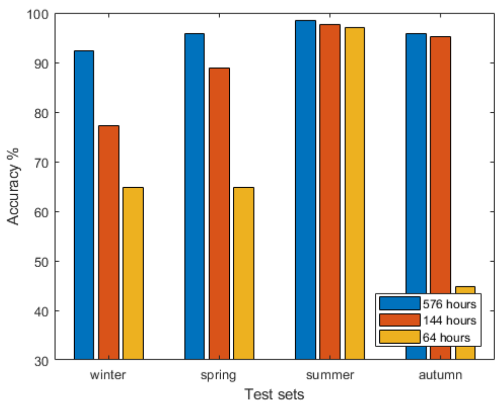
Comparison of the performance results using 576, 144 and 64 hours data is given in Figure 4.6. In most of the cases, one can see that the performance using 576 hours long data is the highest. There is only one exception which is (spring, summer) case where the performance with 144 hours is higher than the performance with 576 hours. Even though in some of the inter-seasonal cases (e.g.: (winter, summer),(summer, autumn)) difference between accuracy values is not significant, in general, the best performance has been obtained in case that 576 hours have been used. When 64 hours data have been used, the performance observed in inter-seasonal cases has dropped radically. Thus, the results of inter-seasonal cases indicate that there are certain limits for the input length, and the less input length the lower classification performance obtained. However, in single-season case, there has not been a dramatic decrease in the classification performance when the input length was changed, and the classification accuracy has remained higher than 95% for 576, 144 and 64 hours long input data.



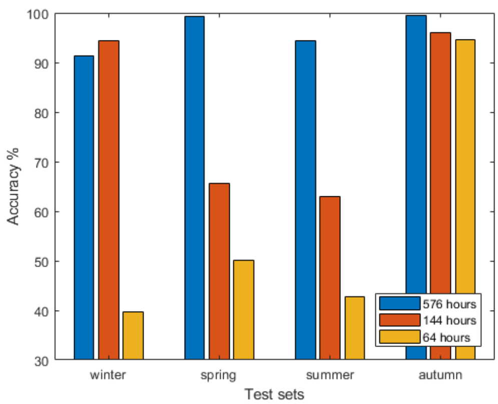
(a) Comparison of the classification performances with 576, 144 and 64 hours long data for winter case



(b) Comparison of the classification performances with 576, 144 and 64 hours long data for spring case



(c) Comparison of the classification performances with 576, 144 and 64 hours long data for summer case



(d) Comparison of the classification performances with 576, 144 and 64 hours long data for autumn case

Figure 4.6: Test results for different input lengths

4.1.4 Performance Analysis for Season Based Classes

In this experiment, it is investigated how the performance will change when the number of the classes are increased based on their seasonal timestamps. For each consumer, four classes based on the seasons (winter, summer, spring, autumn) have been created (e.g., Large Hotel Winter, Large Hotel Spring, Large Hotel Summer, Large Hotel Autumn instead of the class of Large Hotel) and entirely 64 classes have been obtained through 16 consumers. Implemented structure (Figure 3.5) has been tested to classify 64 classes. Tests have been carried out several times to average the results. Performance results show that the classification has been achieved at a very high accuracy, which is 99.46% and only 11 samples have been misclassified among 2048 samples. Figure 4.7 indicates that the accuracy reaches the top after 40 iterations.

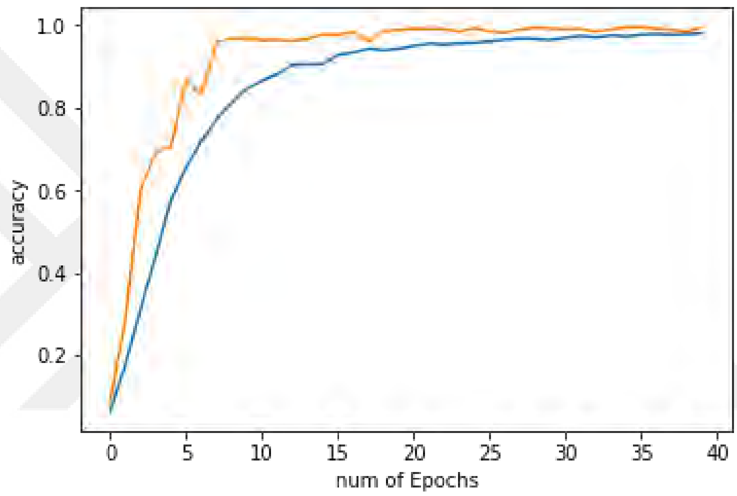


Figure 4.7: Change in accuracy according to the number of epochs

Performance results indicated that when the consumers have been separated based on the seasons, the classification task has not become very complicated even though there were 64 classes. This is because of the significant behavioral differences of the consumers according to the seasons and the capability of the deep neural networks in extracting the features.

4.2 Performance Analysis on Data Set of Human Activities

In Section 4.1, tests have been carried out on electricity consumption data by consumer classification problem in smart grids that is the main issue in this work. Promising performance results which have been obtained on consumer classification bring another question about whether this kind of deep network structure and time series transformation could be a solution for another classification application using time series measurements collected by sensors. To investigate that, a publicly available 'Smart Phone-Based Recognition of Human Activities and Postural Transitions Data Set' has been used [37].

The task which has to be solved using this data is to classify human activities described by the time series data of acceleration and velocity.

4.2.1 Data Set Information

12 basic activities in the data set are labeled as Walking, Walking upstairs, Walking downstairs, Sitting, Standing, Laying, Stand-to-sit, Sit-to-stand, Sit-to-lie, Lie-to-sit, Stand-to-lie, Lie-to-stand. The database contains inertial sensor data which consists of two subsets, which are the measurements captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50 Hz by accelerometer and gyroscope. These measurements have been collected through the experiments fulfilled with 30 different volunteers. For each activity, several experiments have been done with each of the volunteers.

4.2.2 Data Preparation

First, a Wiener filter has been applied to the data set to eliminate noise. 144 samples have been taken from each experiment of users to create square matrices as an input of CNNs as mentioned in section 3.1.2. Since both acceleration and velocity were captured through three coordinates (x,y,z), for each activity 432 samples have been taken. The samples taken from each coordinate have been transformed into 12×12 matrices and added as depth. Finally, $12 \times 12 \times 3$ input matrices have been obtained as indicated in Figure 4.8.

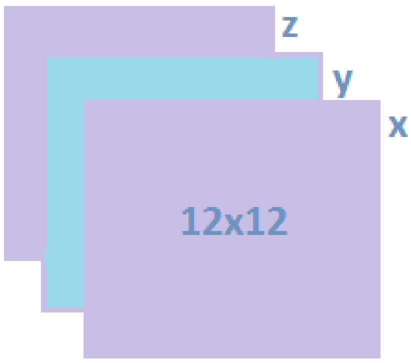


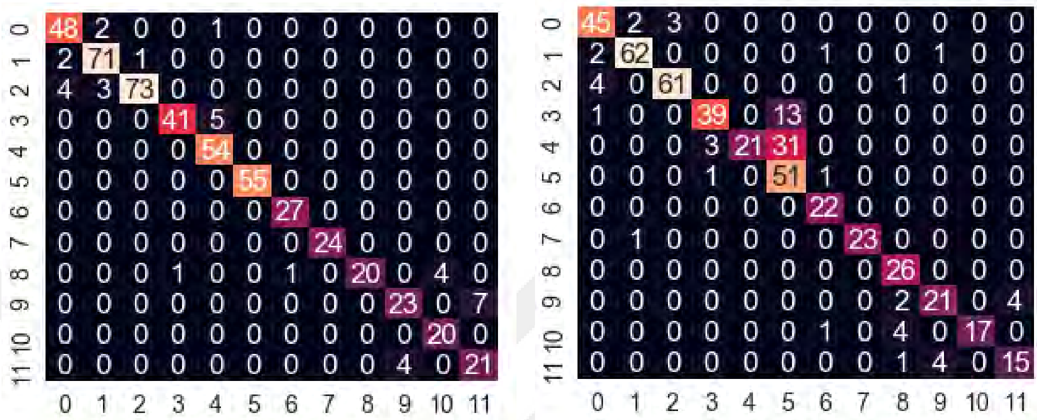
Figure 4.8: Transformation of the data

4.2.3 Training and Test Results

The convolutional neural network implemented in Section 3.2 has been used for classification of human activities. Tests have been carried out on both acceleration and angular velocity data subsets separately. Both data sets contain 1214 vectors in total. This data set has been split into training, test and validation sets in the ratio of 0.4, 0.2 and 0.4 respectively. Performance tests have indicated that the classification was fulfilled at a high level of accuracy, as it can be seen in Table 4.3. Also, Figure 4.9 summarizes the confusion matrices obtained.

Data Subset	Test Accuracy
Acceleration	92.11%
Angular Velocity	87.19%

Table 4.3: Test results on data set of human activities



(a) Confusion matrix obtained on acceleration data

(b) Confusion matrix obtained on angular velocity data

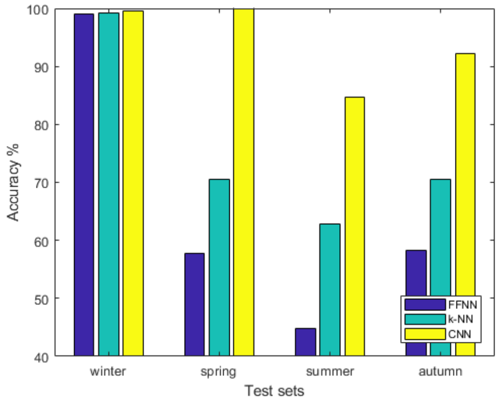
Figure 4.9: Confusion matrices for the classification of human activities

Even though the accuracy values have dropped when compared to the classification performance on the electricity consumption data set, it is still at a high level. These results demonstrate that the convolutional neural networks can be applied to different measurements, for different applications as well. On the other hand, the results can be further improved by preprocessing the noisy data set using appropriate filtering methods.

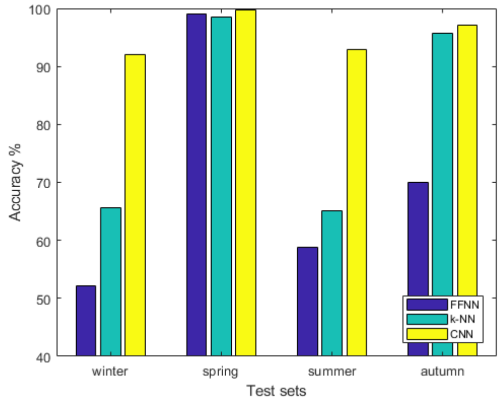
4.3 Comparison of the Related Methods

Section 4.1.1 and 4.1.2 have summarized the performance results obtained using the CNN structure implemented in Section 3.2. To compare these performance results with different algorithms, tests have been carried out on the electricity consumption data set using two main classification algorithms that are feedforward neural networks (FFNN) and k-nearest neighbor (k-NN) algorithms. Figure 4.10a, Figure 4.10b, Figure 4.10c and Figure 4.10d indicate the compared results in the case that training set has been taken from the seasons of winter, spring, summer and autumn respectively.

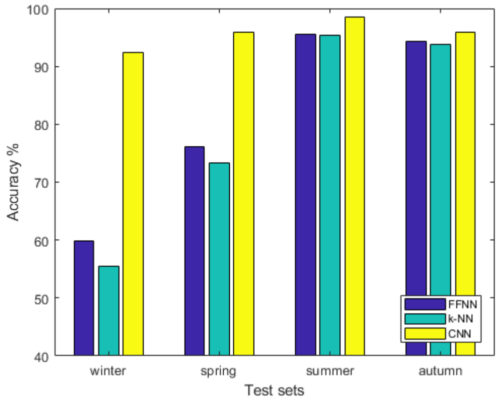
The accuracy values obtained through the compared algorithms have been high and close to each other when the training and test sets have been taken from the same season. However, one can observe that the accuracy values obtained using the CNN structure are closer to 100% in single-season case. On the other hand, the classification performance of the CNN structure is higher than the performances of the other two methods in the



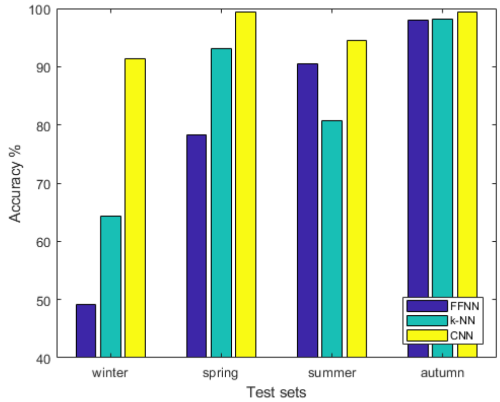
(a) Performance comparison of CNN and other methods when the training set has been taken from winter



(b) Performance comparison of CNN and other methods when the training set has been taken from spring



(c) Performance comparison of CNN and other methods when the training set has been taken from summer



(d) Performance comparison of CNN and other methods when the training set has been taken from autumn

Figure 4.10: Performance evaluation of different algorithms

case when training and test sets have been taken from different seasons. Especially, in some of the cases such as (winter, spring) and (winter, summer) in Figure 4.10a, and (spring, summer) and (spring, winter) cases in Figure 4.10b, one can obviously see that there is a huge difference between performance of the CNN and the other two methods. Thereby, one can say that the CNN solves this classification task at a very high accuracy in most of the cases, and has a highly improved performance in inter-seasonal cases when compared to the two main classification algorithms. In other words, these results indicate that deep neural network based classification is the least influenced method by the seasonal impacts on the data set. This is because that deep neural networks give a better representation of the features belonging to the electricity usage of the consumers. Most importantly, by representing the features, the CNN is able to eliminate the major part of the seasonal effects on the data set.

Chapter 5

Summary

The Internet of Things and Smart Grid have become hot topics due to the need for energy efficiency and automation during the last decade. In the IoT systems and Smart Grid, smart sensors enable to gather a huge amount of measurement data from the connected devices and users. Not only collecting data but also processing the data is significant to have information about the elements of the system. Data processing is the operation to obtain meaningful information from the collected data so that useful planning for a better energy usage can be achieved. There are different size and type of power plants and consumers in smart grids. To achieve a more efficient and better energy distribution and transportation, one of the important tasks using the data measurements collected in the system is to be able to classify the energy consumers accurately and provide the meaningful information to the energy suppliers. The thesis work particularly focused on solving the classification problem of the consumers using a publicly available data set containing electricity consumption measurements belonging to 16 types of consumers (commercial and residential buildings). As well as rapidly increasing data every day, improvements in parallel computing and machine learning allow for data processing to be achieved more accurately. Especially, deep neural networks have become increasingly popular in many applications from image classification to language processing. However, deep neural networks have not been extensively investigated on time series measurements for the solution of this kind of problem. The thesis work focused on processing the measurements in smart grids, more specifically tried to observe the classification capabilities of the CNNs on the data set.

In Chapter 2, the thesis work has reviewed several machine learning algorithms used for clustering and classification applications, as well as the training methods, to obtain a broad background in the methodology.

In Chapter 3, data preparations and implementation steps have been mentioned. It has been tried to transform the time series data into two-dimensional form by building matrices by dividing the time series based on the days. Also, the seasonal impact on the data set has been explained, and for the training and test purposes, the data sets have been divided into four subsets based on the timestamps. Also, design choices and implementation of the CNN structure on Python have been explained in that chapter.

In Chapter 4, the performance results of different tests conducted on the electricity consumption measurements have been presented. One of the purposes of the tests has been to evaluate the performance of the network for a single season and inter-seasonal cases. The performance results have demonstrated that performance for the single season case was higher than inter-season cases. However, one can observe that the classification was achieved at a high accuracy level for both cases. Thereby, it can be said that the seasonal impact has been dropped using deep neural networks. Another experiment was performed to observe the limit of the input length. The results indicate that the method can be deployed using shorter time series in case that the training and testing data are from the same season, but the performance drops further in other cases. Another performance test has been performed in case that the number of the classes were increased based on the seasons. The method accomplished high accuracy for the classification of 64 classes. Tests also have been performed to investigate the performance of the model on another data set for another classification application. This data set contains measurements collected by sensors for the detection of human activities. The performance results indicated that the model was achieved a good performance on the other data set as well. These results can be promising for the solution of similar classification tasks on measurements or time series data using deep neural networks. In the last section of performance analysis, the CNNs have been compared with other two widely used algorithms that are the k-nearest neighbor and feed forward neural networks for both single and inter-seasonal cases, and the results indicated that the CNNs had the best performance for all the cases.

5.1 Conclusion

As deep neural networks have the capability of extracting more specific features and giving a good representation of the characteristics, it is successfully applied to many classification problems. These results we obtained throughout the research indicate that deep neural network based classification can be applied on time series as well, and can demonstrate successful results for the consumer classification problem in smart grid.

Also, the results demonstrate that deep neural networks can be useful to eliminate seasonal characteristics of the data. Even though the current architecture could not eliminate the seasonality, it manages to reduce the dependence on the seasonal trends mostly and demonstrates a classification performance having high accuracy. However, this seasonal influence can be further dropped by transforming the data suitably and using appropriate deep neural network architectures. Also, the results indicate that there are certain limits observed in the required data length, which affects the network size as well. The results also can suggest that in case of consumption time series not only the consumer type but also the time interval (seasons) can be classified. Furthermore, it is observed that deep neural networks can also be applied to the other measurements (time series) for a different classification application.

5.2 Future Work

In the future work, the current deep network structure can be improved or more appropriate deep neural network architectures solving the classification task at a high accuracy level using a minimum length of time series can be tried to find. Different transformation, preprocessing and normalization methods can be applied to the data to eliminate the seasonal effects on time series. Also, it can be investigated whether a neural network based solution can be implemented for clustering of the data. It is expected that by finding the appropriate structure of the neural networks, most of the issues encountered can be solved.

Acknowledgements

First of all, I would like to express my special thanks to my supervisor Dr. Kálmán Tornai for his great support and guidance throughout this research project. His positive outlook and advice inspired and encouraged me so much during the process. I have been so lucky that I have worked with a professor who has always been open to new ideas and has motivated me to do better.

I would like to thank the kind people of the Faculty of Information Technology and Bionics of Pázmány Péter Catholic University for their hospitality and helpfulness during my two years of study in Hungary. I thank the professors of our faculty as they teach us how to be a good researcher and engineer.

Many thanks to my friends in Budapest and Turkey, since they make the difficult moments easier, and the pleasant moments more valuable.

And last but not least, I would like to thank my family who always supports me with their love and belief, who is always with me wherever I am.

Bibliography

[1] C. Floerkemeier and F. Mattern, *The internet of things: first international conference*. Springer, 2008.

[2] N. Jenkins, C. Long, and J. Wu, "An overview of the smart grid in great britain," *Engineering*, vol. 1, no. 4, p. 413–421, 2015.

[3] S. E. Collier, "The emerging enernet: Convergence of the smart grid with the internet of things," *2015 IEEE Rural Electric Power Conference*, 2015.

[4] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid — the new and improved power grid: A survey," *IEEE Communications Surveys Tutorials*, vol. 14, no. 4, p. 944–980, 2012.

[5] P.-N. Tan, M. Steinbach, and V. Kumar, "Data mining cluster analysis: basic concepts and algorithms," *Introduction to data mining*, 2013.

[6] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," 1967.

[7] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[8] J. Kwac, J. Flora, and R. Rajagopal, "Household energy consumption segmentation using hourly data," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 420–430, 2014.

[9] M. A. Carreira-Perpinán, "A review of mean-shift algorithms for clustering," *arXiv preprint arXiv:1503.00687*, 2015.

[10] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda, "Uniqueness of the gaussian kernel for scale-space filtering," *IEEE transactions on pattern analysis and machine intelligence*, no. 1, pp. 26–33, 1986.

[11] D. M. Blei, "Hierarchical clustering," *Lecture Slides, February*, 2008.

[12] B. Zhang and S. N. Srihari, "Fast k-nearest neighbor classification using cluster-based trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 525–528, April 2004.

[13] J. Walters-Williams and Y. Li, "Comparative study of distance functions for nearest neighbors," *Advanced Techniques in Computing Sciences and Software Engineering*, p. 79–84, 2009.

[14] S. Chan, B. Kao, C. Yip, and M. Tang, "A brief survey on sequence classification," in *Database Systems for Advanced Applications, 2003.(DASFAA 2003) Proceedings Eighth International Conference on*, pp. 119–124, IEEE, 2003.

[15] O. Maimon and L. Rokach, eds., *Data Mining and Knowledge Discovery Handbook.* Springer, 2005.

[16] Z. Xing, J. Pei, and E. Keogh, "A brief survey on sequence classification," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 1, pp. 40—48, 2010.

[17] Z. Zakaria, K. L. Lo, and M. H. Sohod, "Application of fuzzy clustering to determine electricity consumers' load profiles," in *2006 IEEE International Power and Energy Conference*, pp. 99–103, Nov 2006.

[18] S. Haykin, *Neural Networks, A Comprehensive Foundation.* Pearson, Prentince Hall, 3rd ed., 2008.

[19] K. L. Lo and Z. Zakaria, "Electricity consumer classification using artificial intelligence," in *39th International Universities Power Engineering Conference, 2004. UPEC 2004.*, vol. 1, pp. 443–447, Sept 2004.

[20] D. Jurafsky and J. H. Martin, *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition.* Dorling Kindersley Pvt, Ltd., 2014.

[21] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, pp. 369–376, ACM, 2006.

[22] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[23] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *Proceedings of International Joint Conference on Neural Networks (IJCNN'11)*, 2011.

[24] S. Srinivas, R. K. Sarvadevabhatla, K. R. Mopuri, N. Prabhu, S. S. S. Kruthiventi, and R. V. Babu, "A taxonomy of deep convolutional neural nets for computer vision," *Frontiers in Robotics and AI*, vol. 2, Nov 2016.

[25] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, pp. 2278–2324, 1998.

[26] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proceedings of the 30 th International Conference on Machine Learning*, 2013.

[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[28] "Optimization: Stochastic gradient descent." http://ufldl.stanford.edu/tutorial/supervised/OptimizationStochasticGradientDescent/. Accessed: 2018-04-29.

[29] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[30] S. Ruder, "An overview of gradient descent optimization algorithms," *CoRR*, vol. abs/1609.04747, 2016.

[31] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[34] "Commercial and residential hourly load profiles for all tmy3 locations in the united states." https://openei.org/datasets/dataset/commercial-and-residential-hourly-load-profiles-for-all-tmy3-locations-in-the-united-states. Accessed: 2017-08-30.

[35] "U.s. climate regions." https://www.ncdc.noaa.gov/monitoring-references/maps/us-climate-regions.php, journal=National Climatic Data Center.

[36] F. Chollet *et al.*, "Keras," 2015.

[37] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017.

Appendices

The CD attached in the diploma contains:

1. The pdf file of the thesis document.

2. Python code of the implemented CNN structure.