



Technical University of Crete

**Prediction using Artificial neural
network and geographic representations
using GIS software of the Urban Heat
Island phenomenon**

by

Kostas Gobakis

kgobakis@isc.tuc.gr

Supervising comity:

Prof. Giorgos Stavrakakis (Advisor)

Prof. Kostas Kalaitzakis

Assi. Prof. Dionysia Kolokotsa

Chania, 5 April 2012

Abstract

The urban heat island phenomenon is mainly caused by the differences in the thermal behaviour between urban and rural settlements that are associated with the thermal properties of urban materials, urban geometry, air pollution, and the anthropogenic heat released by urban activities. The UHI has a serious impact on the energy consumption of buildings, increases smog production, while contributing to an increasing emission of pollutants from power plants, including sulfur dioxide, carbon monoxide, nitrous oxides and suspended particulates.

This thesis presents the applicability of artificial neural networks (ANNs) and learning paradigms for UHI intensity prediction in Athens, Greece. The proposed model is tested using Elman, Feed-Forward and Cascade neural network architecture. The data of time, ambient temperature and global solar radiation are used to train and test the different models. The prediction accuracy is analysed and evaluated.

A new innovative way of visualize the urban heat island using geographic information systems was developed. This will give a better perspective about the problem of urban heat island to the general public.

Preface

A version of Chapter 4 has been published: Gobakis, K., Kolokotsa, D., Synnefa, A., Saliari, M., Giannopoulou, K. & Santamouris, M. (2011). Development of a model for urban heat island prediction using neural network techniques. *Sustainable Cities and Society*, 1(2), 104-115.

Acknowledgement

Firstly I wish to thank Prof. George Staurakakis for his guidance in the completion of this dissertation. His support contribute significantly in the progress of the research. I would also like to thank Prof. Dionysia Kolokotsa for her contribution in the development of the artificial neural networks and her overall instuctions for technical matters. Moreover I would like to thank Prof. Kostas Kalaitzakis for his targeted comments on the dissertation. Last but not least special acknowledgement should be given to my family and my friends for their support during my studies.

The present research was performed in the framework of the FP7 project BRIDGE (SustainaBle uRban plannIng Decision support accountinG for urban mEtabolism). The BRIDGE programme was launched in 2008 ta assist urban planners to design, generate or present alternatives towards a sustainable city.

Contents

1	Introduction	1
1.1	Urban heat island	1
1.2	Scope of the thesis	4
1.3	Structure of the thesis	5
2	Literature Review	6
3	Methodology and measurements	10
3.1	Experimental site description	10
3.2	Data collection	13
4	Prediction using Artificial Neural Network	16
4.1	Artificial Neural Network	16
4.1.1	Introduction	16
4.1.2	Brief historical overview	17
4.1.3	Biological neuron	18
4.1.4	Artificial neuron	19
4.1.5	Artificial neural networks	23
4.1.6	Basic characteristics of neural networks	24
4.1.7	Architecture of neural networks	25
4.1.8	Key features of neural networks	25
4.1.9	Benefits from using neural networks	26

4.1.10	Disadvantages of neural networks	27
4.1.11	Learning and recall	28
4.1.11.1	Introduction	28
4.1.11.2	Knowledge representation	29
4.1.11.3	Learning with error correction	31
4.2	Experimental procedure	33
4.2.1	Creation of data sets	33
4.2.2	Architectures of ANN used in our case study	35
4.2.2.1	Feed-Forward	36
4.2.2.2	Elman	37
4.2.2.3	Cascade	37
4.2.3	Training algorithms of neural networks used in our case study	39
4.2.3.1	Levenberg-Marquardt	39
4.2.3.2	Scaled conjugate gradient	39
4.2.3.3	Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton	40
4.2.3.4	Gradient descent	40
4.2.3.5	Gradient descent with momentum and adaptive learning rate	41
4.2.3.6	Resilient back propagation	41
4.2.3.7	Comparison between different neural network architecture and training algorithms	42
4.2.4	Results	46
5	Interpolation using Geographic Information System	55
5.1	Introduction on Geographic Information System	55
5.1.1	Introduction	58
5.1.2	Representation discrete objects - Vector models	60
5.1.2.1	Point entities	60
5.1.2.2	Linear entities	60

5.1.2.3	Polygonal entities	60
5.1.2.4	Representation of continuous space - raster model	61
5.1.2.5	Triangular model (Triangulated Irregular Network)	62
5.1.2.6	Coordinates system	63
5.1.3	Interpolation using Geographic Information System	66
5.1.3.1	Methods of interpolation	67
5.1.3.1.1	Inverse Distance Weighted	67
5.1.3.1.2	Spline	68
5.1.3.1.3	Kriging	68
5.1.3.2	Data preparation	69
5.1.3.3	Analysis	71
5.1.3.3.1	Interpolation methods comparison	71
5.1.3.3.2	Data representation	72
6	Summary	77
6.1	Future research prospects	77
6.2	Conclusion	77
A	Python code for Kriging Spherical interpolation method	79

List of Figures

1.1	Summer and winter air temperatures in the central business district (urban) and airport (rural) of Melbourne, Australia	3
1.2	Summer and winter differences in air temperature between the central business district and the airport of Melbourne, Australia	4
3.1	The location of the 14 meteo station in the GAA	11
3.2	Schematic presentation of the eight synoptic categories over GAA	14
4.1	Typical biological neuron	18
4.2	Artificial neuron	20
4.3	Activation function graphs	22
4.4	Equivalent natural-artificial neuron	22
4.5	Structure of a Artificial Neural Network	23
4.6	Schematics diagram for the Feed-Forward neural network.	36
4.7	Equivalent natural-artificial neuron	38
4.8	Equivalent natural-artificial neuron	38
4.9	Comparison between the three different ANN for 1-h prediction horizon .	45
4.10	Comparison between the three different ANN for 24-h prediction horizon	46
4.11	Measured-predicted temperatures for 19-24/06/2009	47
4.12	Measured-predicted temperatures for 05-09/06/2009	48
4.13	Monthly comparison for measured-predicted temperature for Haidari experimental station	49

4.14	Monthly comparison for measured-predicted temperature for Agia Barbara experimental station	50
4.15	Response of the ANN to weather changes	50
4.16	The UHI in Athens during 18/06/2009	51
4.17	The UHI in Athens during 01/07/2009	52
4.18	The measured UHI versus predicted UHI 24h for the Agia Barbara site . .	53
4.19	The measured UHI versus predicted UHI 24h for the Egaleo site	53
4.20	The measured UHI versus predicted UHI 24h for the Halandri site	54
5.1	The components of a GIS	56
5.2	Polygonal, linear and point entities in a neighbourhood	61
5.3	Raster Data	62
5.4	Triangulated Irregular Network	64
5.5	The azimuthal, the conical and cylindrical projection	66
5.6	Location of meteorological stations	70
5.7	Spatial database	70
5.8	Data structure for interpolation	70
5.9	Temperature interpolation 12/06/2009 05:00	73
5.10	Temperature interpolation 06/07/2009 16:00	74
5.11	Temperature interpolation 20/08/2009 11:00	75

List of Tables

3.1	The location of the 14 experimental sites	15
4.1	Snapshot from the input dataset-one hour prediction horizon	34
4.2	Performance comparison of difference training functions for Koridalos site.	43
4.3	Performance comparison of difference training function for Peristeri site. .	44
4.4	Performance comparison of three ANN architectures for Koridalos site. . .	45
4.5	Seasonal variations of standard deviation between measured and predicted temperatures for Agia Barbara and Haidari sites.	49
5.1	Interpolation Comparison	72

Abbreviations and Acronyms

ANN	Artificial Neural Networks
NN	Neural Networks
VLSI	Very-large-scale integration
UHI	Urban Heat Island Intensity
GAA	Greater Athens Area
GIS	Geographic Information Systems
TIN	Triangulated Irregular Network
WGS84	World Geodetic System 1984
UTM	Universal Transverse Mercator
GYS	Hellenic Military Geographical Service
GGRS 87	Greek Geodetic Reference System 1987
GPS	Global Positioning System
IDW	Inverse Distance Weighted
MEA	Mean Absolute Error

Chapter 1

Introduction

1.1 Urban heat island

Urban development leads to radical land cover change. As cities expand into surrounding forests, grasslands, and deserts, the natural cover is replaced with roads, buildings, parks, and gardens. The environmental implications of this change are often subtle, as in primitive or sparsely populated settlements, but in most modern cities the implications are dramatic and long term. Land cover change in cities has significant effects on local climate: temperature, precipitation, humidity, wind, and, to a lesser extent, cloud and radiation are all noticeably different between a city and its countryside. These differences are sufficiently well documented in scientific literature that climatologists regard cities as having unique local climates, much like lakes, valleys, and coastlines. One of the most critical variables that best distinguishes city and country climates is the air temperature. Substantive research points to the conclusion that cities are warmer, on average, than their natural surroundings. The region of warmth associated with cities is known as an urban heat island (UHI). In brief, the primary causes of heat island are related to a city's thermal, moisture, and radiation properties, all of which are markedly different from those of the country. Heat islands are formed in urban and suburban areas because of the fact that many common construction materials absorb and retain

more of the sun's heat than natural materials in less-developed rural areas. There are two main reasons for this heating. First, most, most urban building materials are impermeable and watertight, so moisture is not readily available to dissipate the sun's heat. Second, dark materials in concert with canyon like configurations of buildings and pavement collect and trap more of solar energy. Temperatures of dark, dry surfaces in direct sun can reach up to $88^{\circ}C$ during the day, while vegetated surfaces with moist soil under the same conditions might reach only $18^{\circ}C$. Anthropogenic heat, or human produced heat, slower wind speeds and air pollution in urban areas also contribute to heat island formation.

Heat island phenomenon contributes to human discomfort, health problems, higher energy bills and increased pollution. On top of the effects of global warming, heat island is further reducing the habitability of urban and suburban areas. Considering that more than 75 per cent of the world's population lives in these areas ([26]), heat island impacts are extremely consequential.

Heat islands exhibit five common characteristics:

1. When compared to undeveloped, rural areas, a heat island is warmer in general, with distinct daily patterns of behaviour. Heat islands are often warmer, in relation to rural surroundings, after the sun goes down, and cooler after the sun rises. Urban air in the 'canopy layer', below the tops of trees and buildings, can be as much as $6^{\circ}C$ warmer than the air in rural areas.
2. Air temperatures are driven by the heating of urban surfaces, since many man-made surfaces absorb more of the sun's heat than natural vegetation does.
3. These differences in air and surface temperatures are enhanced when the weather is calm and clear.
4. Areas with the least vegetation and greatest development tend to be hottest, and heat islands tend to become more intense as cities grow larger.

5. Heat islands also display warmer air in the ‘boundary layer’, a layer of air up to 2000 metres high. Heat islands often create large plumes of warmer air over cities, and temperature inversions (warmer air over cooler air) caused by heat islands are not uncommon.

Heat islands have air temperatures that are warmer than temperatures in surrounding rural areas. The difference between urban and rural air temperatures, also called the heat island intensity, is often used to measure the heat island effect. This intensity varies throughout the day and night. In the morning, the urban–rural temperature difference is generally at its smallest. This difference grows throughout the day as urban surfaces heat up and subsequently warm the urban air. The heat island intensity is usually largest at night, since urban surfaces continue to give off heat and slow the rate of night-time cooling.

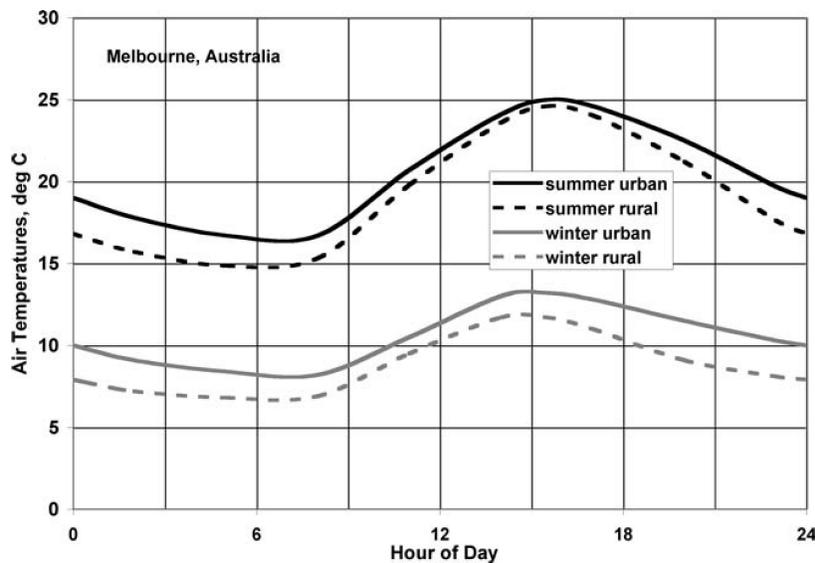


Figure 1.1: Summer and winter air temperatures in the central business district (urban) and airport (rural) of Melbourne, Australia

Source: ([20])

Figures 1.1 and 1.2 show air temperatures and heat island intensity for typical summer and winter days in a heat island. Figure 1.1 plots daily variations in air tem-

perature in the central business district and at the airport of Melbourne, Australia [20]. These daily profiles are averaged from hourly data for December 1997 and for January and February 1998 (summer) and for June, July and August 1998 (winter). This plot shows that temperatures are always warmer in the central business district than they are at the airport. From Figure 1.2, which plots the difference between the urban and rural air temperatures, it is seen that the heat island is strongest at night [2.4°C differential at 8:00pm in winter, 2.2°C at midnight in summer] and weakest during the day [1.0°C at 11:00am in winter, 0.4°C at 3:00pm in summer].

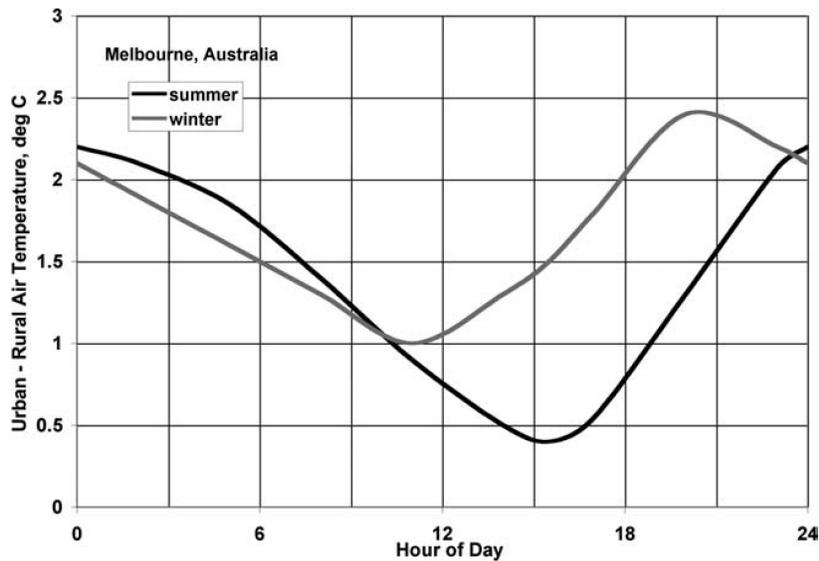


Figure 1.2: Summer and winter differences in air temperature between the central business district and the airport of Melbourne, Australia

Source: ([20])

1.2 Scope of the thesis

The aim of the thesis is the prediction and visualisation of the UHI intensity in the greater Athens area. The prediction of the phenomenon is achieved using artificial neural network. Different types and training methods of ANN are investigated in order to

achieve the optimal prediction using small amount of data. The Geographic Information System are used for the visualisation of the phenomenon using interpolation methods in order to construct the temperature value for all intermediate points between the meteorological stations. The final result is a video-animation in which each frame visualise one interpolation image, that shows the time evolution of the phenomenon.

1.3 Structure of the thesis

The thesis divides logically into 6 chapters:

- Chapter 2 cover background material to the urban heat island effect and intensity prediction.
- Chapter 3 presents the experimental site and data collection.
- Chapter 4 describes the urban heat island intensity prediction using artificial neural network.
- Chapter 5 describes the interpolation of temperature using geographic information system.
- Chapter 6 provides a summary and concludes this thesis.

Chapter 2

Literature Review

Urban heating has been considered a major problem of the cities' inhabitants. The specific characteristics of urban-structures enable them to capture, store and release higher quantities of heat as compared to their counterparts in rural areas. The presence of normally abundant sources of anthropogenic heat in urban areas is the second driving force of urban heating. The urban heating, caused by the specific characteristics of urban structures and anthropogenic heat sources, increases urban area temperatures as compared to surrounding rural areas. Due to the severity of the problem, vast research effort has been dedicated in order to measure and predict the UHI intensity.

In the study by Yi and Prybuton [29], neural network model used for prediction of daily maximum ozone concentration in Dallas-Fort Worth area. Ozone in urban areas varies with the meteorological and vehicle emission parameters. The meteorological parameters with the highest correlation to ozone concentrations include maximum temperature, wind speed, wind direction, sky cover, humidity, and mixing height. The data used in this study contain the average hourly ozone measurement and the average hourly meteorological measurements for variables such as temperature, wind speed, and wind direction. A standard three-layer ANN model with nine inputs and four hidden nodes, found to be superior the statistical methods.

According to a study by Jiang [10], a three-layer ANN with 17 inputs was developed

to predict the air pollution levels (SO_2 , NO_2 and PM_{10} ¹) of cities in China. As results from this study an Air Pollution Index (API) reporting system, based on health effects, was introduced for a consistent comparison among the pollution levels by different air pollutants. The case study area was Shanghai but due the greatest prediction accuracy from the previous model and the inputs to the model were not site-specific, allowing the model to be applied to a number of locations across China. Another conclusion from this study was that the needed dataset for proper training the ANN was at least 1 year data.

In the study by Maqsood, Khan and Abraham [15], air temperature, wind speed and relative humidity in Saskatchewan Canada were predicted for 24 h in advance by ANN. They found that combining the outputs of a standard Feed-Forward ANN, a recurrent ANN, a radial basis function network and a Hopfield network into a simple "winner-take-all" ensemble led to more accurate predictions of wind speed, relative humidity and air temperature than any of the individual component network. The meteorological data used in this study was only from one year (2001) and was divided into 4 different datasets representing the four seasons of the year. For each season of the year a complete different set of ANN was developed, achieving the best accurate temperature prediction on winter time and the worst at fall.

According to the study by Smith, Hoogenboom and McClendon [25], air temperature prediction model for prediction horizons of 1 to 24 hours using Ward-style ANNs. The prediction mean absolute error (MAE) for a year-round evaluation set ranged from $0.516^{\circ}C$ at the one-hour horizon to $1.873^{\circ}C$ at the twelve-hour horizon. The researchers use for training the ANN data from the year 1997 to 2000 and consisted of approximately 1.25M measurements. Ward-style ANN use 258 inputs, including air temperature, wind speed, relative humidity, solar radiation, and rainfall and hourly rates of change at the time of prediction as well as the history of prior observations at 1-h intervals going back 24 h. Also among the models' inputs were four cyclic time-of-day and four cyclic day-of-year terms. Also the above model were integrated into a general

¹particulate matter $< 10\mu m$

decision support system.

Tasadduq, Rehman, and Bubshait [27] used a back propagation ANN with batch learning scheme for 24-h prediction in ambient temperature on a coastal location in Saudi Arabia. They found that temperature can be predicted even with only one input with good accuracy. The training dataset for the ANN was one year's of hourly temperature values. For the evaluation of the ANN they used data from 3 different years with mean absolute error of $2.83^{\circ}C$.

Additionally, a number of urban heat island prediction studies are based in the ANN technology. In the study by Mihalakakou, Santamouris, Papanikolaou, Cartalis and Tsangassoulis [19], a neural network architecture was developed to predict the urban heat island intensity in Athens during both day and night period. The selected neural network architecture consists of one hidden layer of sixteen to twenty seven log-sigmoid neurons follows by an output layer of one linear neuron for predicting the night time UHI and one hidden layer of eighteen to twenty two log-sigmoid neurons follows by an output layer of one linear neuron for predicting the day time UHI. For every one of the twenty three experimental station two neural networks were constructed one for day and one for night time prediction. Data from two years were used in order to train and test the accuracy of the prediction. The mean square error for UHI intensity prediction during the night were 0.2 to $0.3^{\circ}C$ while for the day were 0.1 to $0.3^{\circ}C$.

In the study performed by Kolokotroni, Davies, Croxford, Bhuiyan, Mavrogianni [12], a validated method for predicting air temperatures within the UHI at discreet locations based on input data from one meteorological station for the time the prediction is required and historic measured air temperatures within the city. It uses London as a case-study to describe the method and its applications. The described prediction model comprises of a suite of ANN models to predict site specific hourly air temperature within the Greater London Area. The model was developed using a feed forward back-propagation ANN model with one hidden layer with seventeen neurons, based on hourly air temperature measurements at 77 fixed temperature stations and hourly meteorological data (off-site variables) from Heathrow. The hourly meteorological data

required for the predictions are air temperature, relative humidity, cloud cover, wind speed and global solar radiation.

Chapter 3

Methodology and measurements

3.1 Experimental site description

The Greater Athens Area (GAA) is situated on a small peninsula located on the southeastern edge of the Greek mainland (Figure 3.1). It is divided by high mountains in three main parts, which are connected by small openings. The central part is the Athens basin which covers an area of 450km^2 , with a population density of 8000 inhabitants per square kilometer, with the main axis orientated from South-SouthWest to North-NorthEast. Athens basin is surrounded by high mountains in the north (Parnitha, 1426 m), in the west (Egaleo, 458 m) and in the east (Hymettus, 1026 m and Penteli, 1107 m), while it is open to the sea in the south (Saronikos Gulf). The other parts of the Athens area are the Thriassion plain west of the Athens basin and the Mesogia plain in the east. There are only small openings through which the Athens basin communicates with these plains as well as the rest of Greek mainland. These openings play an important role in air mass exchange between the Athens basin and the Thriassion and Mesogia plains. The city of Athens is characterised by a strong heat island effect, mainly caused by the accelerated industrialisation and urbanisation during recent years. From previous measurements' analysis is found that maximum heat island intensity in the Athens centre is almost 16°C while the mean value for the major central area of Athens

reaches 12°C . Also, absolute maximum temperatures in the central area is close to 15°C higher than in the suburban areas, while absolute minimum temperatures are up to 3°C higher in the centre [16].



Figure 3.1: The location of the 14 meteo station in the GAA

A manual scheme for classifying the day-by-day 850-hPa atmospheric circulation over GAA was proposed and employed by [11]. According to this study, the synoptic categories, illustrated in Figure 3.2, are the following :

- Long-wave trough, characterised by intense winds, especially during the cold period of the year (Figure 3.2a).
- South-Westerly flow, characterised by a South-Westerly flow which is usually very strong (Fig. 3.2b). North-Westerly flow, characterised by strong cold air advection from the north or North-West (Figure 3.2c).

- Zonal flow, characterised by a Westerly flow with considerably lower intensity in the warm period of the year (Figure 3.2d).
- Closed low, characterised by the presence of a closed low accompanied by intense winds, usually from the Northern sector and rainfall (Figure 3.2e).
- High pressure ridge, characterised by a weak pressure gradient and weak, variable winds or calm conditions (Figure 3.2f).
- Closed anticyclone, characterised by the presence of a closed anticyclone that extends over the Greek area accompanied by weak winds from the southern or northern sector (Figure 3.2g).
- Category high-low, characterised by strengthening of the pressure gradient and strong North-Easterlies that blow over the Aegean Sea and into the GAA (Figure 3.2h).

The anticyclonic circulation prevails over the Athens area with maximum occurrence in January and June. A significant meteorological feature of the area is the predominance of high pressure systems combined with low pressure ones, resulting in rather complicated flow regimes, especially during July and August. Conditions of a cyclonic type seem to dominate in February and March while a south-westerly flow prevails in November and April. The prevailing anticyclonic circulation in the Athens area favours the strong development of the heat island phenomenon.

In a study performed by Livada, Santamouris, Niachou, Papanikolau and Mihalakakou[14] reporting the results of the heat island study in Athens, it is found that near the sea, the air temperatures are higher in the cold period due to the influence of the sea which supports the maintenance of high air temperatures. It is also reported that high air temperatures during the hot period of the year or low air temperatures in the cold period is mostly related to the synoptic weather conditions and it cannot reasonably be considered as an index for the heat island effect development. The increase of the cooling load in

Athens and the ecological footprint of urban heat island is studied by Santamouris, Papaniaris, Michalakakou[23] and Santamouris, Pavlou, Synefa, Nichou, Kolokotsa [24].

Given the actual penetration of air conditioning in the country, the ecological footprint due to the heat island ranges 1.5–2 times the city’s surface area. Moreover the maximum potential ecological footprint provided that all buildings are air conditioned is almost 110,000 ha. The cost to compensate the heat island is calculated close to 4.13 M€/year or 164 €per household. The additional peak cooling electrical load to compensate the heat island is 82.4 MW.

3.2 Data collection

In the present effort to predict the urban heat island effect in the area of Athens, a network of 14 meteorological stations has been set up corresponding to the 13 Athens municipalities plus the reference station (Table 3.1). The meteorological stations are placed on the administrative municipalities buildings and are all 2 m above ground, North oriented, shaded and ventilated. Each meteorological station contains a fully calibrated high precision data logger (Tiny Tag data loggers) that measures air temperature every 15 min. The sensors’ characteristics are:

- Reading resolution $0.02\text{ }^{\circ}\text{C}$ or better.
- Range $-40\text{ }^{\circ}\text{C}$ to $+125\text{ }^{\circ}\text{C}$.
- Temperature stability $\pm 0.01\text{ }^{\circ}\text{C} / ^{\circ}\text{C}$ change from $25\text{ }^{\circ}\text{C}$.

In addition, other meteorological data (solar radiation, wind velocity, etc.) are collected from the National Observatory of Athens located at Thission, Athens . The specific site is in a greenery area and is considered as the reference station of the overall analysis although it is positioned almost in the centre of the peninsula. The experimental period started on April 2009 in the framework of BRIDGE project (www.bridge-fp7.eu).

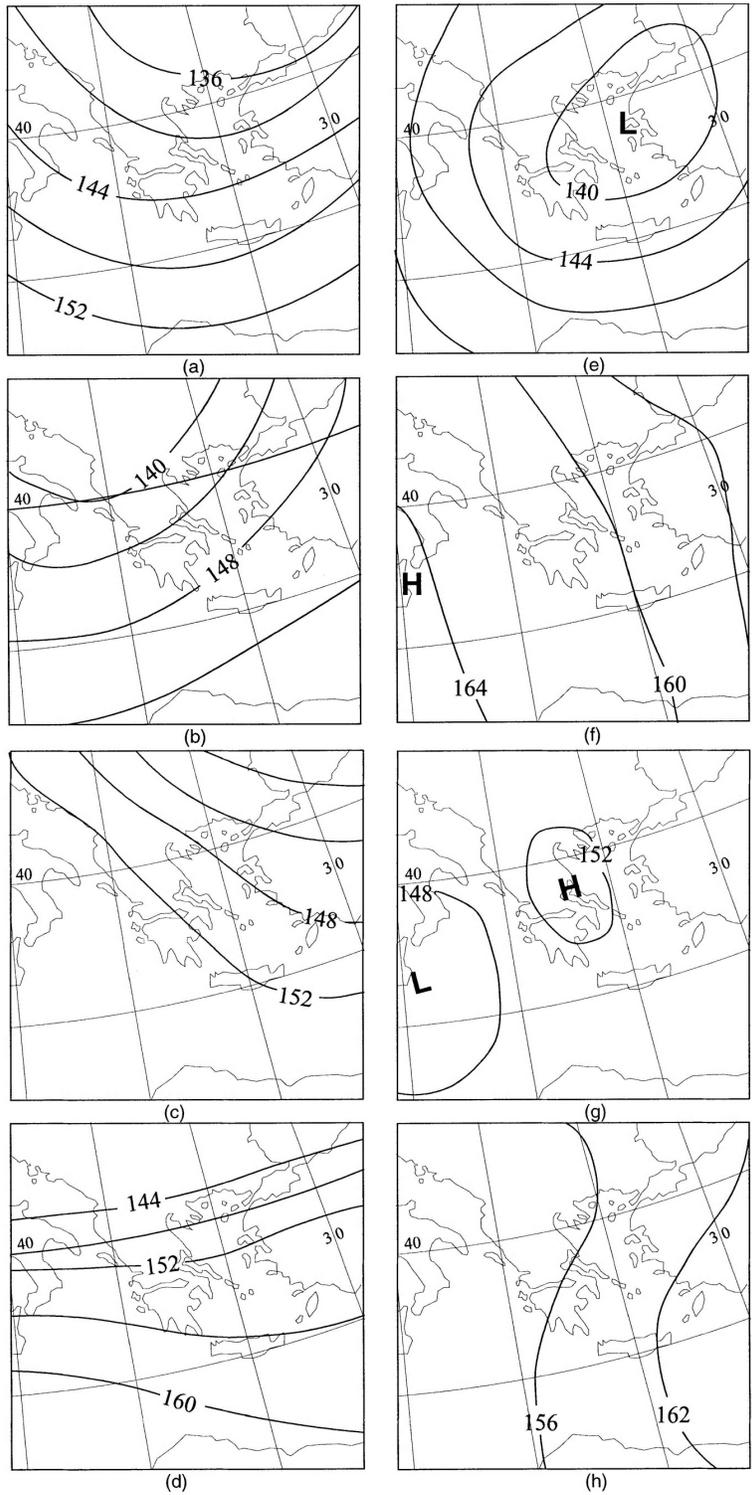


Figure 3.2: Schematic presentation of the eight synoptic categories over GAA

Source: ([11])

The present analysis uses data for one-year period (from April 2009 until May 2010) targeting to minimise the need for long term historic data.

i	Municipality	LATITUDE	LONGITUDE
1	Egaleo	37°59'50"	23°40'5"
2	Korydalos	37°58'45"	23°38'33"
3	Haidari	38°0'45"	23°39'35"
4	Ag. Varbara	37°59'22"	23°39'37"
5	Peristeri	38°0'47"	23°41'43"
6	Kamatero	38°3'35"	23°42'50"
7	Zefyri	38°4'7"	23°43'4"
8	Ilioupoli	37°55'58"	23°45'29"
9	Petroupoli	38°2'26"	23°41'16"
10	Agii Anargyri	38°1'34"	23°43'3"
11	Xalandri	38°0'44"	23°39'34"
12	Ilion	38°1'54"	23°42'27"
13	Kaissariani	37°58'8"	23°45'41"
14	National Observatory of Athens (reference site)	37°58'24"	23°43'5"

Table 3.1: The location of the 14 experimental sites

Chapter 4

Prediction using Artificial Neural Network

4.1 Artificial Neural Network

4.1.1 Introduction

Conserving the issue solving complex problems various systems have been developed. Some belong to the conventional approaches and others in the category of 'smart' systems. The inspiration came from biological neural systems that try to simulate. These systems have made great progress and development and successfully applied to many problems. Researchers from various fields design artificial neural networks for solving various problems related to pattern recognition, forecasting, optimisation, and the auxiliary memory and control.

Conventional approaches have also been proposed for solving such problems. But though they have been successful applications in a well-structured environments, none of which was flexible enough to be successfully implemented in the same environment partially disordered. The advantage of artificial neural networks that generalise about the existing application and therefore can give answers to the problem even when the

initial data change.

4.1.2 Brief historical overview

The research on neural networks went through three periods of growth. The first wave of interest around them, then known as interconnection models or models of parallel distributed processing, emerged in 1943 from the nerve-biologist McCulloch[17] and mathematician specialising in statistics Pitts who made the introduction of simplified neurons in the publication "A Logical Calculus of Ideas Immanent in nervous activity". These neurons were presented as models of biological neurons and as fundamental structures in a chain that would perform calculations. This post inspired John von Neumann to develop a new digital computer or an electronic brain as he called it.

The second period was in the 1960 from the computer scientist Frank Rosenblatt that motivated from the above work, investigated the observations which led to the genesis of the first neural network known as the perceptron and the convergence perceptron theorem in 1962. Here is the book by Mincky and Papert "Perceptrons: An introduction to computational Geometry" in 1969 which shows the constraints on the simple perceptron. Their results were soothing for the enthusiasm that existed from researchers around the subject and especially to the community of computer science. The recession lasted for about 20 years.

In the early 1980's research and interest in these networks showed a significant recovery. The main result behind this development includes a new approach by Hopfield in 1982 [28] and the algorithm "back-propagation algorithm for multilayer perceptrons (multilayer feed-forward networks)" first proposed by Werbos, and then re proposed several times until published by Rumelhart [3] in 1986.

Anderson and Rosenfeld[2] have reported an extensive history surrounding the development of artificial neural networks.

4.1.3 Biological neuron

The human ability to think, remember and solve problems is identified in the brain. As is known from Biology, the structural unit of the brain is the neuron. A typical biological neuron (Figure 4.1) consists of the soma is the nucleus, the dendrites through which signals from neighboring neurons (entry points) and the axon is the output of the neuron and the means of connecting with other neurons. Between two dendrites, one from the axon and one from the soma, is a tiny gap between called a synapse. Synapses by chemical processes accelerate or retard the flow of electrical charges in the body of the neuron. The learning ability and memory, showing on the brain is due to the ability of synapses to alter their conductivity. The electrical signals that enter the body through the dendrites are combined and if the result exceeds some threshold the signal propagates through the axon to other neurons.

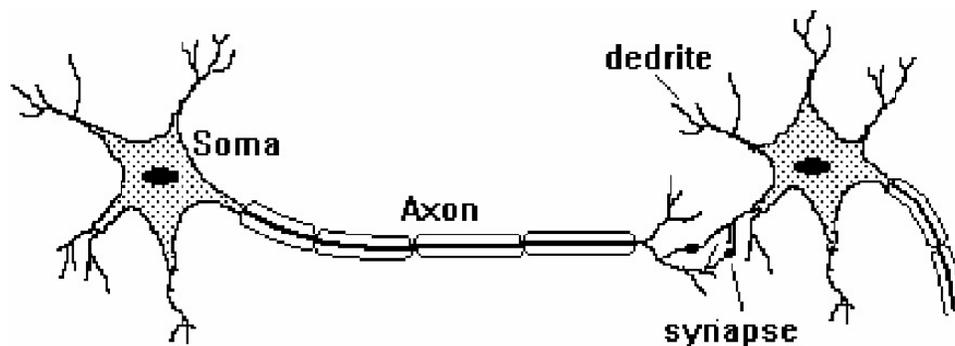


Figure 4.1: Typical biological neuron

The brain of a newborn human consists of about 100 billion neurons, each of which is associated with about 1000 other neurons. This is done through the axon of each neuron in which lead an equal number of dendrites of other neurons. Since any such connection includes a synapse there are about 100 trillion synapses (100 000 000 000 000), which affect the brain function. It is obvious that any attempt to copy the structure and functioning of the brain at such scale is impossible. In fact, models which include built thousands of artificial neurons, are highly artificial one million synapses have very limited functionality.

Although the response time of biological neurons is in the order of milliseconds (msec), but the brain is able to take amazingly complex decisions quickly. In a sense, this is because the computing capacity of the brain and the information it contains is shared throughout the volume. It is about a parallel and distributed computing system. These characteristics are the main motive behind the desire to model the human brain with artificial neural networks.

4.1.4 Artificial neuron

The artificial neuron is a computer model whose parts are assigned directly to those of the biological neuron. A neuron is a processing information unit which is fundamental to the functioning of the NN. In Figure 4.2 we see the model of a neuron and we can observe three basic elements:

1. A set of synaptic connections, which each one is characterised by a weight. Namely a variable-signal x_j at the entrance of the synapse j that is connected to the k -neuron multiplied by the weight of the synapse w_{kj} . It should explain that the first of the indicators in each weight refers to the neuron which is the weight belongs and the second index refers to the neuron from which the input vector originate. The weight w_{kj} is positive if the synapse is excited and negative if it is inhibitory.
2. An aggregate link that adds the input signals after they have been weighted by the weights of the synapses of the neuron. The operations are described to this point make a linear combination.
3. An activation function that reduces-normalizes the field width of the output neuron in a finite field is usually the time $[0, 1]$ or $[-1, 1]$.

Neurons usually include an outer threshold θ_k which has the capacity to reduce the input of the activation function. Also the input of the activation function can be increased by introducing a bias in the position of the threshold, ie the bias makes the inverse energy than the threshold.

In mathematical terms we can describe a neuron k by the following equations:

$$u_k = \sum_{j=1}^p w_{kj} x_j \quad (4.1)$$

$$y_k = \phi(u_k - \theta_k) \quad (4.2)$$

where

- x_1, x_2, \dots, x_p is the signals-input variables
- $w_{k1}, w_{k2}, \dots, w_{kp}$ are the synaptic weights of neuron k
- u_k is the linear combination
- θ_k is the threshold
- $\phi()$ is the action function (linear or non linear)
- y_k is the output of the neuron

There are three typical activation-transfer function:

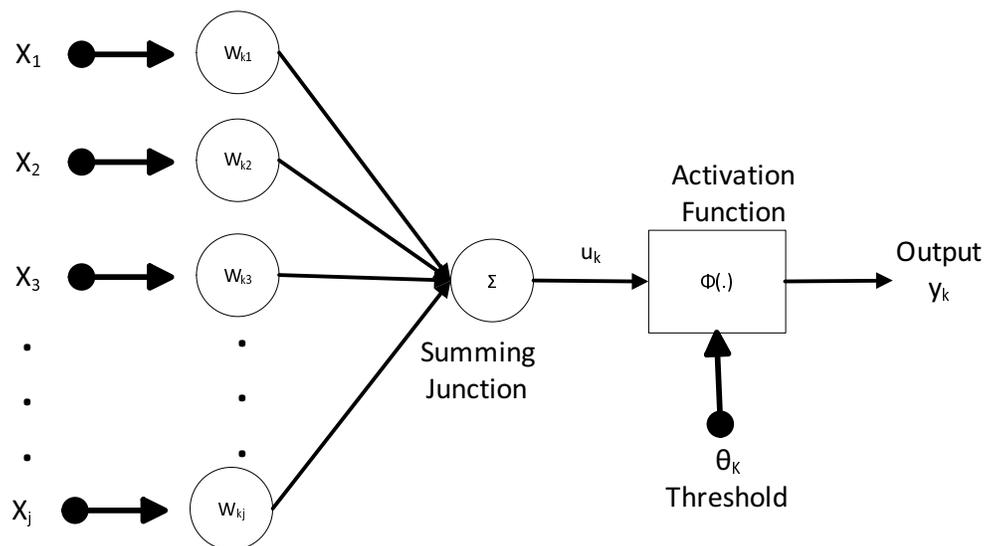


Figure 4.2: Artificial neuron

1. The unit **step** or threshold function Figure 4.3(a), which gives the output result (usually 1) only if the value calculated by the summing function is greater than a threshold value v is expressed by:

$$\phi(v) = \begin{cases} 1 & v \geq 0 \\ 0 & v < 0 \end{cases} \quad (4.3)$$

2. The function **sign** Figure 4.3(b) which gives the output a negative (or positive) information if the value calculated by the summing function is smaller (or larger) than a threshold value v . It is expressed by:

$$\phi(v) = \begin{cases} 1 & v \geq 0 \\ -1 & v < 0 \end{cases} \quad (4.4)$$

3. The **sigmoid** function which is expressed by the general formula:

$$\phi(v) = \frac{1}{1 + \exp(-av)} \quad (4.5)$$

where a is a factor regulating the speed of transition between two asymptotic values. The sigmoid function Figure 4.3(c) is important because it provides non-linearity of the neuron, which is essential for modelling of non-linear phenomena.

Artificial neurons enable the implementation of simple algebraic functions, such as an artificial neuron implementation of logic functions AND, OR, and NOT. For example, the implementation of the NOT function is used as the trigger unit step function with threshold $\theta = -0.5$. The input values can range from 0 (false) to 1 (true). If the input of the neuron is 0 then multiplied by the weight $w = -1$ gives $\Sigma = 0$. This value exceeds the threshold of -0.5 when the produced output is 1. If the input value is 1 then $\Sigma = -1$, the value is below the threshold of -0.5 , thus produce an output 0.

Finally in Figure 4.4 we can observe the correlation of the natural (organic) with artificial neuron seeing the identical structure and function.

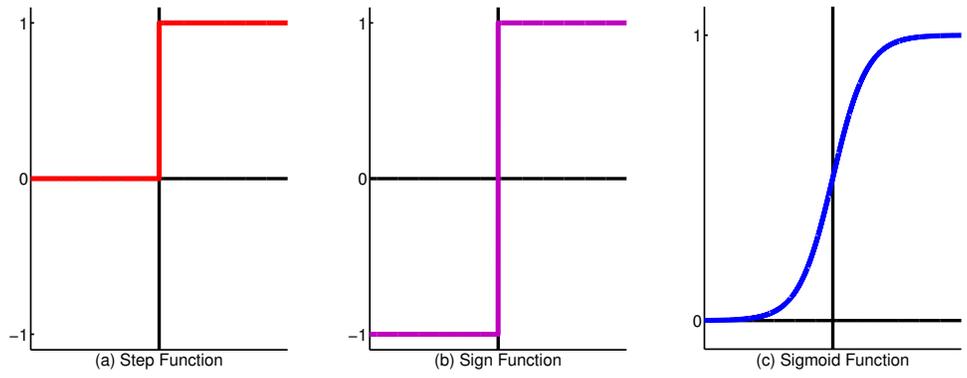


Figure 4.3: Activation function graphs

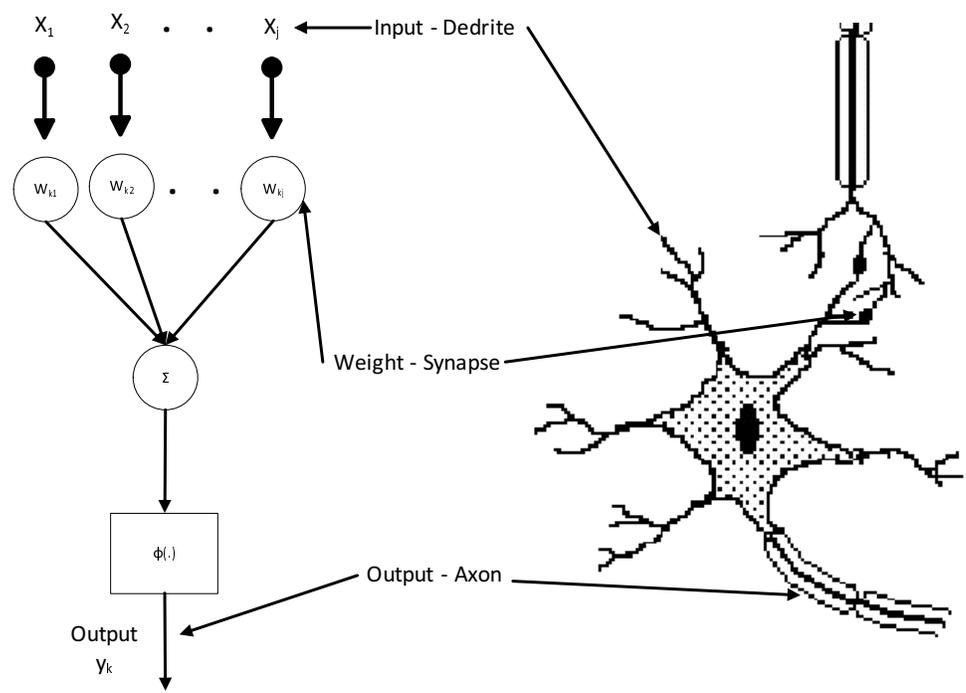


Figure 4.4: Equivalent natural-artificial neuron

4.1.5 Artificial neural networks

Neural Networks (NN) or Artificial Neural Networks (ANN) are an attempt to approach the operation of the human brain by a machine through mathematical functions. They have the ability to perform calculations on massively parallel manner. The architecture is based on the architecture of biological neural networks found in the human brain as consisting of a number of artificial neurons organised into structures similar to those of the human brain. In Figure 4.5 shows the structure of a ANN, discern the inputs, the input layer, the hidden layer, the output layer, outputs and the w_{kj} are the input weight.

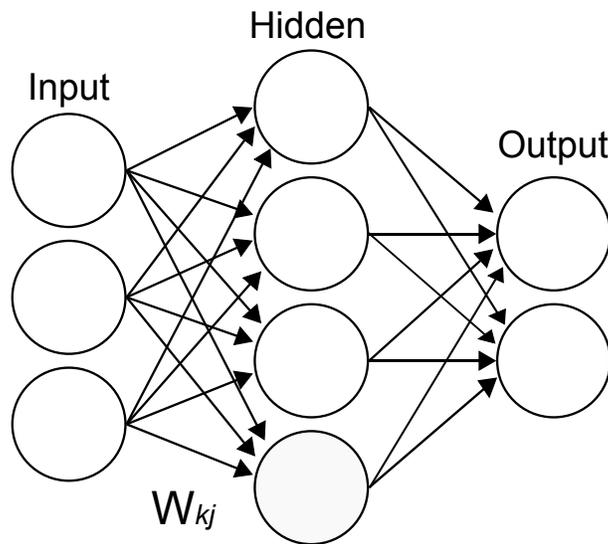


Figure 4.5: Structure of a Artificial Neural Network

The NN is a collection of neurons (Processing Units - PUs) linked together. A neuron is a unit of information processing which is fundamental for the functioning of a neural network. Each neuron has many inputs but only one output which in turn can provide input for other neurons. The connections between neurons differ in their importance which is weight factor. The processing of each neuron is determined by the transfer function that defines each output in relation to the inputs and rates of the weight. To use a NN, first must be trained to learn. Learning consists in determining the appropriate weight factors of the NN to perform the desired calculations, performed using

algorithms known as learning rules-algorithms. The role of weight coefficients can be interpreted as a store of knowledge which is provided through examples. In this way the NN learn from their environment, the physical model that provides the data.

4.1.6 Basic characteristics of neural networks

1. Size of neural network

The size of the NN is sometimes associated with the user experience and the nature of the problem. Beginners tend to use small networks and reduce the length of the application accordingly. Those who have fairly extensive experience leave the nature of the problem to decide on the size of the network. With the simulation of neural networks through programs that is available on personal computers today and the great progress of computer systems, a neural network with thousands of neurons and perhaps a hundred thousand links may not yet be a practical upper limit for non-static examples using forward or reverse propagation.

2. Neuron activation function

Typically the activation function is a continuous function that increases monotonically between a minimum and a maximum of $(-1 \text{ and } 1)$ as the weighted sum of increases in quantity. After one of the primary objectives of activation function is to keep the outputs of neurons within reasonable limits.

3. Number of layers

The back propagation networks usually have three layers, but more can be beneficial under some circumstances. It is sometimes better to use two smaller hidden layers in spite of one bigger layer. Some examples of neural networks which are often predetermined number of layers. The hidden layers act as layers of composition, extracting features from the inputs. Usually a larger number of hidden layers increases the processing power of the neural network but requires significantly more time on education and a greater number of training examples to train the network properly.

4. Number of neurons at each layer

The number of neurons in the input and output layers are defined by the nature of the problem.

4.1.7 Architecture of neural networks

Topological structure is the main feature of ANN and refers to architecture in which available and interconnect multiple neurons. Two basic properties define the architecture of a network: 1)the number layers and 2)the connections between neurons. A third feature, which is normally associated with the manner in which the neurons is structured, is the learning algorithm which is used to training the network.

4.1.8 Key features of neural networks

There are three characteristics inextricably linked with NN.

1. Their ability to learn by example : While NN is not the only systems capable of learning by examples, however distinguished for their ability to organise the information of the input data into useful forms. These forms are essentially a model that represents the true relationship between the input and output.
2. The possibility of considering them as distributed memory: The characterisation of ANN as distributed memory, stems from the fact that the coding information is distributed to all the weights of the network.
3. The ability for pattern recognition: The NN has excellent ability to identify patterns as they are affected by incomplete and or noisy data. Once a ANN trained to recognize conditions and situations require only one cycle of operation to identify a specific situation.

4.1.9 Benefits from using neural networks

These three key features of NN are, yet their advantages. But the use of NN give the following useful properties:

1. **Non-linearity.** This is because a NN is constructed by connecting neurons, which are non-linear devices. The non-linearity is very important property, especially if the physical mechanism to produce the input signals are non-linear (the case of most physical problems).
2. **Design of Input-Output** A common example of learning called supervised learning involves changing the NN weights, using a training set of samples or examples of projects. Each example consists of a unique signal input and desired response. The practice of repeated network for many examples, until the network reaches a steady state, now where there are changes in the weights. Thus the network learns from examples of constructing a design entry and exit to the problem at hand.
3. **Adaptability.** NN have the potential to adjust their weights to changes in their environment. Sometimes adjustments lead to reduced system performance, so we must be sufficient stability-plasticity dilemma.
4. **Evidentiary Response.** A NN is designed to provide information not only on the specific model chosen, but also for confidence in the decision is taken.
5. **Related Information.** Knowledge is represented by a very structured and active state of the NN.
6. **Resistance to errors.** The NN has great tolerance to structural errors. This means that the malfunction or destruction of a neuron or some links are not likely to seriously disrupt their operations and, as mentioned, the information they enclose are not localised to a specific point but pervasive throughout the network.

7. **Implementation with VLSI technology** . The compact parallel nature of the NN makes it possible to implement in VLSI technology, so that NN can be used in real applications.
8. **Uniformity of Analysis and Planning**. The concept is that the same notation is used in all fields that contain the application neural network which is indicated in several ways: These neurons represent a common ingredient in all neural networks. This property makes it possible to share theories and learning algorithms in different applications of neural networks.
9. **Analogy with Neurobiology**. The design of neural networks is an analogy of the brain. The neurobiologists view NN as investigated to explain neurobiological phenomena. Similarly, the engineers look to neurobiology for new ideas to solve complex problems.
10. Neural networks can provide each model time series $(t + 1)$.

4.1.10 Disadvantages of neural networks

Are mentioned below without a particular analysis of the disadvantages of the NN and their use:

1. The NN offers many degrees of freedom in modelling.
2. Training is essential.
3. The explanation and translation of weights in the NN is not possible due to non-linearity.
4. The influence of events can not be removed directly.

4.1.11 Learning and recall

4.1.11.1 Introduction

The ANN perform two basic functions: learning and recall. Learning is the process of changing the weights values of the network, so given a specific input vector to produced this output vector. This procedure is also called training of ANN. Their ability to learn from examples makes neural networks an extremely powerful programming tool when the main rules are not completely defined or when there is a inaccuracy rate and controversy in the data. Recall is procedure for calculating an output vector for a given vector input and weights.

The general way to modify the weights in an ANN is training, witch is categorised in three types of learning in ANN, the **supervised learning**, the **graduated learning** and **unsupervised learning**.

In supervised learning in the network are pairs of input vector - desired output. The ANN with the current state (weight), produces an output that differs from the original desired output. This difference is called error and based on that and a learning algorithm is usually adjusts the weights.

In graduated learning the output is classified as 'good' or 'bad' based on a numerical scale and the weights adjusted by based on this characterisation.

Finally, the response of the unsupervised learning is based on its ability to be re-configured by the input vectors. This internal reconfiguration is so that a specific set inputs to react strongly a particular neuron. Such sets entries correspond to concepts and features of the real world which the ANN is required to learn.

In practice, most applications of ANN use supervise learning, for which there are several algorithms. In algorithm based on the Delta rule learning, the difference between actual and desired output is minimised through a process least squares. The algorithm back-propagation, changes the weights based on the calculation of the contribution each weight in the total error. In competitive learning artificial neurons compete, somehow, between the and only one with the greatest response to given input modifies

the weights it. Finally, the random learning , changes in the weights introduced randomly and depending on whether the output is improved or not by some predefined criteria from the user, the changes are adopted or rejected.

4.1.11.2 Knowledge representation

The term knowledge refers to stored information or models used by a person or machine to interpret, predict and approximate react to the outside world[6].

The maximum work for a neural network to learn a model of the world (environment) in which they are installed and maintain the model satisfactorily with the real world to achieve its objectives an application of interest. The knowledge of the world consists of two types of information:

1. The known state of the world, represented by factors on what is known. This type of knowledge referred to as a prior information.
2. Observations of the world, acquired from sensors designed to explore the environment in which the neural network supposed to work. Usually these observations contain noise for sensors, and subject to errors . In any case, observations obtained in this way give the main part information, from which we get the examples used to train the neural network.

Each example consists of a pair of input/output: an input signal and the corresponding desired response for the NN. This is the reason why a set of examples represent knowledge about the environment of interest. The given set of examples of such a design a neural network can be made as follows:

- First a suitable architecture is selected for the NN, with an input layer consisting of as many neurons as the number of input parameter for the problem to be solved by neurons and an output layer consisting as many neurons as outputs parameters required for this problem. A subset of example data is used for the network's education through an appropriate algorithm. This phase is called learning.

- Second, the performance results given by the trainee network, the remaining dataset that have not been introduced before. The performance of the network then calculated up by comparing the output gave us the neural network compared with the desired output.

The second phase of the network is called generalisation, a term which is borrowed from psychology. In this lies a fundamental difference between designing a neural network and classical information processing. In the latter case usually proceed first forming a mathematical model environmental observations verifying the model with real data and then building the design from the base model. In contrast design a neural network based directly on real data with all data to make direct all the work. For that the NN not only provides a perfect model of the environment in which installed, but shows a job processing large of interest.

In a particular neural network architecture the representation of the environment knowledge is defined by the values of the network's free parameters (eg weights and activations thresholds). The way of knowledge representation is different from one neural network architecture to another thus it holds the key to the performance.

The issue of knowledge representation in an ANN is yet very complicated. The issue becomes even more complex when multiple sources of information act on the network and these sources interact. However, there are 4 knowledge representation rules that is common sense. These rules described below:

Rule 1 Similar inputs from similar groups should usually produce similar representations within the system, for this should be classified in the same category.

Rule 2 Items are categorised into different groups should have different representations on the network. The second rule is contrary to first.

Rule 3 If a particular feature is important, then there should be a large number of neurons involved in the representation of this object on the network.

Rule 4 Prior information and non volatilities should not be included in designing of the NN thus simplifying the system planning.

The last rule is particularly important because the inheritance of results provide the NN with a specialised structure. This is highly desirable for several reasons:

- Biological, optical and control networks are known to be very specific.
- A NN with specialised structure usually has fewer free parameters available for normalisation than a fully connected network. Therefore requires less data entry for training, learns quickly and often generalizes better.
- The rate of information transmission through a specialised network is accelerating.
- The cost of building a dedicated network is reduced because of its smaller size compared with a fully connected.

4.1.11.3 Learning with error correction

In this type of learning the required update (change, adaptation) of synaptic weights calculated by presenting the ANN input (vectors), comparing the resulting responses to the desired response and then changing the weights for direction to reduce the error .

In particular :

- $d_k(t)$: the desired output (target response) of neuron k in the discrete time time t .
- $x(t)$: the input vector applied to the input layer ANN.
- $y_k(t)$: the response of the k^{th} neuron

Apparently the pair $(x(t), d_k(t))$ is an example in neuron at time t . The error (difference) between the desired output $d_k(t)$ and the actual output $y_k(t)$ is: $e_k(t) = d_k(t) - y_k(t)$

under which we define the following criterion (cost function):

$$I = E \left[\sum_k e_k^2(t) \right] \quad (4.6)$$

called Mean Square Error criterion (MSE) and expresses the average value of the sum of squared errors. Here $E[\]$ is the statistical operator expected (mean) value and the ANN operates in a stationary environment with probabilistic unknown probability distributions. The summation in I extends to all k output, all output neurons. The problem of learning is now selected synaptic weights to minimize the mean square error. The exact solution of this problem requires knowledge of the static properties of stochastic processes inherent in each case. For this reason an approximate solution by minimizing the instantaneous squared error criterion:

$$J = \frac{1}{2} \sum_k e_k^2(t) \quad (4.7)$$

in the synaptic weights of the ANN w_{kj} where w_{kj} is the weight of the j^{th} synapse of the k^{th} neuron. The renewed (new) value $w_{kj}(t + 1)$ of the presumed synaptic weight is given by:

$$w_{kj}(t + 1) = w_{kj}(t) + \Delta w_{kj}(t) \quad (4.8)$$

To begin the learning rule (4.8) we need some preliminary knowledge for the weights values at time $t = 0$. If the ANN contains only linear data processing, so J (4.7) is exactly square, then the algorithm leads step-by-step in global minimum. But when the ANN contains non linearities, then the global minimum can no always be taken because algorithm can be trapped in a local minimum. Moreover, because the learning error correction behaves like closed recursion system the value of learning rate γ must be chosen very carefully to ensure process stability. This is because the learning rate has great influence on the performance of the method and affects not only the convergence speed of learning but also in its outcome. If γ has a small value the process goes smoothly,

but can take a long time the system to converge to a stable solution. Conversely, if the value of γ is large the course of learning is accelerated, but there is a risk that the process differs and the system becomes unstable.

4.2 Experimental procedure

4.2.1 Creation of data sets

The most important factors on the creation of successful ANN, in order to solve the specific problem, is the selection of the input variables as well as the creation of the data sets. In our study the input variables for the prediction of urban heat island intensity using NN are as follows:

- Date to represent the yearly climatic variations (the date is converted into the number of days starting from the 1st of January) and ranges within [1,365].
- Time, the time is converted into minutes of the day and ranges within [0,1380].
- Ambient temperature ($^{\circ}C$) is measured by the various experimental sites described in the previous section.
- Global solar radiation (W/m^2) measured by the National Observatory of Athens.

The raw ambient temperature data of the different experimental sites had difference starting and ending dates, so only the common dates were chosen. Furthermore the sampling rate of the experimental sites were ten minutes but global radiation data had a sample rate of one hour thus only one temperature measurement for six, were kept in order to have the same time period. The measurement that was selected to be included in the final data set had initial time stamps from 00 to 09 minutes for every hour of the day due to the difference synchronisation between the experimental sites. All remaining measurements were given a new time stamp of 00 minutes for every hour.

Input Data				Output Data
Day of year	Time (min)	Ambient Temperature ($^{\circ}C$)	Global solar radiation (W/m^2)	Ambient Temperature ($^{\circ}C$)
91	0	6.52	0	6.636
91	60	6.636	0	6.633
91	120	6.633	0	6.459
91	180	6.459	0	6.14
91	240	6.14	0	6.09
91	300	6.09	0	5.682
91	360	5.682	5	5.02
91	420	5.02	33	4.707
91	480	4.707	93	4.922
91	540	4.922	145	5.348
91	600	5.348	263	5.803
.
.
.
125	960	26.288	319	25.233
125	1020	25.233	294	22.457
125	1080	22.457	81	21.197
125	1140	21.197	0	19.905
125	1200	19.905	0	20.487
125	1260	20.487	0	20.357
.
.
.
194	1020	22.282	330	21.998
194	1080	21.998	126	21.65
194	1140	21.65	10	19.857
194	1200	19.857	0	18.703
194	1260	18.703	0	18.665
194	1320	18.665	0	18.273
194	1380	18.273	0	17.69

Table 4.1: Snapshot from the input dataset-one hour prediction horizon

Neural networks generally provide improved performance when the data are normalised. The use of original data as input to the neural network may cause a convergence problem. All the temperature and global solar radiation data sets are, therefore, normalised in the range $[-1, 1]$ by dividing the difference between the actual value x and minimum values by the difference between the maximum x_{max} and minimum values x_{min} , i.e.

$$x_{nor} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4.9)$$

The main goal of normalisation, in combination with the weight initialisation, is to allow the squashed activity function to work at least at the beginning of the learning phase. Thus, the gradient, which is a function of the derivative of the non-linearity, will always be different from zero. At the end of each algorithm, the outputs are transformed into its original data format.

For every experimental site three different data sets were created. The only difference in the data sets were the output or result data. Each one of three data set represent one of the three different prediction horizon used in our study. The three prediction horizons are: one, twelve and twenty-four hours ahead. To create the data set a special function was constructed in order to shift the output data to the correct place. Initially the structure of the data were $date_x, time_x, ambientTemperature_x, globalRadiation_x$ as input and $ambientTemperature_x$ as output data, where $subscript_x$ represent the line number in the data set. If the prediction horizon was set to be twelve hours, all the output data will be moved twelve places up, so for the input $date_1, time_1, ambientTemperature_1, globalRadiation_1$ the output would be $ambientTemperature_{13}$. The first 12 measurement of the output data will be discarded and the last day of data set.

4.2.2 Architectures of ANN used in our case study

The selection of the networks architecture is based on various results presented in the literature as well as in a preliminary trial and error procedure for various neural net-

works types. The non-linear autoregressive network with exogenous inputs (NARX), Hopfield, Radial basis function network and Learning Vector Quantization (LVQ) neural networks were also tested without providing any encouraging results for the specific problem.

4.2.2.1 Feed-Forward

Feed-Forward NN (Figure 4.6) is an artificial neural network where input layer consists of the inputs of the neural and connected to an output layer composed of neurons (computational nodes). In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network. The feed-forward NN was the first and arguably simplest type of artificial neural network devised.

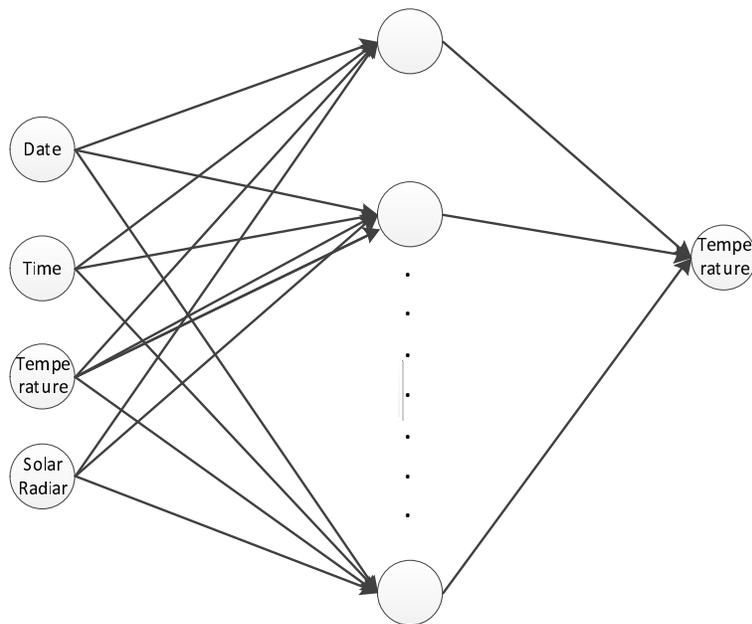


Figure 4.6: Schematics diagram for the Feed-Forward neural network.

4.2.2.2 Elman

Elman NN (Figure 4.7) is a simplified version of recurrent NN. Recurrent NN are an architecture of NN where connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behaviour. Unlike feed-forward NN, recurrent NN can use their internal memory to process arbitrary sequences of inputs. This makes them applicable to tasks such as unsegmented connected handwriting recognition, where they have achieved the best known results. Elman is three-layer network, with the addition of a set of "context units" in the input layer. There are connections from the middle (hidden) layer to these context units fixed with a weight of one. At each time step, the input is propagated in a standard feed-forward fashion, and then a learning rule is applied. The fixed back connections result in the context units always maintaining a copy of the previous values of the hidden units (since they propagate over the connections before the learning rule is applied). Thus the network can maintain a sort of state, allowing it to perform such tasks as sequence-prediction that are beyond the power of a standard multilayer perceptron.

4.2.2.3 Cascade

Cascade is an architecture (Figure 4.8) and supervised learning algorithm for artificial neural networks developed by Scott Fahlman at Carnegie Mellon in 1991[5]. Instead of just adjusting the weights in a network of fixed topology, Cascade begins with a minimal network, then automatically trains and adds new hidden units one by one, creating a multi-layer structure. Once a new hidden unit has been added to the network, its input-side weights are frozen. This unit then becomes a permanent feature-detector in the network, available for producing outputs or for creating other, more complex feature detectors. The Cascade-Correlation architecture has several advantages over existing algorithms: it learns very quickly, the network determines its own size and topology, it retains the structures it has built even if the training set changes, and it requires no back propagation of error signals through the connections of the network.

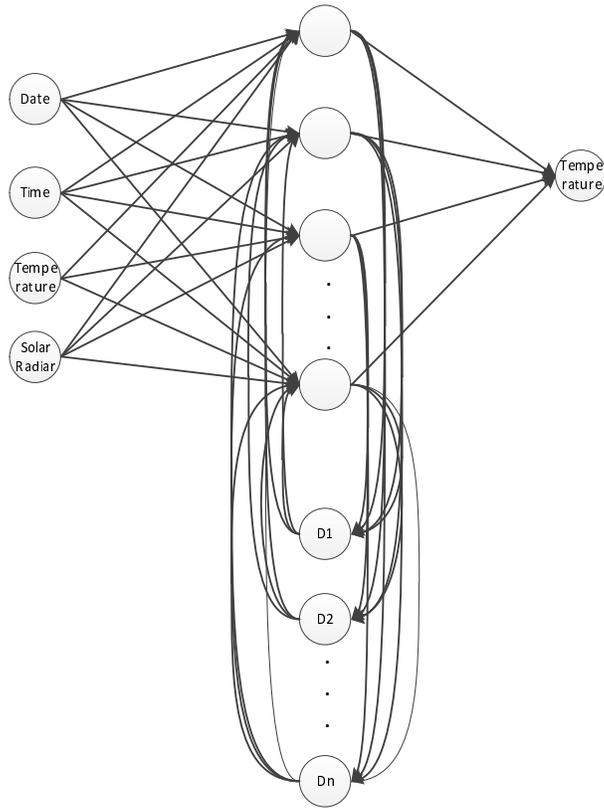


Figure 4.7: Equivalent natural-artificial neuron

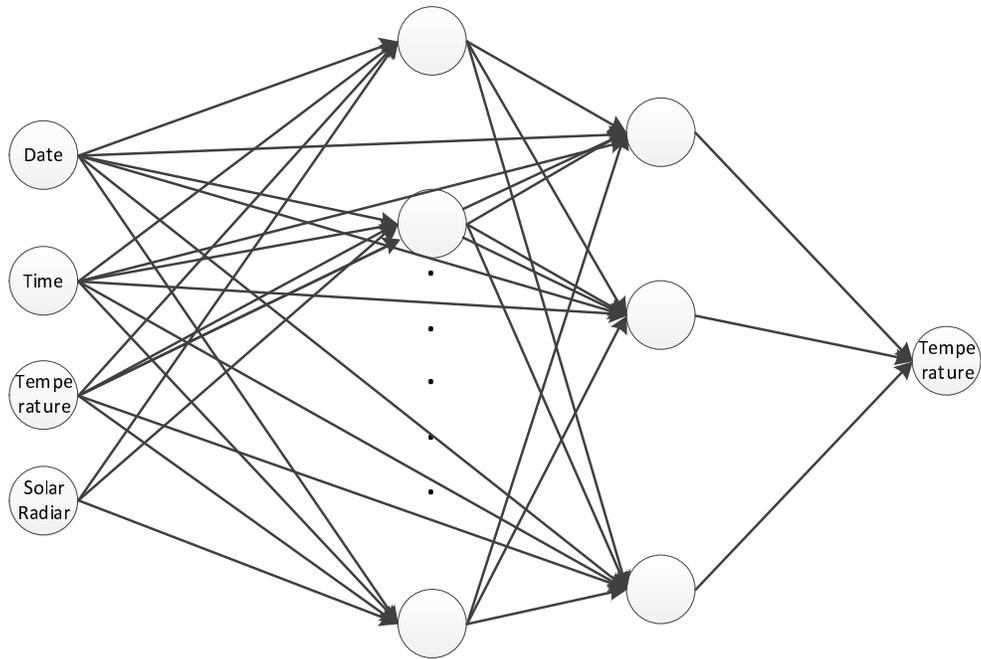


Figure 4.8: Equivalent natural-artificial neuron

4.2.3 Training algorithms of neural networks used in our case study

4.2.3.1 Levenberg-Marquardt

The Levenberg-Marquardt algorithm (`trainlm`) was designed to approach second-order training speed without having to compute the Hessian matrix¹. When the performance function has the form of a sum of squares (as is typical in training feed-forward networks), then the Hessian matrix can be approximated as

$$H = J^T J \quad (4.10)$$

and the gradient can be computed as

$$g = J^T e \quad (4.11)$$

where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The Jacobian matrix can be computed through a standard back-propagation technique [8] that is much less complex than computing the Hessian matrix. The iterative calculation of weights using Levenberg-Marquardt is

$$w_{k+1} = w_k - (J^T J + \mu I)^{-1} J^T e \quad (4.12)$$

where μ a rate that increases or decreases depending on whether the recurrence is successful or not, ie if the cost function is increased or decreased respectively after the iteration. I is the diagonal identity matrix.

4.2.3.2 Scaled conjugate gradient

The scaled conjugate gradient algorithm (`trainscg`) uses a numerical approximation for the second derivatives (Hessian matrix)[18], but it avoids instability by combining the

¹Hessian matrix is the square matrix of second-order partial derivatives of a function

model-trust region approach from the Levenberg-Marquardt algorithm with the conjugate gradient approach. This allows scaled conjugate gradient to compute the optimal step size in the search direction without having to perform the computationally expensive line search used by the traditional conjugate gradient algorithm. Of course, there is a cost involved in estimating the second derivatives.

4.2.3.3 Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton

Newton’s method (trainbfg) is an alternative to the conjugate gradient methods for fast optimisation. The basic step of Newton’s method is:

$$w_{k+1} = w_k - A_k^{-1} g_k \quad (4.13)$$

where A_k^{-1} is the Hessian matrix of the performance index at the current values of the weights and biases. Newton’s method often converges faster than conjugate gradient methods. Unfortunately, it is complex and expensive to compute the Hessian matrix for feed-forward neural networks. There is a class of algorithms that is based on Newton’s method, but which does not require calculation of second derivatives. These are called quasi-Newton (or secant) methods. They update an approximate Hessian matrix at each iteration of the algorithm.

4.2.3.4 Gradient descent

With the gradient descent (traingd) learning algorithm the weight and biases are updated in the direction of the negative gradient of the performance function. Gradient descent requires the definition of an error function to measure the neuron’s error in approximating the target. The sum of squared errors:

$$\mathcal{E} = \sum_{p=1}^{P_r} (t_p - o_p)^2 \quad (4.14)$$

is usually used, where t_p and o_p are respectively the target and actual output for the p -th pattern, and P_T is the total number of input-target pairs in the training set. The aim of gradient descent is to find the weight values that minimize \mathcal{E} . This is achieved by calculating the gradient of \mathcal{E} in weight space, and to move the weight vector along the negative gradient.

4.2.3.5 Gradient descent with momentum and adaptive learning rate

Gradient descent with momentum and adaptive learning rate (`traingdx`) combines adaptive learning rate with momentum training. `Traingd` training function updates the weight and bias values according to gradient descent momentum and an adaptive learning rate.

4.2.3.6 Resilient back propagation

Resilient back propagation (`trainrp`), is a learning heuristic for supervised learning in feed-forward artificial neural networks[22]. This is a first-order optimisation algorithm. `Trainrp` takes into account only the sign of the partial derivative over all inputs, and acts independently on each weight. For each weight, if there was a sign change of the partial derivative of the total error function compared to the last iteration, the update value for that weight is multiplied by a factor η^- , where $\eta^- < 1$. If the last iteration produced the same sign, the update value is multiplied by a factor of η^+ , where $\eta^+ > 1$. The update values are calculated for each weight in the above manner, and finally each weight is changed by its own update value, in the opposite direction of that weight's partial derivative, so as to minimise the total error function. η^+ is empirically set to 1.2 and η^- to 0.5.

4.2.3.7 Comparison between different neural network architecture and training algorithms

After preliminary test three different neural network architectures and six training algorithms were selected (mention above). Each neural network consists of one to three hidden layers with 20–40 neurons each, followed by an output layer of one neuron. The tangent sigmoid function was selected as a definitive result from the preliminary tests as transfer function 4.5. The Koridalos site was randomly selected to evaluate the performance of the 6 different training algorithms and 3 neural network architecture. A series of test were performed in order to find the optimal training function for each one of the three neural network architectures. The 1 and 24 hours datasets were used in order to carry out those tests.

The results of the tests are shown in Table 4.2.3.7. The mean value, standard deviation and MSE² of the absolute difference between the measured and predicted ambient temperature for each hour of Koridalos site. The best performance is achieved by:

- The Cascade neural network using the BFGS quasi-Newton (4.2.3.3) as training function.
- The Elman neural network with the Levenberg-Marquardt (4.2.3.1) as training function.
- Feed-Forward neural network with scaled conjugate gradient (4.2.3.2) as training function.

Another sires of test was conducted using another location, Peristeri, to verify the previous results. The results are shown in Table 4.3 and confirm the results of the previous experiment.

In the next step the results from the three neural networks, are compared in order to examine the prediction accuracy. The optimum results for each neural network were

²Mean Square Value

Training function	1h			24h		
	Mean value	Standard deviation	MSE ²	Mean value	Standard deviation	MSE ²
Feed-Forward						
trainlm	2.220	1.601	0.621	2.513	2.274	0.955
trainscg	2.041	1.385	0.484	2.291	1.970	0.820
trainbfg	22.355	12.376	38.023	30.355	17.376	45.023
traingd	7.471	7.215	6.859	8.873	6.997	8.285
traingdm	15.643	10.772	20.319	18.323	12.033	26.518
traingnrp	10.66	7.677	11.252	7.533	6.074	6.409
Cascade						
trainlm	1.965	1.483	0.423	2.153	2.681	1.07
trainscg	1.954	1.342	0.393	2.163	1.978	0.773
trainbfg	1.666	1.084	0.593	0.942	0.845	0.131
traingd	3.606	3.259	1.812	3.476	3.14	1.668
traingdm	3.042	3.032	1.23	3.593	3.799	2.303
traingnrp	2.088	1.547	0.535	2.325	2.487	1.052
Elman						
trainlm	1.137	1.192	0.342	2.53	2.357	1.026
trainscg	1.897	1.295	0.367	1.517	1.085	0.303
trainbfg	1.975	1.321	0.614	1.254	0.967	0.652
traingd	2.56	1.723	0.78	1.556	1.294	0.314
traingdm	2.443	1.445	0.574	1.383	1.068	0.235
traingnrp	3.215	1.687	0.691	1.469	1.325	0.325

Table 4.2: Performance comparison of difference training functions for Koridalos site.

Training function	1h			24h		
	Mean value	Standard deviation	MSE ²	Mean value	Standard deviation	MSE ²
Feed-Forward						
trainlm	3.606	3.259	1.812	4.128	5.655	4.348
trainscg	3.174	2.457	1.407	3.087	3.087	2.253
trainbfg	7.707	5.638	9.454	14.607	10.348	23.87
traingd	9.201	7.23	12.596	7.677	5.402	6.144
traingdm	6.98	6.779	8.493	10.595	8.019	13.464
traingnpr	6.89	6.391	7.157	11.877	10.265	14.669
Cascade						
trainlm	2.93	2.269	1.166	3.691	3.908	2.745
trainscg	2.805	2.155	1.043	2.764	2.682	1.454
trainbfg	2.89	2.281	1.157	2.912	2.636	1.411
traingd	6.317	7.253	6.964	5.172	5.109	3.599
traingdm	6.831	5.684	5.88	4.689	4.776	3.54
traingnpr	2.994	2.282	1.233	3.381	4.217	2.475
Elman						
trainlm	2.234	1.351	0.911	1.142	0.736	0.151
trainscg	3.299	2.041	1.16	1.268	0.923	0.173
trainbfg	3.754	2.545	2.84	3.412	1.785	0.985
traingd	2.886	1.92	0.52	1.835	1.575	0.571
traingdm	3.315	5.818	3.122	3.202	7.685	3.652
traingnpr	3.125	2.958	2.958	3.587	2.218	2.504

Table 4.3: Performance comparison of difference training function for Peristeri site.

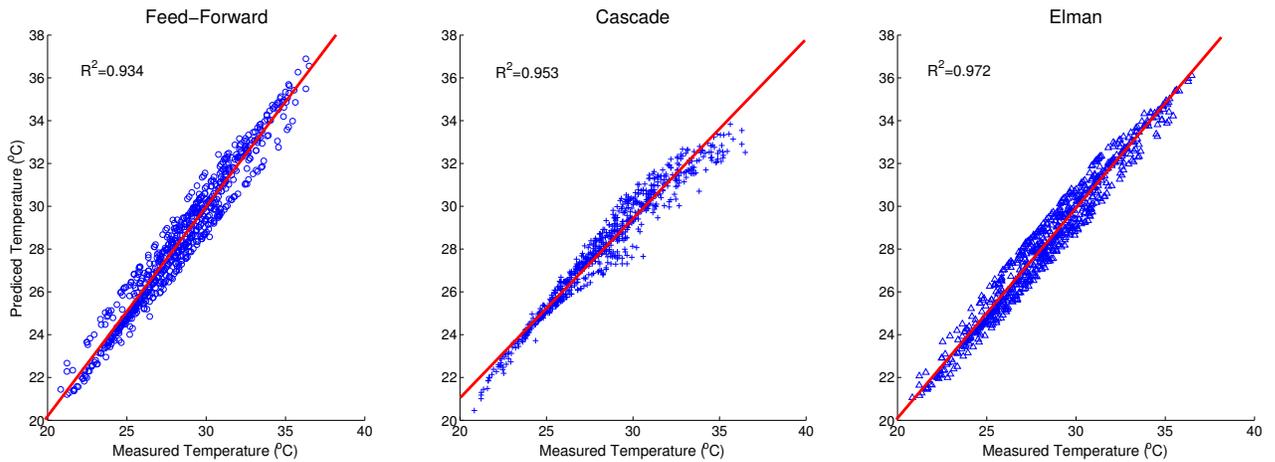


Figure 4.9: Comparison between the three different ANN for 1-h prediction horizon

chosen. Figure 4.9 and 4.10 present the measured versus the predicted values for Koridalos site for 1-h and 24-h prediction horizon respectively. The best fit of the measured to the observed data is achieved by the Elman followed by the Cascade and the Feed-Forward architecture. Another method was used for verifying the previous result. The mean value and standard deviation of the percentage error (4.15) were utilised to calculate the difference between the measured and the predicted temperature values. Also the mean square value were calculated (Table 4.2.3.7).

$$PrecentError = \frac{|Experiment - TheoreticalValue|}{Theoreticalvalue} \times 100\% \quad (4.15)$$

Training function	1h			24h		
	Mean value	Standard deviation	MSE ³	Mean value	Standard deviation	MSE ³
Feed-Forward	2.8	2.2	1.12	2.0	1.9	0.93
Cascade	2.4	1.5	0.65	1.3	1.2	0.79
Elman	1.8	1.0	0.35	0.8	0.5	0.32

Table 4.4: Performance comparison of three ANN architectures for Koridalos site.

The results from both test clearly shows that the optimal ANN for predicting the urban heat island intensity is Elman with the Levenberg-Marquardt (4.2.3.1) as training function. The specific network architecture is used for all sites.

4.2.4 Results

Training and verification of the ANN is performed using the data collected during the period from 06/04/2009 to 07/09/2009 for each experimental site. Therefore the training and verification period is shortened to five months. The data are fed into the ANN as blocks of 24 values corresponding to each hour of the day. The neural network has a training period of 40-60 days. The remaining data are used to verify the quality of network and adaptation of the neural network to the new data. Figure 4.12 and ?? show the measured and predicted temperatures of 5 different locations (i.e. Ilion, Zefyri, Petoupoli, Aegaleo and Agii Anargyri) and for two different dates 19-20/06/2009 and 05-06/07/2009 respectively. As we can see in Figure 4.12, the diurnal fluctuation of temperatures is very smooth and is followed by the 1 hour and 24 hours Elman prediction algorithms quite accurately. Furthermore, although the daily temperature fluctuations depicted in Figure ?? are not as smooth and predictable, the Elman NN manages to follow the measured data most of the time.

Another significant aspects of the proposed neural network architecture that should be examined are the alterations in prediction accuracy due to seasonal variations and the necessities to retrain the network when the season changes. In Figures 4.14 and 4.14 the measured and predicted temperatures are depicted for Haidari and Agia Barbara

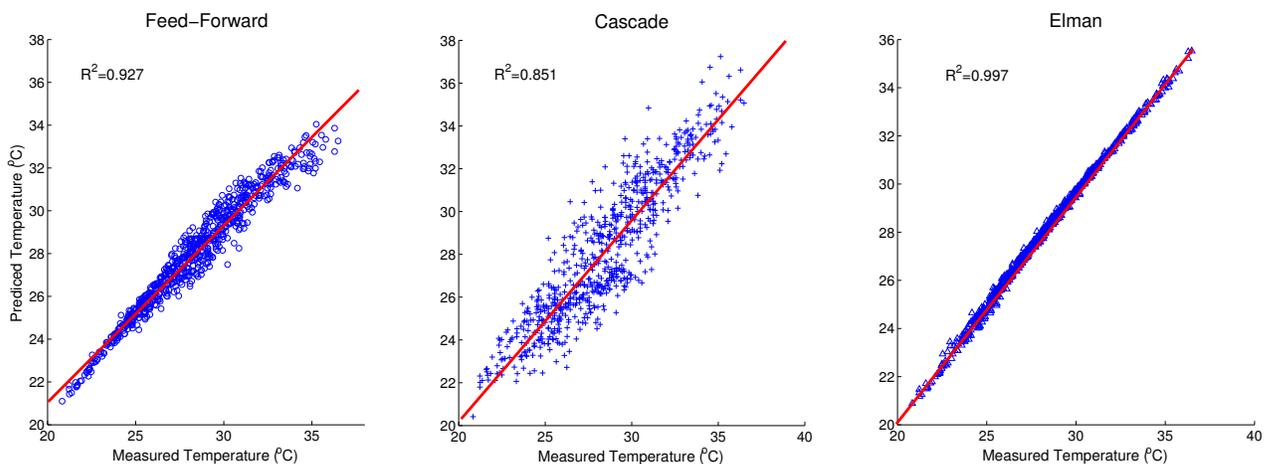


Figure 4.10: Comparison between the three different ANN for 24-h prediction horizon

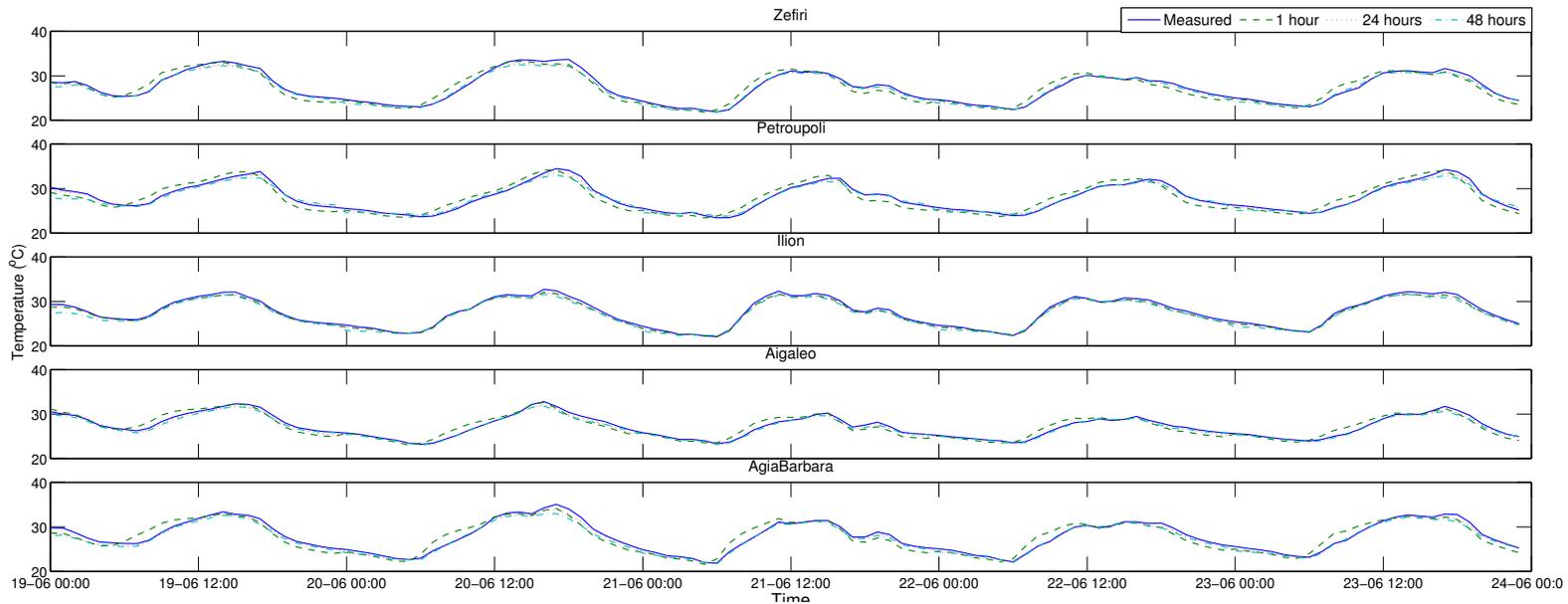


Figure 4.11: Measured-predicted temperatures for 19-24/06/2009

experimental sites. Although the prediction accuracy is not always the highest possible, especially for the 48 hours prediction horizon, the overall prediction accuracy does not change significantly with the seasonal changes. The seasonal changes of the standard deviation between the measured and predicted values for Haidari and Agia Barbara sites are tabulated in Table 4.5. Another significant aspect is the methodology's response to different weather conditions or during weather changes. For this reason the ANN's response is studied for two experimental sites i.e. Koridalos and Haidari as well as for day to day temperature changes (Figure 4.15). As we can see in the specific figure, although outdoor temperature is considerably decreased from 19/6/2009 to 20/6/2009, and also increased from 23/8/2009 to 24/8/2009, the ANN follows this change in a successful manner especially for the 24 hours prediction horizon.

Moreover isothermal images are developed to imprint the UHI intensity over Athens. The mapping of the region is performed using Google Earth while the isothermal lines are added by Surfer 8 software. For each day that the UHI over Athens is analysed, a set of four images is constructed to visualise the ANN prediction:

- The first image maps the isotherms over Athens using the measured data of the

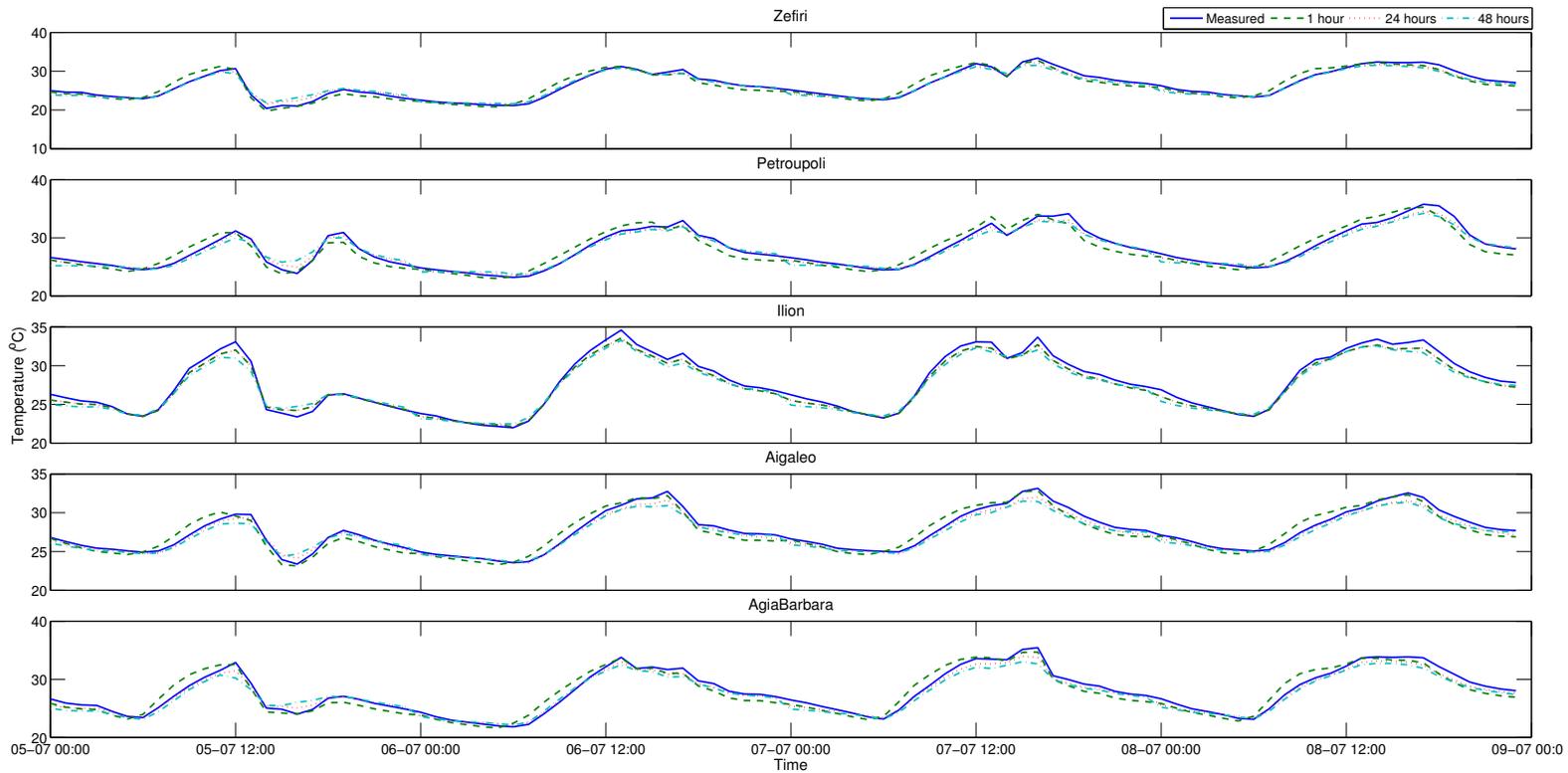


Figure 4.12: Measured-predicted temperatures for 05-09/06/2009

specific day and time.

- The second image represents the isotherms of Athens urban heat island based on the 1 hour prediction results for the specific day and time.
- The third image maps the isotherms of Athens urban heat island based on the 24 hour prediction results for the specific day and time.
- The fourth image plots the isotherms of Athens urban heat island based on the 48 hour prediction results for the specific day and time.

Indicatively the isothermal maps of the UHI intensity over Athens for two days (i.e. 1/7/2009 and 18/6/2009) are illustrated in Figures 4.16- 4.17. The prediction of the maximum temperatures for the 1/7/2009 has a maximum error of $1.6^{\circ}C$ and $1.9^{\circ}C$ for the 24 hour and 48-hour prediction horizon respectively. Moreover the visualisation of UHI intensity prediction shows that the isotherms of the 24 hour prediction are very

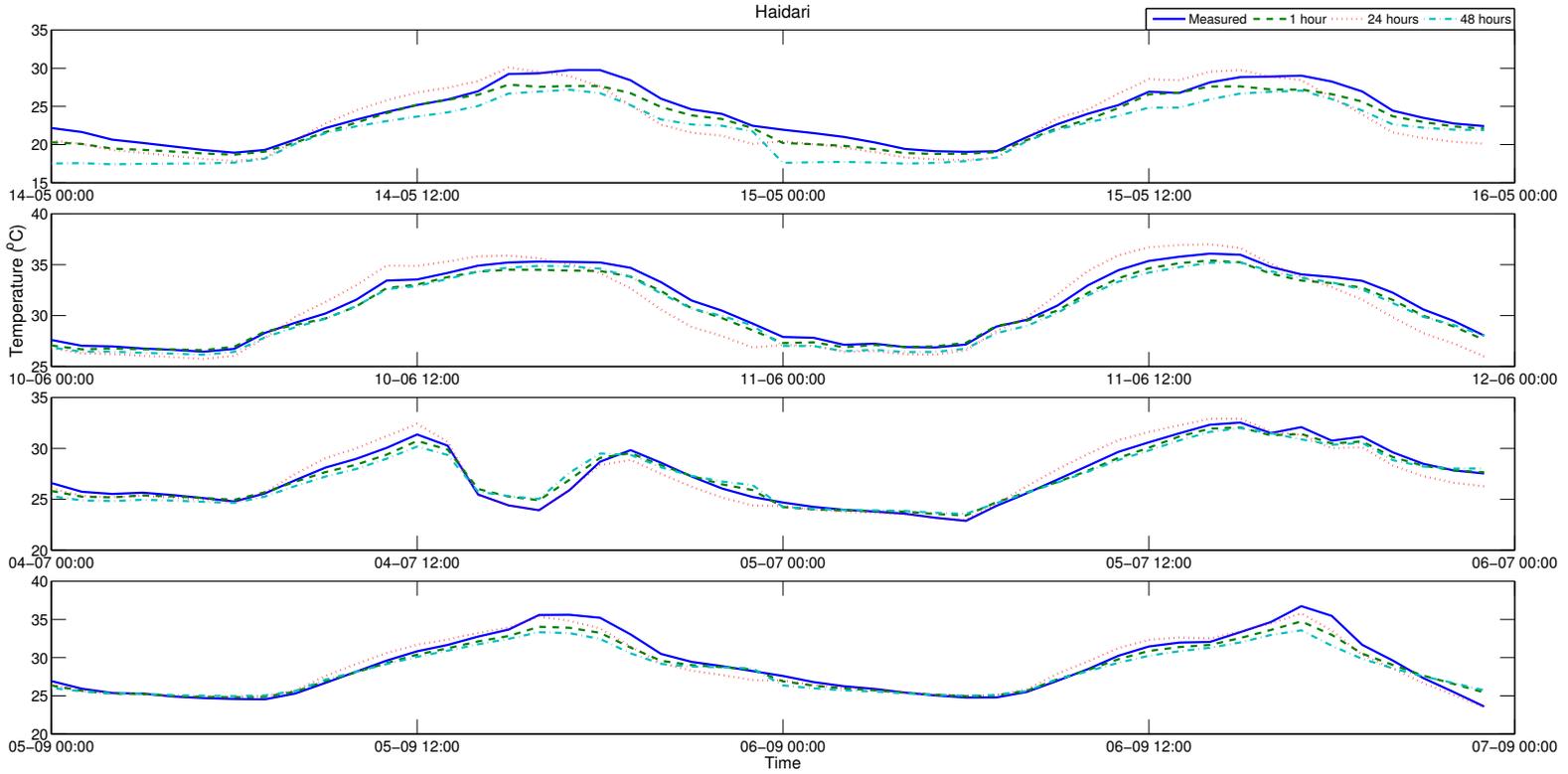


Figure 4.13: Monthly comparison for measured-predicted temperature for Haidari experimental station

Prediction horizon	Agia Barbara			Haidari		
	1 h	24 h	48 h	1 h	24 h	48 h
12/05/2009 - 31/06/2009	0.5516	0.9301	0.7581	0.5532	1.4003	1.0013
01/06/2009 - 30/06/2009	0.4392	0.8087	0.7330	1.0013	1.0267	0.5104
01/07/2009 - 31/07/2009	0.7787	0.7930	1.2449	0.5767	0.8057	0.9998
01/08/2009 - 31/08/2009	0.4127	0.5587	0.9892	0.5326	0.5731	1.0996
01/09/2009 - 06/09/2009	0.4682	0.5061	0.9300	0.7029	0.7274	1.1727

Table 4.5: Seasonal variations of standard deviation between measured and predicted temperatures for Agia Barbara and Haidari sites.

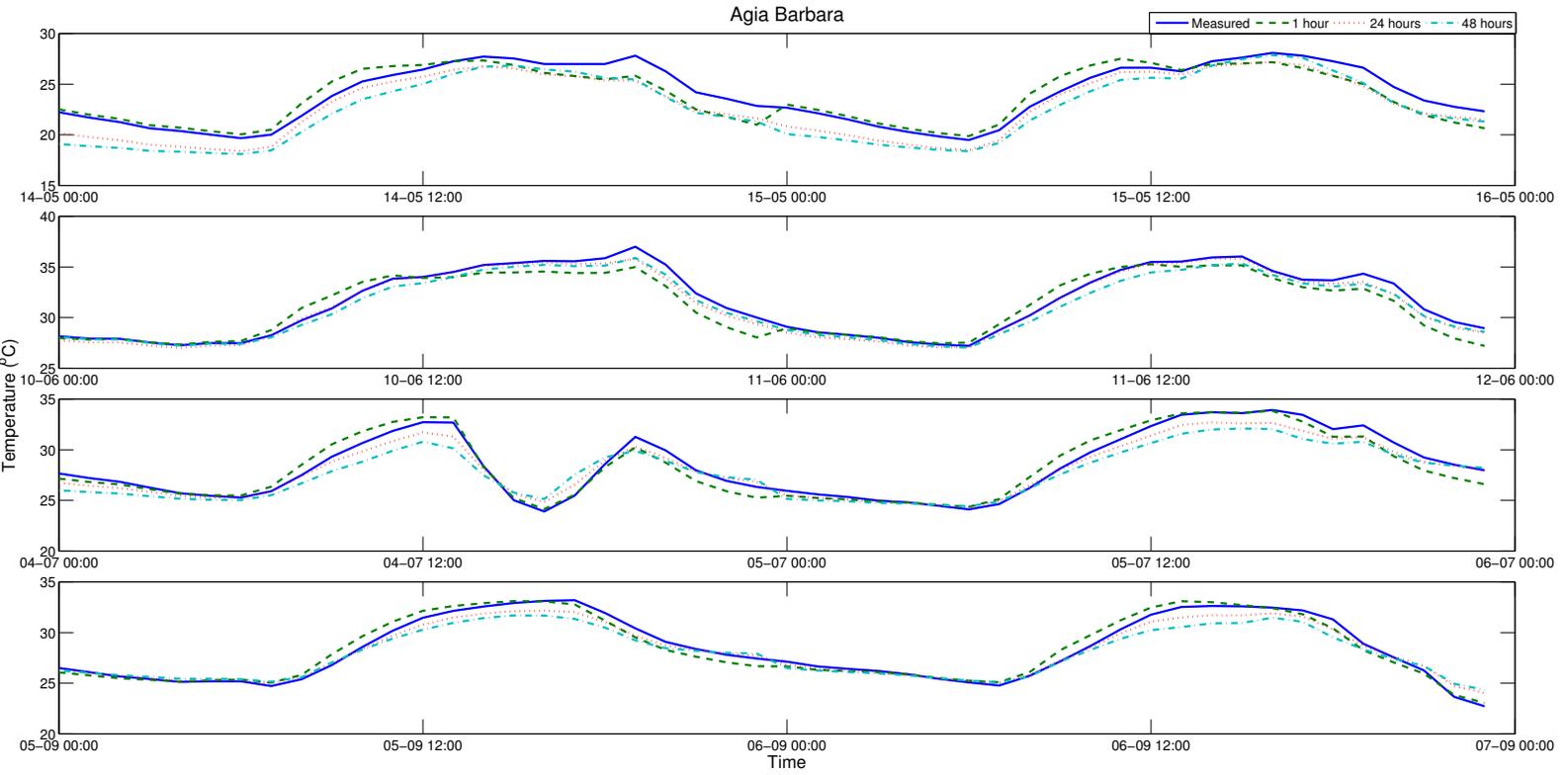


Figure 4.14: Monthly comparison for measured-predicted temperature for Agia Barbara experimental station

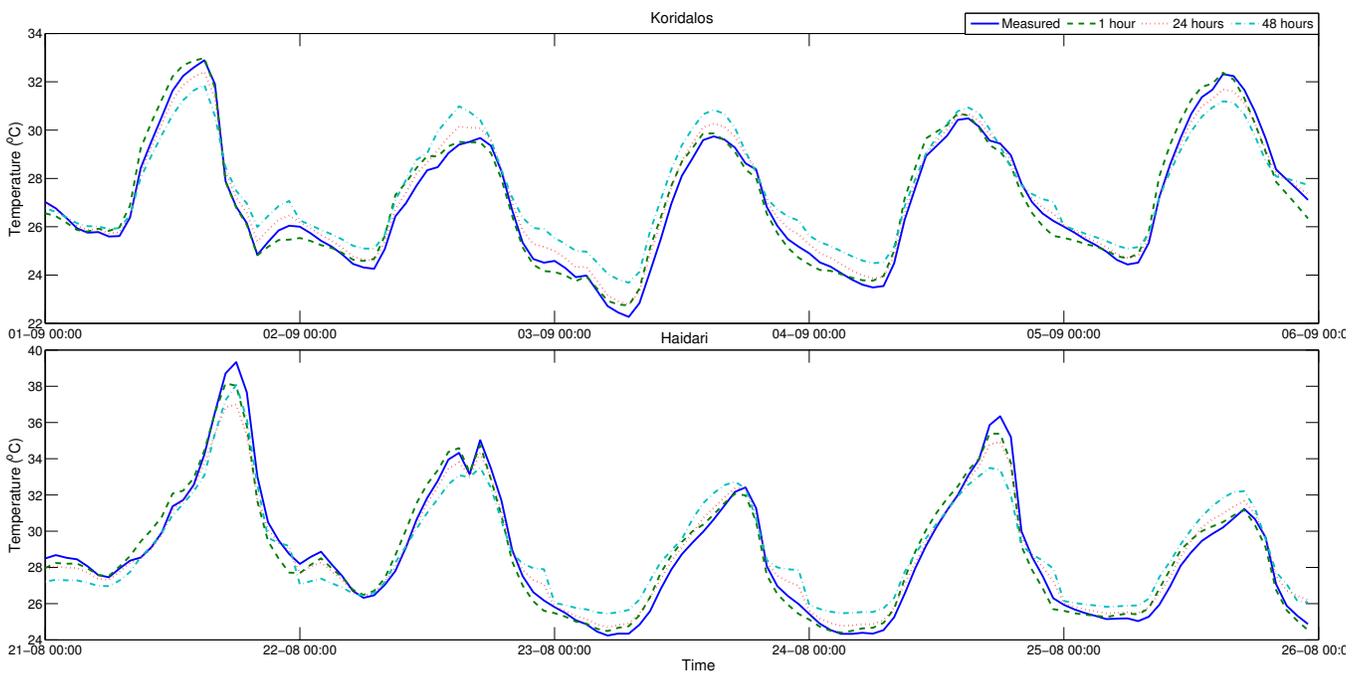


Figure 4.15: Response of the ANN to weather changes

close to the actual measured ones while the 48 hours prediction has a slightly different picture. Therefore the specific NN architecture and methodology followed is quite accurate for the 24 hours prediction horizon.

The urban heat island intensity is then calculated versus the reference station, i.e. National Observatory of Athens. The predicted versus the measured urban heat island intensity for three sites (i.e. Agia Barbara, Egaleo and Halandri) and for 24-hours prediction are depicted in Figures 4.18, 4.19, 4.20 respectively. The figures show a satisfactory fitting with a RMSE less than 0.3 and R^2 to be close or higher than 0.9 for all three sites which represents a good prediction of the urban heat island intensity.

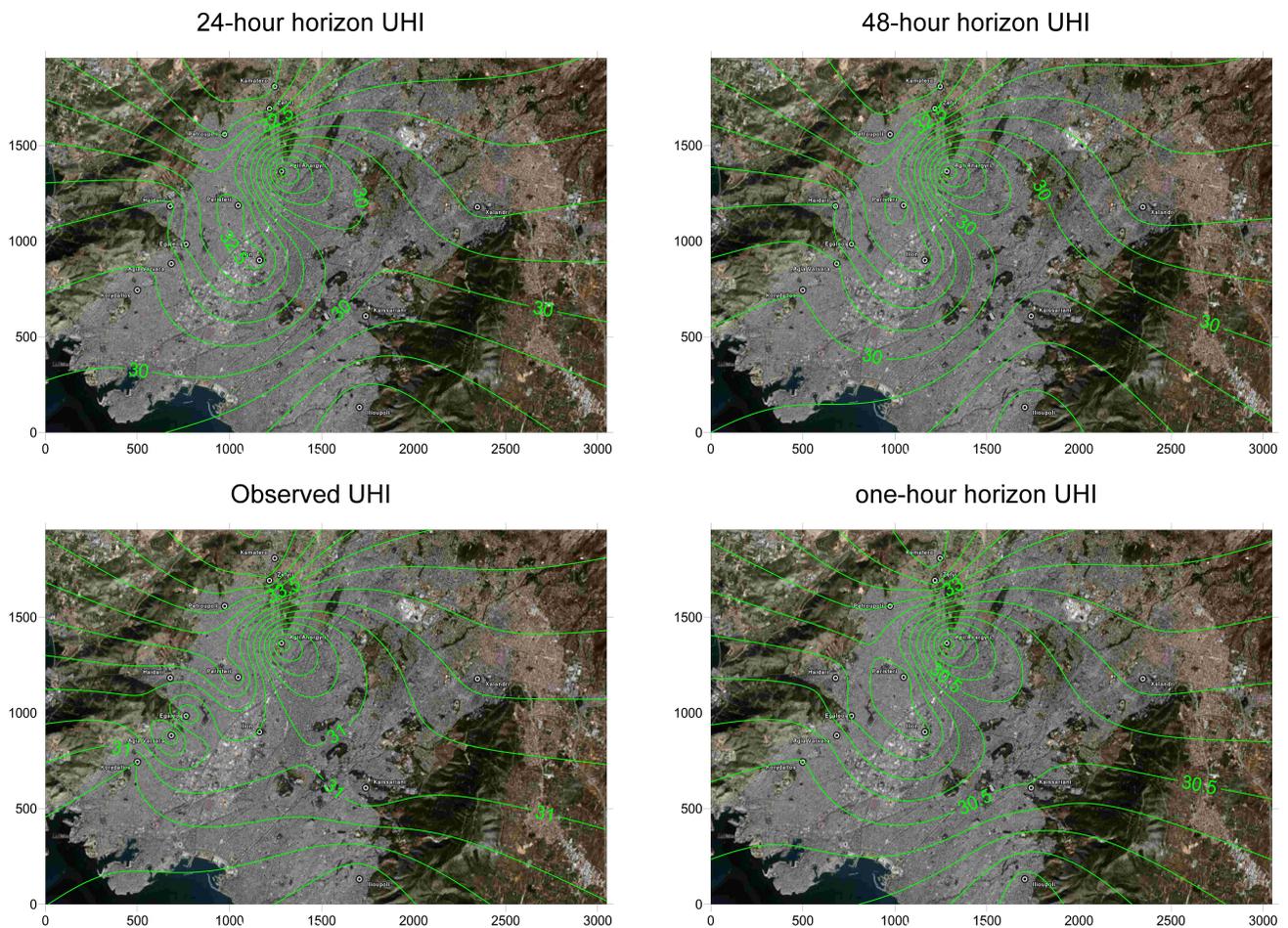
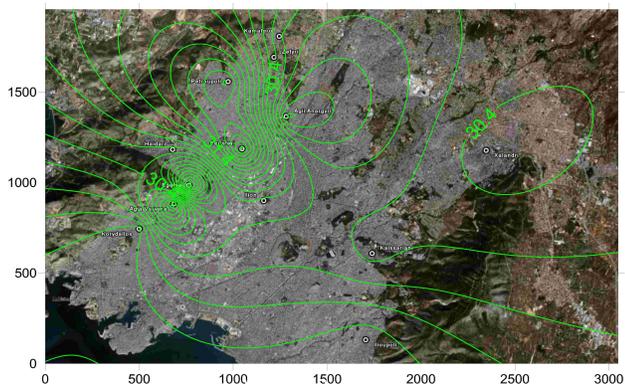
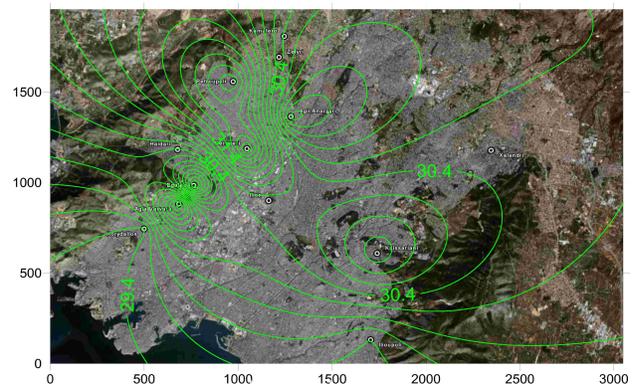


Figure 4.16: The UHI in Athens during 18/06/2009

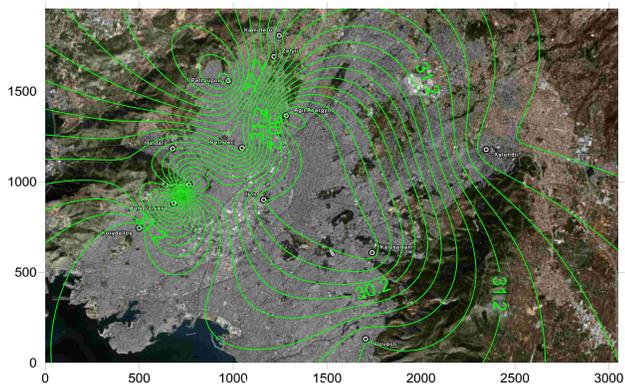
24-hour horizon UHI



48-hour horizon UHI



Observed UHI



one-hour horizon UHI

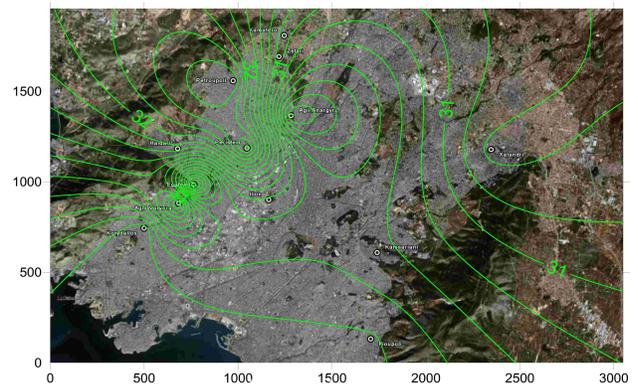


Figure 4.17: The UHI in Athens during 01/07/2009

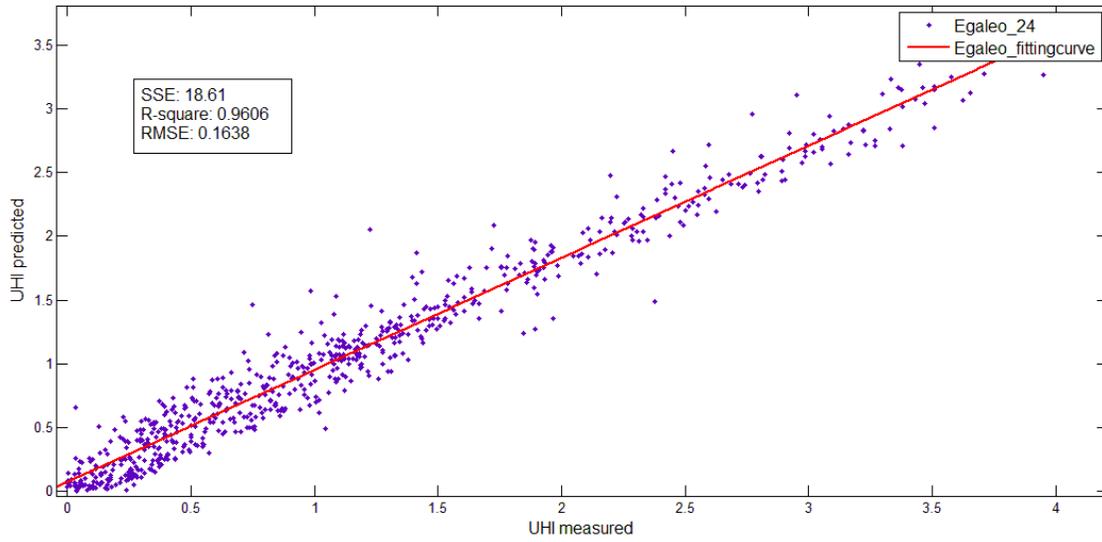


Figure 4.18: The measured UHI versus predicted UHI 24h for the Agia Barbara site

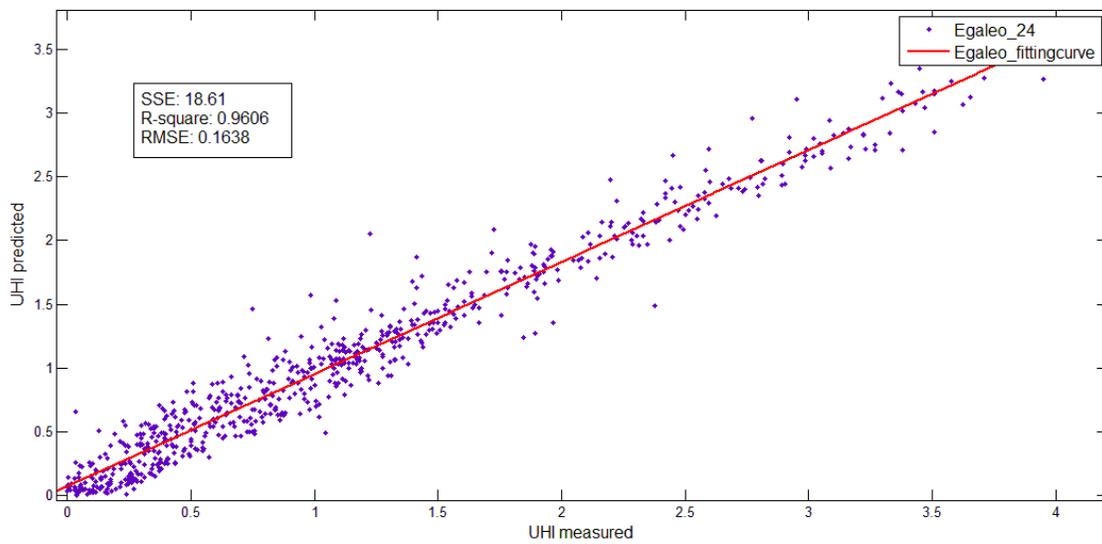


Figure 4.19: The measured UHI versus predicted UHI 24h for the Egaleo site

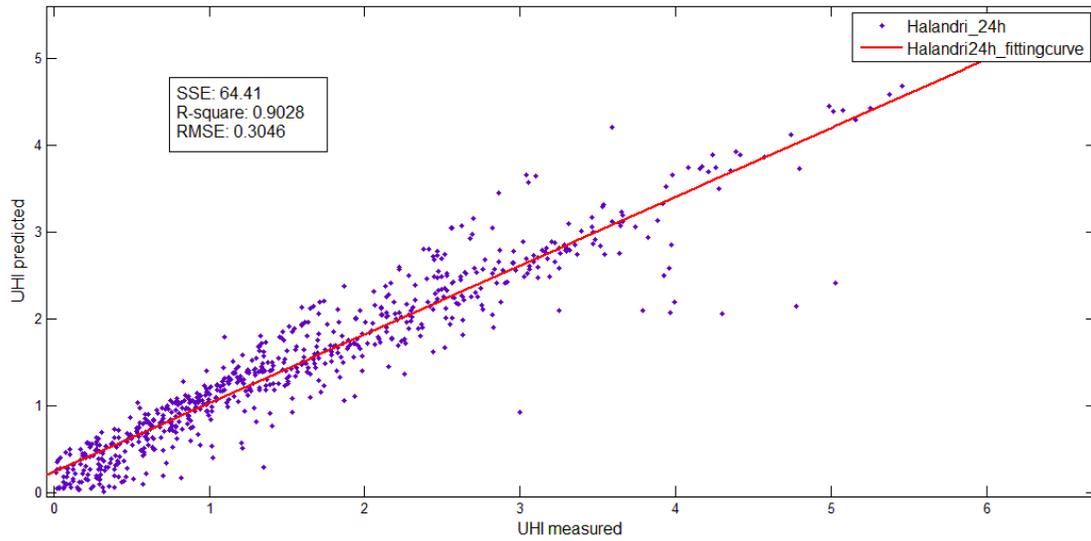


Figure 4.20: The measured UHI versus predicted UHI 24h for the Halandri site

Chapter 5

Interpolation using Geographic Information System

5.1 Introduction on Geographic Information System

Geographic Information System (GIS) is an organised collection of hardware and software computer systems, spatial data and human resources (Figure 5.1) with the purpose of collecting, recording, updating, managing, analysing and informing in any form with respect to the geographical environment.

The GIS offers quick and easy access to large volumes of geographic data. The key to success is that they provide the ability to edit, analyse data in a region and manipulate individual characteristics. The functions of a GIS can be used where there is a need for spatial data management or even where there is a need for analysis of the spatial dimension of data.

There are essentially two types of GIS data: vector and raster. These differ in how the spatial data is displayed and stored. Morris and Simmonds[20], Gordon and Kapetsky[7] presented in a summary manner useful comparisons between them. In both systems, a geographic coordinate system is necessary for viewing the site. Many geographic coordinate systems are defined, ranging from simple Cartesian coordinate system XY,



Figure 5.1: The components of a GIS

Source: (<http://www.gsdi-africa.com>)

grid spatial presentation that corresponds to the real world such as pairs of longitude and latitude and reaching the World Geodetic System (WGS84) or Universal Transverse Mercator (UTM).

In the vector representation of spatial data are represented as points, lines and polygons. A point defined as a simple set of coordinates. Examples of items can be considered as a police station or locations of meteorological stations etc. Features such as rivers, canals and roads are easily shown as lines. Lines have start and end, points are referred to as nodes but they can include a large number of vertices ie sides. The polygons are shown as areas which are within a set of lines. A polygon consists of a number of lines but is characterised by the fact that the initial and final node is the same. For this study examples of polygons can be reported as a separate type of land use, city limits or the coastline of a country.

In the raster representation during a spatial analysis, either events or otherwise continuous data are represented. The raster data are generally divided into two categories the layers and images. Values in a layer raster represent certain measured quantities or categorise a phenomenon like altitude, pollution, temperature, etc. For example with a value of 10 in a cell, one can represent forest in a land cover map. Corresponding values in a cell with an image may represent the reflected light or energy of a satellite image or a scanned image.

As a simple raster represents only one subject like land use, soils, rivers, roads, etc., many rasters should be used to fully illustrate an area. A raster consists of a number of cells. Each cell is a square that corresponds to a particular part of a region. All the cells of a raster should have the same size. The size of cell can be any size but should normally be sufficiently small to serve the requested analysis. So a cell may represent a square kilometre, one square meter or one square centimetre.

The cell size of a raster of a region that is selected depends on the analysis of data required. The cell should be small enough to capture the required detail, but large enough so in order to achieve productivity for the computer storage. The more homogeneous variables exist in a specific area in terms of topography and land cover, the

greater may be the size of the cell without affecting the accuracy of the analysis. The GIS allows the storage and analysis of different raster examination in the same database.

5.1.1 Introduction

During the last decades many countries found that the needs for reliable and updated information about the earth, society and the environment could not be met by the traditional ways of collecting, recording, reporting and information processing. Thus, especially since the early 80s, enormous growth has been performed in the Geographic Information Systems (GIS). There are many definitions of what a GIS is. The GIS represents "a powerful set of tools for collecting, storing, taking, at any time transformation and display spatial data in real world".

The GIS has three main components which are in constant balance and interdependency. These three parts are machines (hardware), algorithms (software) and available resources. It should be noted that the overwhelming progress done for the first two components of GIS, makes any discussion of specific computer systems and software meaningless. The mechanical parts of a GIS is computers, networks and various peripheral devices such as designers, printers, scanners, digitizers, etc.

The last 25 years, problems associated with geographic information management were solved in a global and national level with the help of GIS. Before the GIS development we used to study the world using models such as maps and the globe sphere. But with the rapid development of hardware and software industry these models could be developed and introduced to PC. The computational models are combined with analysis tools to construct a GIS. In a GIS it is not only possible to study a specific map, but every possible map. With the right data, one can see very easily and quickly in front of any computer any information from anywhere in the world ranging from political boundaries, cities and population density of the earth to land use, energy consumption and impressions deposits a small Greek island.

In a digital map these processes are simpler, because information consists of levels,

or a collection of geographic features among themselves. So adding or subtracting one or more levels of information, very quickly creates various versions of the same map. Each geographical object or entity (river, city, etc.) is called feature. For example, sea level does not consist of a collection of geographic entities, but it is a single, continuous area that changes from one area to another as needed, with water depth. The geographical area is called the surface. Similar surfaces can represent soil temperature, altitude, and generally phenomena that vary continuously in space (continuous fields).

Geographic objects can be represented by an infinite number of shapes. In reality, however, an object is represented by one of the three geometric shapes: Polygon, Line, Point. Polygons are represented as all entities that have boundaries such as provinces of Greece, lakes etc. Geological entities such as rivers, roads, utility networks etc are represented as lines. As points are represented entities that are too small to be represented as polygons such as cities, elevation points, sampling locations, fire hydrant locations, hot spots (WiFi) places etc. In general all the features are represented as lines, points or polygons and are called vector data.

In contrast to the features, surfaces (raster data) have a numeric value rather than shape. Phenomena that vary continuously in space such as temperature, rainfall, slope, altitude do not have any shape such as roads and rivers. Such geographical phenomena are more easily represented as surfaces than as attributes (features). The surface is represented as a square table of equal size cells, with each cell to represent a unit area, for example, 1 square meter. It contains a numeric value with the phenomenon it represents, for example, with a temperature of $20^{\circ}C$, or a depth or altitude of 200 meters.

A feature is not only the shape and geographical information but also a host of other features which are not related to the geometry of that feature. For example, the descriptive characteristics of a layer containing polygons of the prefectures may be the population, name, the unemployment rate, etc. All of the descriptive characteristics of the layer are stored in a table. This table has one record (row) for each attribute and each record consists of fields (columns) in which descriptive characteristics data are stored.

The scale of a map is expressed as a ratio and the relationship between the size of the map features and size that have the characteristics of the real world. A scale 1 : 1000000 means that the characteristics of the map you see on screen is 1000000 times smaller than in reality. In a digital map you can enlarge or reduce any feature.

5.1.2 Representation discrete objects - Vector models

The GIS represent the real world, in various ways. Each entity can be recorded and presented in different ways. The purpose of each application and scale visualisation of our data leads us to the need to decide how to represent each entity using the GIS. There are three ways to display entities. The points, lines and polygons.

5.1.2.1 Point entities

The points represent small entities whose their depiction of polygons is impossible. Each section contains information relating to the coordinates of many times the altitude at which it is located. An important factor, of course, is the scale at which we work. For example, a house can be visualised as a point if the scale is small. In case, however, the scale of the object is large, could be presented as polygons.

5.1.2.2 Linear entities

The lines help us represent entities that are in fact linear, such as roads, rivers and rail-ways. Also used to present virtual entities such as the borders of states or the aircraft runways. Linear entities are essentially a sequence of points which are joined together. In this case we again recorded the coordinates and altitude of all points that make up the entity. We can thus see the entity in three dimensions.

5.1.2.3 Polygonal entities

The polygon illustrate surfaces which may be lakes, states, forests, etc. Technically, these consist of a set of rows. Usually, the larger the scale imaging, the more entities may be

represented by polygons. The figure below we see how we can represent the real world with points, lines and polygons. In this example, the point entities depict fire hydrants, streets linear and polygonal parcels of a neighbourhood.

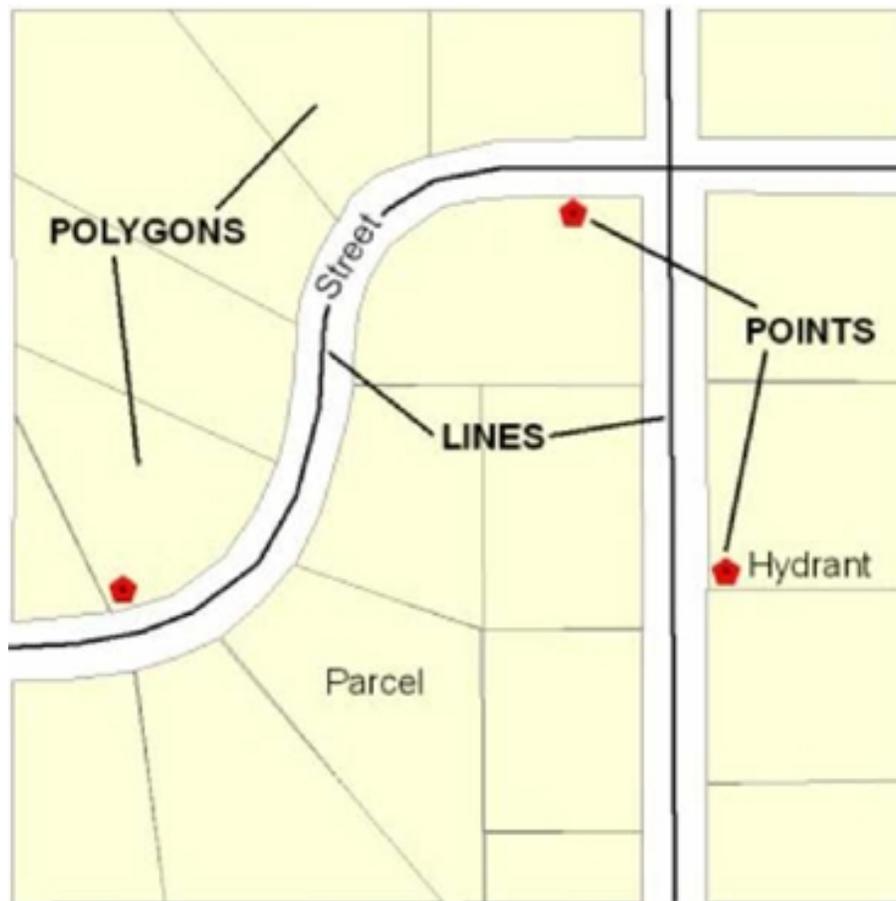


Figure 5.2: Polygonal, linear and point entities in a neighbourhood

Source: www.co.dakota.mn.us

5.1.2.4 Representation of continuous space - raster model

The simplest form of visualisation data in raster model, consisting of serial (scalar) placement of the cell (Figure 5.3). Specifically, each cell is determined by the pair row-column of the corresponding table and a number value identifying the type or attribute value that it represent. This approach provides a standard but arbitrary polygons system for storing geographic data. Therefore, this technique is by nature associated with

a coordinate system, but does not necessarily require perfect correlation. Basically the system uses a raster table i, j , to express/represent the spatial variations in a PC.

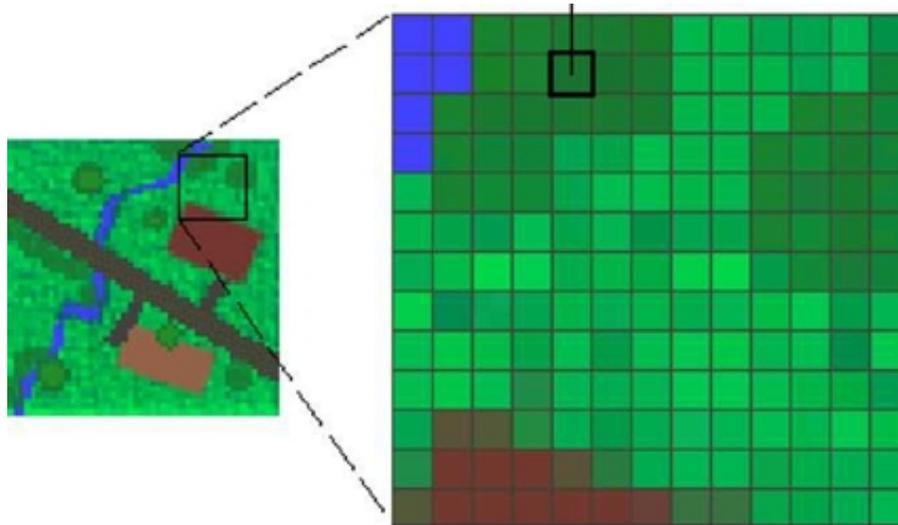


Figure 5.3: Raster Data

Source: webhelp.esri.com

5.1.2.5 Triangular model (Triangulated Irregular Network)

A triangular model (TIN) is an efficient and accurate model representation of a continuous surface, especially when it comes to heights. A TIN model created with special software with the following way:

- Originally collect coordinates points x, y, z , collection lines (break lines) which have abrupt change of the surface, collecting exclusion areas such as lakes, etc. and gathering areas which must be the TIN as shoreline islands, etc.
- The software taking into account the above data sets and other parameters of accuracy and tolerances, creating an optimum network of triangles, which is called a triangulation of Delaunay⁽¹⁾. In a TIN, each triangle is created to be equilateral

¹In mathematics and computational geometry, a Delaunay triangulation for a set P of points in a plane is a triangulation $DT(P)$ such that no point in P is inside the circumcircle of any triangle in $DT(P)$. Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation; they tend to avoid skinny triangles.

as possible.

- Each triangle has a façade with a gradient change in slope. From a TIN an altitude of a point X, Y coordinates can be found by locating the corresponding triangle first and then intermediate interpolation the amount within the triangle. In a TIN a flat surface requires a small number of points to be represented and a mountainous region requires a large number of points especially if you have sudden changes in terrain.

A TIN consists of points, lines and polygons. Points (mass points) are elevation points measured by topographical instruments, or data conversion and have coordinates X, Y, Z. The cracks show areas where there is a large discontinuity in surface soil. Examples are the streams, ridges and edges of buildings, etc. The exception areas are polygons that do not want to create triangles, such as lakes in particular, rivers, roads, etc. Finally, the model can be confined within the limits of any area desired. These limits are usually defined by the shoreline of an island or limits an area you want to calculate volumes.

5.1.2.6 Coordinates system

The representation of a point entity in a map is made using a pair X, Y coordinates, in relation with a reference system. A straight line obviously needs two pairs of coordinates, at the beginning and at the end of the line. If the line changes direction, then we need a pair of coordinates at each change. Finally a polygon can be considered as a line with the same beginning and end. We use two kinds of coordinates, or a combination for the determination of each point: linear (distance from a start of measurement) and angular (a measure of the angle between any content management - starting measuring the angular size - and direction to the point of interest. The linear coordinates in the orthonormal system of axes we say often Cartesian. Geodetic coordinates² are called the two angles that define the position of a point on the surface of the ellipsoid of reference. The geodetic latitude ϕ is the angle between the perpendicular to the surface

²Refers to a location on earth defined by its latitude, longitude and elevation.

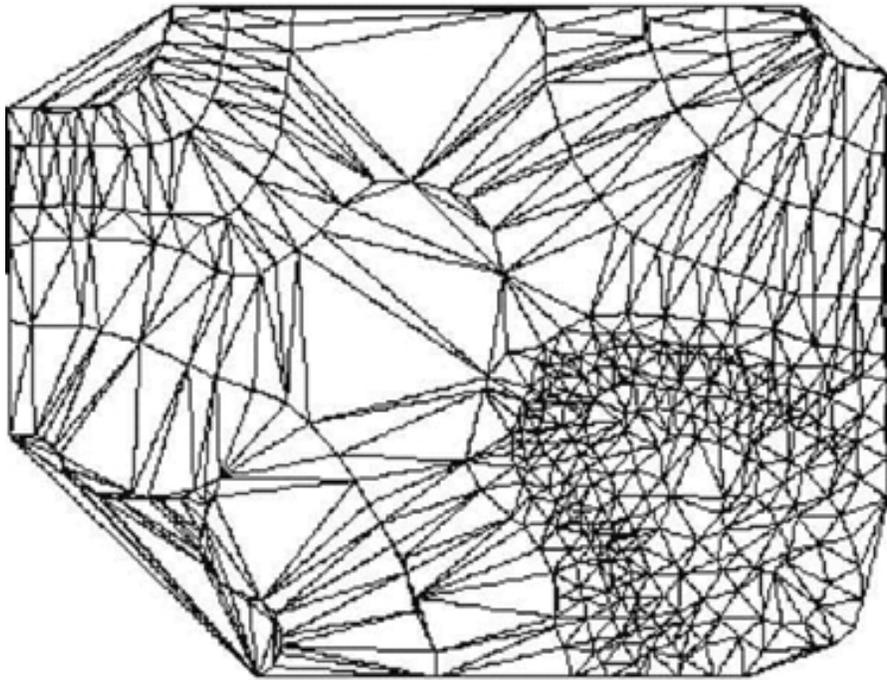


Figure 5.4: Triangulated Irregular Network

Source: www.geog.buffalo.edu

of the ellipsoid and the equatorial plane of the ellipsoid. The geodetic latitude λ is the angle between the plane of the prime meridian and the meridian that passes through the point of interest.

The main projection systems used in Greece are:

1. The system HATT with datum ³, the Old Greek Datum, with beginning to the pedestal of the Athens Observatory and the elliptic Bessel. All Greece is divided into about 130 spheroids tables to reduce distortion. The centres of trapezes changing every 30' with integer coordinates in degrees and 15' or 45'. This system was used by various political services and the Hellenic Military Geographical Service (GYS) in scales 1:5000, 1:2000, etc.
2. The Greek Geodetic Reference System 1987 (GGRS 87), is the new Greek Datum implemented in 1987, using the ellipsoid GRS80, beginning with the central pedestal of Dionysus. Greece is divided into one zone, central meridian $\lambda_0 = 24^0$ and scale factor 0.9996. The Central Meridian at Y-axis and the equator as the axis of X having no negative coordinates, the central meridian considered to have a value of $X_0 = 500000$ meters. The projection is Transverse Mercator
3. The UTM system with the European Datum of 1950 (also known as ED50)) has as its starting point the Postdam in Germany and uses the ellipsoid of Hayford. Greece is divided into two zones of the sixth with central meridians $\lambda_0 = 26^0$ and $\lambda_0 = 27^0$ with scale factor 0.9996. These zones are number 34 and 35 respectively. The system is used by the GYS maps 1:50000 and 1:250000.

Finally another coordinate system used in Greece is the World Geodetic System of 1984 (WGS84). The U.S. Department of Defense planned, financed and maintained the program, launch satellites and the operation of ground stations for monitoring the global positioning system (GPS). The measurement of distance between the control stations and the satellites allowed the calculation of the location of these stations in a global

³geodetic reference system

three-dimensional Cartesian geocentric system. The spatial frame of reference created by the way, was one of the elements to create the first single global system, WGS72. The first fundamental overhaul of the system was in 1984. The new WGS84 system includes the equivalent of an older original point, which is now the geocentric coordinates of points in the spatial frame of reference.

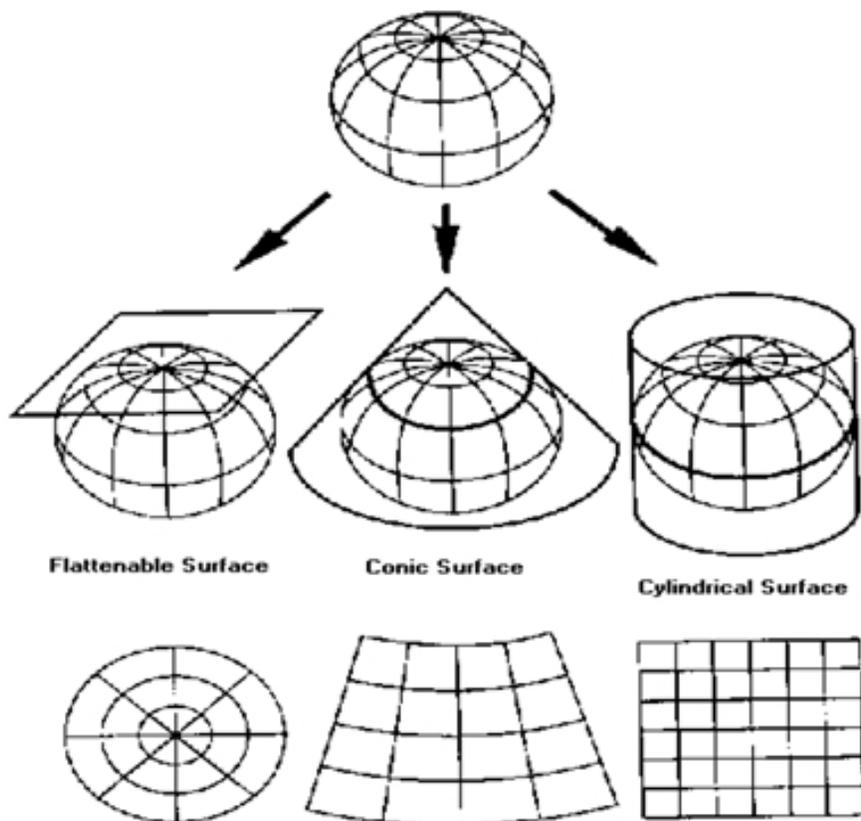


Figure 5.5: The azimuthal, the conical and cylindrical projection

5.1.3 Interpolation using Geographic Information System

The geostatistical analyst interpolation techniques are used to create a continuous surface either from the measured sample points stored in a point feature layer or by using the polygon centroids and then predict the values at unmeasured locations from the surface created. Many studies have been done using the geostatistical analyst and some

of the fields that benefit by virtue of these interpolation techniques include agricultural production, temperature data, soil contamination, mining, health care and meteorology. Some of these studies that used the geostatistical analyst interpolation techniques are summarised below.

5.1.3.1 Methods of interpolation

5.1.3.1.1 Inverse Distance Weighted One of the most commonly used techniques for interpolation of scatter points is inverse distance weighted (IDW) interpolation. Inverse distance weighted methods are based on the assumption that the interpolating surface should be influenced most by the nearby points and less by the more distant points. The interpolating surface is a weighted average of the scatter points and the weight assigned to each scatter point diminishes as the distance from the interpolation point to the scatter point increases. To predict a value for any unmeasured location, IDW will use the measured values surrounding the prediction location. Those measured values closer to the prediction location will have more influence on the predicted value than those farther away. Thus, IDW assumes that each measured point has a local influence that diminishes with distance. It weighs the points closer to the prediction location greater than those farther away, hence the name inverse distance weighted.

A general form for IDW is:

$$Z_{s_0} = \sum_{i=1}^N \lambda_i Z_{s_i} \quad (5.1)$$

where Z_{s_0} represents the estimated value at location s_0 , N denotes the number of points around the estimated point and λ_i is:

$$\lambda_i = \frac{d_{i_0}^p}{\sum_{i=1}^N d_{i_0}^p} \quad (5.2)$$

where d_{i_0} is the distance between location s_0 and location s ($d_{i_0}^2 = (X_{s_0} - X_s)^2 + (Y_{s_0} - Y_s)^2$) and p is a positive real number, called the power parameter. The most common value

of p is 2. A general form of interpolating a value using IDW is:

$$Z = \frac{\sum_{i=1}^N \frac{Z_i}{d_i^2}}{\sum_{i=1}^N \frac{1}{d_i^2}} \quad (5.3)$$

Z is the value at an unknown location.

5.1.3.1.2 Spline Spline interpolation consists of the approximation of a function by means of series of polynomials over adjacent intervals with continuous derivatives at the end-point of the intervals. Smoothing spline interpolation enables to control the variance of the residuals over the data set. The solution is estimated by an iterative process. It is also referred to as the basic minimum curvature technique or thin plate interpolation as it possesses two main features: (a) the surface must pass exactly through the data points, and (b) the surface must have minimum curvature.

$$Z_0 = \frac{\sum_{i=1}^s Z_i \frac{1}{d_i^k}}{\sum_{i=1}^s \frac{1}{d_i^k}} \quad (5.4)$$

In this formula Z_0 is the estimated value of point 0; Z_i value of control point i ; d_i is the distance between control point i and point 0; s is the number of control point in estimating; k is a designated power.

5.1.3.1.3 Kriging Geostatistical methods create surfaces incorporating the statistical properties of the measured data. Because geostatistics is based on statistics, these methods produce not only prediction surfaces but also error or uncertainty surfaces, giving the user an indication of how good the predictions are. Kriging is divided into two distinct tasks: quantifying the spatial structure of the data and producing a prediction. Quantifying the spatial data structure, known as variography, is fitting a spatial dependence model to the data. To make a prediction for an unknown value for a specific location, Kriging will use the fitted model from the variography, the spatial data config-

uration and the values of the measured sample points around the prediction location. Geostatistical analyst provides many tools to help determine the parameters to be used and the defaults are also provided so that a surface can be created quickly. Kriging is a moderately quick interpolator that can be exact or smooth depending on the measurement error model. It is very flexible and allows the user to investigate graphs of spatial autocorrelation. Kriging uses statistical models that allow a variety of map outputs including predictions, prediction standard error, standard error of indicators, and probability. The flexibility of Kriging can require a lot of decision making. Kriging assumes that the data comes from a stationary stochastic process. A stochastic process is a collection of random variables that are ordered in space and/or time such as elevation measurements. The selection of a Kriging method is based on the autocorrelation of radon concentrations between two points, which is formulated as follows:

$$Z(s) = \mu(s) + \epsilon(s) \quad (5.5)$$

$Z(s)$ consists of two parts: a deterministic trend $\mu(s)$, (i.e. flow direction) and a random auto correlated error $\epsilon(s)$. The symbol s simply indicates the location of a point.

5.1.3.2 Data preparation

ArcGis from ERSI were selected as GIS software for the implementation of the case study. A feature layer were constructed with the location of all meteorological station (Figure 5.6) and a spatial database (with ArcSDE) with all the available data (time stamp and temperature) as shown in Figure 5.7. In order to perform the interpolation methods for the temperature data, a join had to be done between the location and spatial data. The data had to be in the form as presented in Figure 5.8.

A limitation exist on the database software that the ArcGis use, Microsoft Sql Express. The limitation is that the database cannot have more than 254 different tables (a table for every different interpolation is needed). So with this implementation only 254 hours could be available for interpolation at any given time. So an alternative database

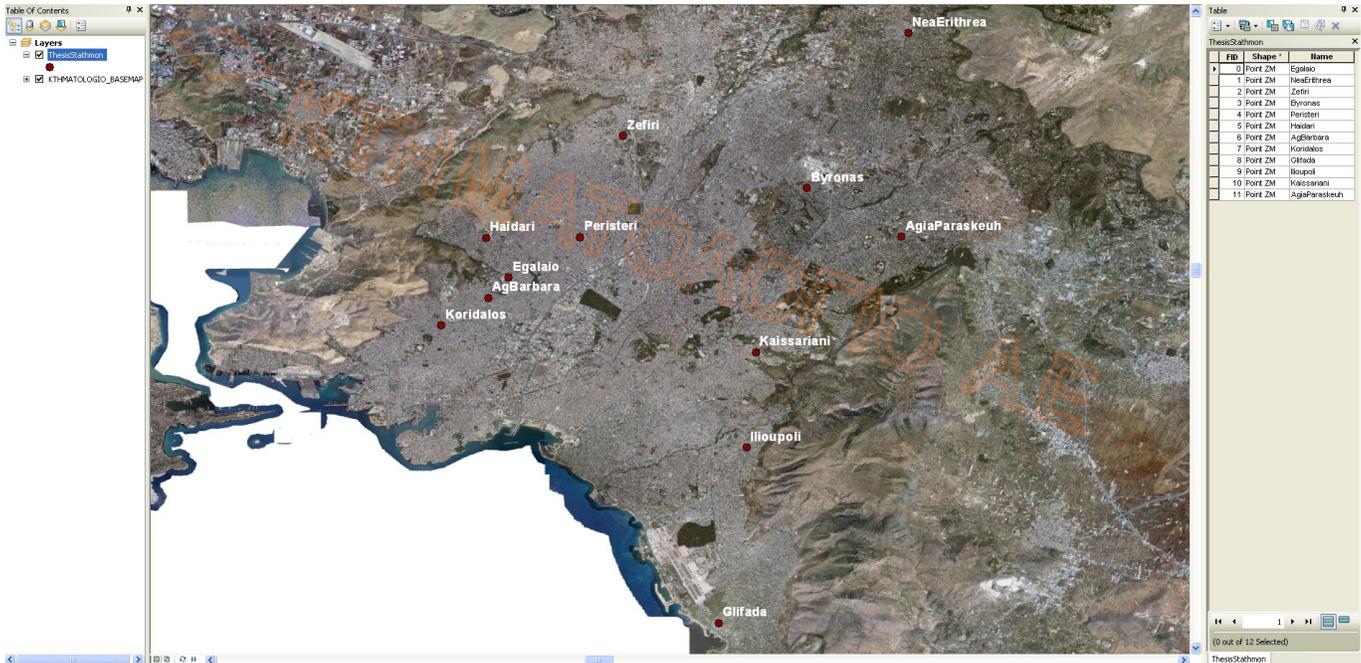


Figure 5.6: Location of meteorological stations

Data and Time	Aigalaio	Xaidari	Zefiri	Byronas	Peristeri	haidari	Koridalos	Glifada	Hlioupoli	Kaisariani	Nea Erithrea	Agia Barbar	Agia Paraskeuh
21/5/2009 11:00:00 μμ	21.769953	21.61681	21.654	22.94657	22.23525	21.61681	22.047433	22.309495	26.542248	21.027588	20.155758	21.8805	21.544513
22/5/2009	21.606688	21.304703	21.1185	22.552728	22.05775	21.304703	21.606665	22.063258	26.405338	20.702445	19.557403	21.5265	21.289995
22/5/2009 1:00:00 πμ	21.251013	21.071633	21.13475	22.204115	21.615	21.071633	21.494963	21.886078	26.24149	20.751815	19.28787	21.3495	21.032195
22/5/2009 2:00:00 πμ	21.028258	20.800855	21.06575	22.045915	21.3495	20.800855	21.287113	21.763143	26.100483	20.605145	19.300333	21.1725	20.869228
22/5/2009 3:00:00 πμ	20.47468	20.281368	20.1465	21.989015	20.9955	20.281368	20.9219	21.12968	26.041165	20.582985	19.270825	20.6415	20.75468
22/5/2009 4:00:00 πμ	20.356365	20.217793	19.92775	21.785703	20.553	20.217793	20.81481	20.811645	25.97855	20.146123	18.685623	20.553	20.477875
22/5/2009 5:00:00 πμ	20.3678	20.293145	19.367	21.424588	20.46475	20.293145	20.847388	20.541865	25.915593	19.759038	18.78514	20.553	20.365655
22/5/2009 6:00:00 πμ	20.316028	20.23601	20.3	21.31045	20.3765	20.23601	20.790855	20.532558	25.88117	20.252993	19.17829	20.553	20.518618
22/5/2009 7:00:00 πμ	20.88328	20.709293	20.73325	21.678203	21.97	20.709293	21.563335	21.11959	25.872655	21.171383	19.050613	21.3495	21.171558
22/5/2009 8:00:00 πμ	21.755748	21.425788	22.1405	22.136408	24.9125	21.425788	22.048518	22.55514	25.908565	21.967938	19.704983	22.23525	22.756743
22/5/2009 9:00:00 πμ	23.05554	22.482118	24.18575	23.52725	27.26025	22.482118	22.825883	23.801523	26.11088	22.777048	20.690783	23.21275	24.762103
22/5/2009 10:00:00 πμ	24.661925	23.907595	26.1605	26.27978	27.4425	23.907595	24.130975	24.742388	26.365583	23.75216	21.675145	24.64175	25.698118
22/5/2009 11:00:00 πμ	26.144168	25.289858	27.32825	27.0591	27.8995	25.289858	25.205673	25.614493	26.676013	24.917248	22.691385	25.80975	25.51686
22/5/2009 12:00:00 μμ	27.332755	26.1554	28.722	27.594028	28.725	26.1554	25.984548	26.766043	25.809713	25.889448	23.656675	26.98725	25.355305
22/5/2009 1:00:00 μμ	28.080118	27.06998	29.6565	27.867653	28.633	27.06998	26.619458	27.687633	26.22747	26.63995	24.658773	27.44225	25.368948
22/5/2009 2:00:00 μμ	28.715343	28.15384	29.9075	27.833345	28.817	28.15384	27.158675	28.540935	26.756178	26.914068	25.782033	27.8075	25.134058
22/5/2009 3:00:00 μμ	28.515808	29.89538	30.43775	27.038953	28.541	29.89538	27.245563	29.186635	27.11719	26.858418	26.771758	27.899	25.110513
22/5/2009 4:00:00 μμ	28.574678	31.800055	30.91625	26.371588	28.449	31.800055	27.62105	29.26697	27.65375	26.918923	27.623223	28.082	25.153555
22/5/2009 5:00:00 μμ	29.197993	32.39274	30.453	26.69201	28.2655	32.39274	27.82581	29.254233	27.738815	26.773768	28.067783	28.2655	25.23156
22/5/2009 6:00:00 μμ	28.588888	32.285633	29.44	26.56051	27.62475	32.285633	27.542063	28.27021	27.86629	26.391625	28.230675	27.9905	25.108313
22/5/2009 7:00:00 μμ	27.543965	31.519053	28.232	26.082473	26.98725	31.519053	27.6625	27.460893	27.172888	25.704163	27.902358	28.35725	24.719573
22/5/2009 8:00:00 μμ	26.317923	29.04116	26.07775	25.44057	25.99025	29.04116	26.713278	26.349645	25.128118	24.839003	25.123613	26.3535	24.153613
22/5/2009 9:00:00 μμ	25.112623	26.017388	24.80025	24.902613	25.18	26.017388	25.511048	25.535083	23.886348	23.898405	23.33056	24.91075	23.65175
22/5/2009 10:00:00 μμ	24.167265	24.585195	23.64775	24.451423	24.46225	24.585195	24.601628	24.601385	23.302978	23.0097	22.324665	24.10425	23.38967
22/5/2009 11:00:00 μμ	23.267083	23.575533	23.08325	24.095538	23.74725	23.575533	24.258063	23.82397	22.355278	22.332948	21.77035	23.30175	23.15419

Figure 5.7: Spatial database

ID	Shape	Name	OID	A051900000	A052001000	A052002000	A052003000	A052004000	A052005000	A052006000	A052007000	A052008000	A052009000	A052010000	A052011000	A052012101	A052013000	A052014000	A052015000
0	Point ZM	Egaleio	0	22,87595	22,62929	22,32204	21,9125	21,08853	20,72897	20,09004	19,93447	20,5388	21,82303	22,30446	24,63715	25,25796	25,7703	25,52661	25,76284
1	Point ZM	Nea Erithrea	1	20,56664	19,91632	19,4534	19,18324	18,47625	18,1166	17,7216	17,60598	18,07723	18,92925	20,18556	21,54886	22,64558	22,94533	23,73729	24,30448
2	Point ZM	Xalandri	2	22,58485	22,39166	22,04773	21,54749	20,43075	20,2178	19,74126	19,61474	20,29427	22,4941	23,73819	24,46079	24,94011	25,69404	26,3219	26,72519
3	Point ZM	Zefiri	3	21,9802	21,26875	20,85375	20,41725	20,218	19,736	19,493	19,49875	20,01375	20,6205	23,219	26,437	27,9875	28,94625	29,83325	28,72575
4	Point ZM	Byronas	4	23,38802	23,07447	22,81017	22,33112	22,06834	21,6867	21,32884	21,11621	21,38317	21,84986	24,45417	26,14292	27,31725	27,44713	27,00617	26,42583
5	Point ZM	Peristeri	5	23,034	22,5015	21,989	21,615	21,1725	20,845	20,5825	20,553	21,261	23,751	26,8055	26,867	26,3575	27,9905	28,35725	27,35125
6	Point ZM	Haidari	6	22,58485	22,39166	22,04773	21,54749	20,43075	20,2178	19,74126	19,61474	20,29427	22,4941	23,73819	24,46079	24,94011	25,69404	26,3219	26,72519
7	Point ZM	Agia Varvara	7	22,679	22,679	22,05775	21,7035	20,9955	20,64175	20,0235	19,93525	20,73	22,59075	23,83625	25,3625	24,6415	25,09025	25,539	25,26975
8	Point ZM	Koridalos	8	22,74248	22,32963	21,92893	21,58154	20,98032	20,41433	20,08965	20,60929	21,33352	25,51317	26,35386	23,94708	24,38529	24,95225	25,12707	25,31147
9	Point ZM	Glyfada	9	22,99371	22,3271	22,12951	22,04196	21,73608	21,1767	20,85311	21,03435	20,95195	21,19395	22,56893	24,42204	25,61556	26,40402	26,48012	26,98401
10	Point ZM	Hlioupoli	10	22,00356	21,36609	20,93916	20,60443	20,22623	19,84467	19,44451	19,16406	19,37223	19,93554	21,99375	24,19639	24,6788	25,75544	27,03359	27,57286
11	Point ZM	Kaisariani	11	22,03052	21,43254	21,06661	20,67783	20,38938	19,66924	19,20107	19,23242	19,86627	20,11078	22,05031	23,73216	25,0352	24,94519	25,28816	24,60233
12	Point ZM	AgiaParaskeu	12	22,19153	21,77564	21,46407	21,19315	20,85779	20,60729	20,39818	20,50151	21,14269	21,69337	23,42486	24,4064	24,70079	24,33476	24,00758	23,56739

Figure 5.8: Data structure for interpolation

was investigated in order to lift the above limitation. The Microsoft Sql Server edition was selected with the cooperation of a special feature of ArcGis software, ArcSDE spatial view⁴. With that specific feature was possible to make a left outer join (⁵) between the tables (part of the feature layer) with the location information of each meteorological station and the table with temperature from the database. The time stamp of each measurement set was selected to be the name of the table with the following convention: character 'A'+ 'month +day of the month' + 'time in 24-hour format'. Character 'A' was needed due to the name convention that the program use. All the above procedure had to be done in order to transform the data in a compatibly form for the software to be able to execute different methods of interpolation.

5.1.3.3 Analysis

5.1.3.3.1 Interpolation methods comparison After bibliographic research [4], three different methods of interpolation were selected to be tested as more suitable for temperature prediction. The three interpolation method selected are: 1)IDW, 2)spline and 3)Kriging. The following procedure were selected in the test. A subset of entire dataset was selected in order to test the accuracy of the different methods of interpolation. The subset consist 384 measurements from 11-05-2008 to 27-05-2008. A randomly selected meteorological station (Egaleo) was exclude from the data as a test station. For every one of the three interpolation methods and the whole subset, the predicted value from the interpolation was extracted from the interpolation images-rasters and compared with the measured temperature value. Preliminary test were conducted in order to find the optimum values for the different parameters of each interpolation method ie number of points, maximum distance etc. Due to the large number of data and computational power-time needed for the interpolations, a script for each interpolation method was developed using Python as programming language (for more information about

⁴ArcSDE Spatial Views : tool for organizing information from multiple feature classes and geodatabase tables into a single "virtual feature class" or table at the database level.

⁵left outer join for table A and B always contains all records of the "left" table (A), even if the join-condition does not find any matching record in the "right" table (B).

the script see Appendix A) for automation the process. Mean Absolute Error (MAE) and Mean Square Root Error (MSRE) are chosen as the two indicators to evaluate the precision of the interpolation methods. The results are shown in Table 5.1 (the values are in $^{\circ}C$). Firstly, the errors of the four interpolation methods are not great. The smallest MSRE is about $1^{\circ}C$, the greatest MSRE $1.46^{\circ}C$; the smallest MAE is about $0.68^{\circ}C$, the greatest MAE $0.95^{\circ}C$. Kriging-spherical and Kriging-circular are the best two methods of interpolation for our case, spline is the second and IDW interpolation method has the lowest precision. It can be clearly seen from the above results, Kriging-spherical was selected as interpolation method for the rest of the study.

Interpolation Method	MAE	MSRE
Kriging Spherical	0.68	1.03
Kriging Circular	0,72	1,08
IDW	0.78	1.22
Spline	0.95	1.46

Table 5.1: Interpolation Comparison

5.1.3.3.2 Data representation As a result from the previous investigation, Kriging Spherical was selected as the more suitable method of temperature interpolation-prediction for our study. Later on using the ArcSDE a join was made using the hole dataset - 2481 hourly temperature measurements for all thirteen meteorological stations with their location. In this step all the available meteorological stations are included in the interpolation proses. The Spherical Kriging interpolation Python script was properly modified in order to access the spatial data from the SQL server database in order to perform the interpolation. As a result from the processes 2481 different interpolation images were constructed for all the available dataset. Figure 5.9 shows an interpolation image with one of the highest days Figure 5.10 shows an average day and Figure 5.11 shows one of the coolest day on the dataset.

A video-animation was constructed will all the interpolation images. This was possible using a new feature of the GIS software called time aware raster. The animation

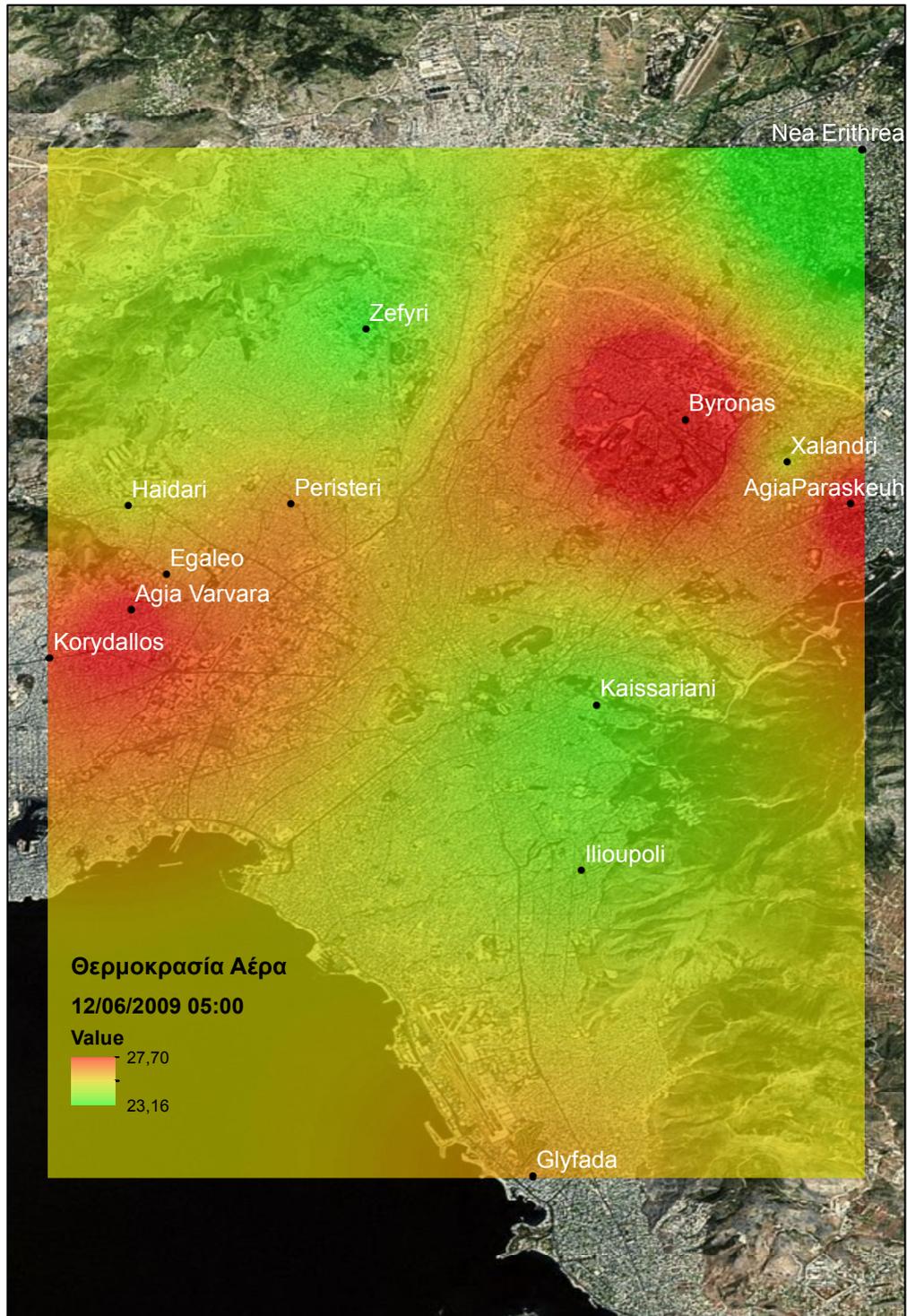


Figure 5.9: Temperature interpolation 12/06/2009 05:00

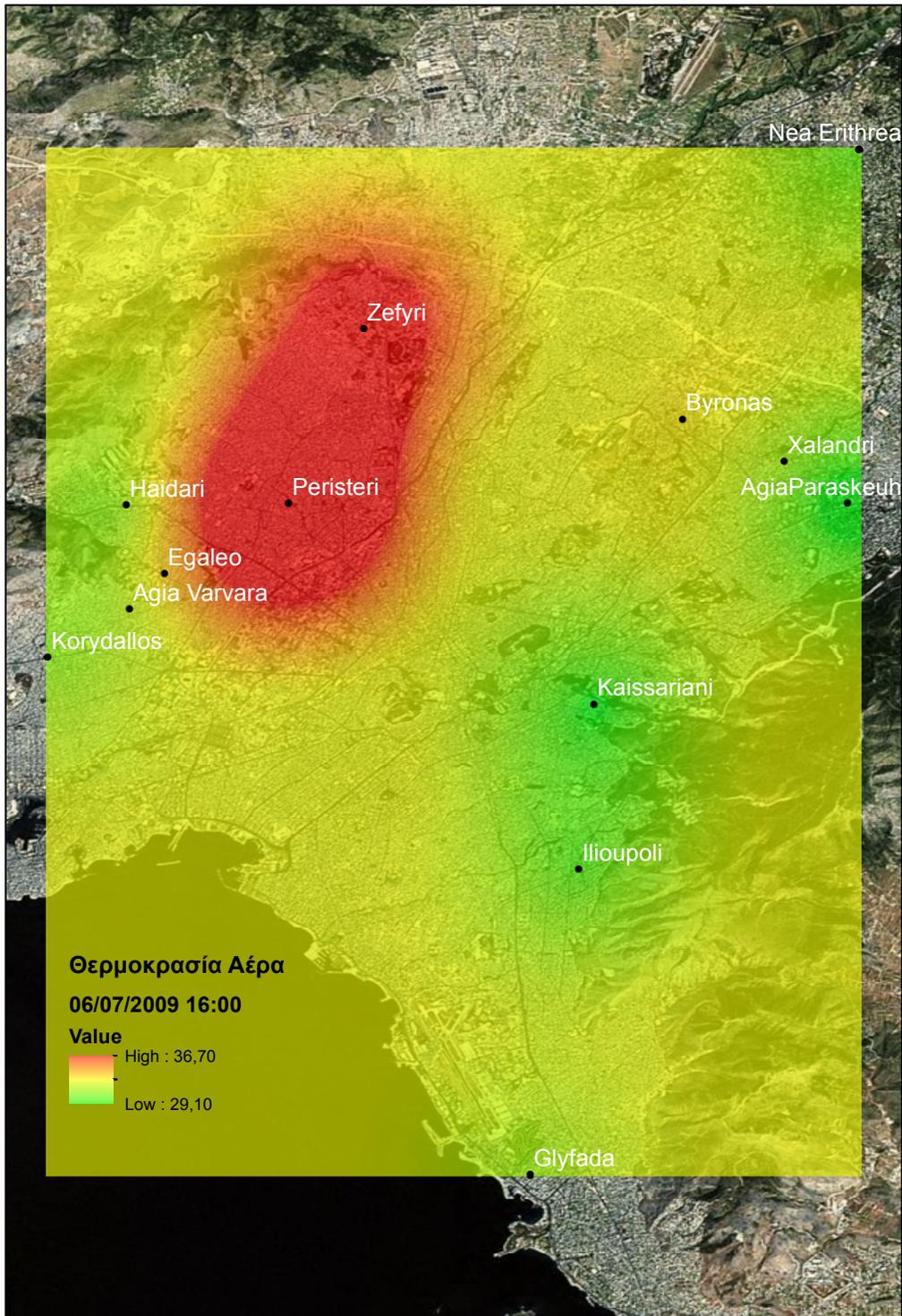


Figure 5.10: Temperature interpolation 06/07/2009 16:00

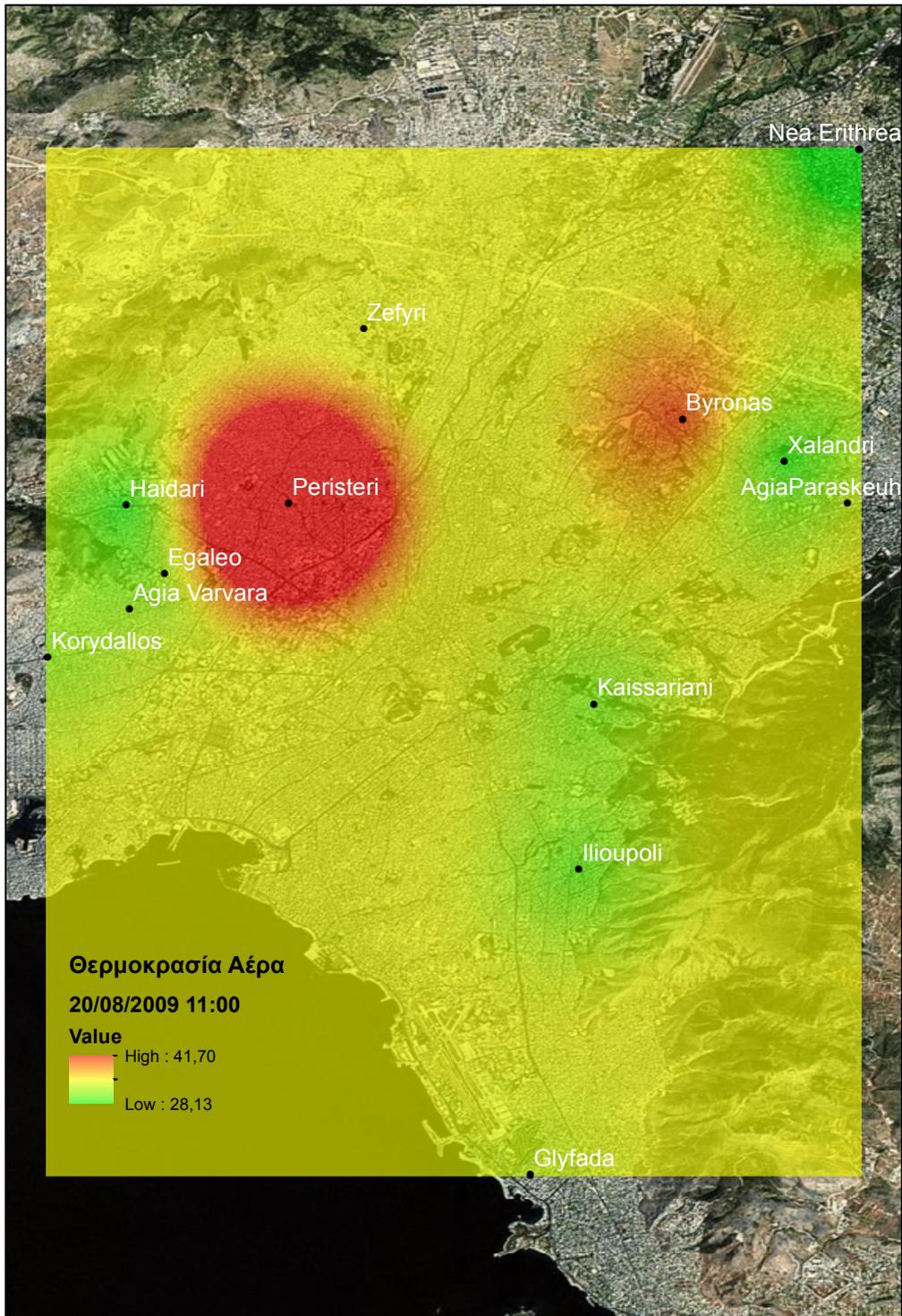


Figure 5.11: Temperature interpolation 20/08/2009 11:00

created is a great tool to visual examine the evolution of the UHI phenomenon during the small periods (day) and longer periods (months) of time. Also the animation can be used for the information of the general public about the UHI phenomenon.

Chapter 6

Summary

6.1 Future research prospects

While this thesis examines the prediction using ANN and visualization of the UHI in GAA , there are still possibilities for further study of this phenomenon.

1. Examine the association between the UHI intensity, electricity demand[1] and population .
2. On line implementation of the prediction and visualisation.
3. Integration with an urban DSS (Decision support system) system.
4. Year around temperature measurements for verification of the predictive ability of ANN on the autumn, winter, spring season.
5. Measurement of relative humidity[9] and wind speed[21] in order to examine their impact and the potential prediction in UHI intensity.

6.2 Conclusion

Important heat island studies have been performed in Europe during the last decades showing that the deep understanding of the phenomenon plays an important role in

fighting its consequences to the climate change. Advanced artificial intelligence techniques such as neural networks offer on the other hand a valuable tool to be used for the prediction of the specific phenomenon. The neural networks prediction accuracy is mainly based on the quality and quantity of the available data. The aim of the present thesis was to investigate the feasibility of predicting the urban heat island phenomenon using a limited data series. The Athens case study was used to demonstrate the feasibility and accuracy of the overall approach. The methodology presented in the present thesis showed that the urban heat island intensity can be predicted quite accurately for at least a 24-h prediction horizon using a limited set of data. Therefore the NN prediction methodology can be an important tool for peak energy load predictions during heat waves and hot summer days contributing to the demand and supply energy management.

In this thesis, we compared different spatial interpolation techniques for spatial temperature prediction. Using GIS and interpolation techniques a new innovative way was presented to visualize the UHI. As the above investigation using ANN; the main goal was to investigate methods in order to produce sufficiently accurate results with limited data. The animation produced can be a very effective tool for the information of the general public about the problem of UHI.

In conclusion, the final words of this thesis should come from the doyen of urban climatology, Helmut Landsberg[13], who stresses this importance of applying the knowledge gained from academic work on urban climates:

The knowledge we have acquired about urban climates should not remain an academic exercise. It should be applied to the design of new towns or the reconstruction of old ones.

Appendix A

Python code for Kriging Spherical interpolation method

```
#  
# The feildName is a shapefile with the locations of the stations and with  
# temperature  
#  
#  
#  
# Import system modules  
import arcpy  
from arcpy import env  
from arcpy.sa import *  
from numpy import *  
import xlrd  
import xlwt  
  
#
```

```

# Initialitation
#
method="krigingSphericaPoint6"
workspace="c:\\TempInter\\testDiaforeInterpolation\\"+method+".gdb"
scratch="c:\\tempInter\\temp.gdb"
feildName="ThesisStathmonOutEgaleo" #shapefile name
#for getting the temperature 23.667893 37.997403
x=23.667893
y=37.997403
xlsOutfilename=workspace+"\\ "+method+".xls"
#xlsFilename=workspace+"\\allw.xls"
xlsFilename="C:\\TempInter\\testDiaforeInterpolation\\egaleo.xls"

# set the workspace
arcpy.env.workspace=workspace
arcpy.env.scratchWorkspace=scratch
# overite data
arcpy.env.overwriteOutput = True
# get all the feilds
filds= arcpy.ListFields(feildName)
# rid of the 4 colus and hold olny the temperature values >filds[0].name #=
filds = filds[4:len(filds)]

#calculate the inderpolatein
for fild in filds:
#     a= Idw(feildName ,fild.name)

```

```

a= Kriging(feildName , fild .name, "Spherical" ,0.0007 ,RadiusFixed (8))
a .save (workspace+"\\ "+fild .name)

# get the rasterNames
data = arcpy .ListDatasets ("*" ,"Raster")
filanemes=[]
resualts=[] #ones ((1 ,len (data)) ,dtype=float )
# get cell value
for tx in range (0 ,len (data)):
    resualts .append (arcpy .GetCellValue_management (data [tx] , str (x)+"_" +str (y)
    filanemes .append (data [tx ])

#get the real values of the point
book = xlrd .open_workbook (xlsFilename)
sheet = book .sheet_by_index (0)
colValues= sheet .col_values (0 ,0)

diafora= ones ((1 ,len (colValues)) ,dtype=float )

for tt in range (0 ,len (colValues)):
    diafora [0 ,tt]= float (str (colValues [tt]))- float (str (resualts [tt ]))

#save data to the excel file
book = xlwt .Workbook (encoding="utf -8")
sheet = book .add_sheet ("Resaults_" +method)

```

```
sheet.write(0,0,"Data")
sheet.write(0,1,"x=" + str(x))
sheet.write(1,1,"y=" + str(y))
sheet.write(0,2,method)

for x in range(2,len(results)):
    sheet.write(x,0,filanemes[x])
    sheet.write(x,1,float(str(results[x])))
    sheet.write(x,2,float(str(colValues[x])))
    sheet.write(x,3,float(str(colValues[x])) - float(str(results[x])))

book.save(xlsOutfilename)
```

Bibliography

- [1] H Akbari. Energy Saving Potentials and Air Quality Benefits of Urban Heat Island Mitigation. *Lawrence Berkeley National Laboratory LBNL Paper LBNL58285*, pages 1–19, 2005.
- [2] J.A. Anderson, E. Rosenfeld, and Andras Pellionisz. *Neurocomputing*. Number τ . 1 in Bradford Books. MIT Press, 1993.
- [3] E. Anderson, J. A. and Rosenfeld. *Neurocomputing: Foundations of Research*. MIT Press, Cambridge, 1988.
- [4] Wenjing Cao, Jinxing Hu, Xiaomin Yu, and A Interpolation Methods. A Study on Temperature Interpolation Methods Based on GIS. In *7th International Conference on Geoinformatics*, number 4, pages 0–4, 2009.
- [5] Scott E Fahlman and Christian Lebiere. The Cascade-Correlation Learning Architecture. *Defense*, (4976), 1991.
- [6] Martin A Fischler and Oscar Firschein. *Intelligence: the eye, the brain, and the computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1987.
- [7] C Gordon and James McDaid Kapetsky. Land use planning for aquaculture: A West African case study. Technical report, Food and Agriculture Organization of the United Nations, 1991.
- [8] M T Hagan and M B Menhaj. Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, 1994.

- [9] I. Ibrahim and A. A. Samah. Preliminary study of urban heat island: Measurement of ambient temperature and relative humidity in relation to landcover in kuala lumpur. 2011.
- [10] Zhang Y Hu X Zeng Y Tan J Shao D Jiang D. Progress in developing an ANN model for air pollution index forecast. *Atmospheric Environment*, 38(40 SPEC.ISS.):7055–7064, 2004.
- [11] P Kassomenos, H A Flocas, S Lykoudis, and M Petrakis. Analysis of Mesoscale Patterns in Relation to Synoptic Conditions over an Urban Mediterranean Basin. *Theoretical and Applied Climatology*, 59:215–229, 1998.
- [12] Davies M Croxford B Bhuiyan S Mavrogianni A Kolokotroni M. A validated methodology for the prediction of heating and cooling energy demand for buildings within the Urban Heat Island: Case-study of London. *Solar Energy*, 84(12):2246–2255, 2010.
- [13] H E Landsberg. *The urban climate*. International geophysics series. Academic Press, 1981.
- [14] I Livada, M Santamouris, K Niachou, N Papanikolaou, and G Mihalakakou. Determination of places in the great Athens area where the heat island effect is observed, 2002.
- [15] Khan M R Abraham A Maqsood I. An ensemble of neural networks for weather forecasting. *Neural Computing and Applications*, 13(2):112–122, 2004.
- [16] Demosthenes N. Asimakopoulos Matheos Santamouris. *Energy and climate in the urban built environment*. Routledge, 2001.
- [17] W S McCulloch and W Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5(4):115–133, 1943.

- [18] Martin Fodslette Meiller. A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. 6:525–533, 1993.
- [19] G. Mihalakakou, M. Santamouris, N. Papanikolaou, C. Cartalis, and a. Tsangrasoulis. Simulation of the Urban Heat Island Phenomenon in Mediterranean Climates. *Pure and Applied Geophysics*, 161(2):429–451, feb 2004.
- [20] C J G Morris and I Simmonds. Associations between varying magnitudes of the urban heat island and the synoptic climatology in Melbourne, Australia. *International Journal of Climatology*, 20(15):1931–1954, 2000.
- [21] C J G Morris, I Simmonds, and N Plummer. Quantification of the Influences of Wind and Cloud on the Nocturnal Urban Heat Island of a Large City. *Journal of Applied Meteorology*, 40(2):169–182, 2001.
- [22] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In *IEEE International Conference on Neural Networks*, pages 586–591. IEEE.
- [23] M Santamouris, K Paraponiaris, and G Mihalakakou. Estimating the ecological footprint of the heat island effect over Athens, Greece, 2007.
- [24] M Santamouris, K Pavlou, A Synnefa, K Niachou, and D Kolokotsa. Recent progress on passive cooling techniques. Advanced technological developments to improve survivability levels in low-income households, 2007.
- [25] Brian a. Smith, Gerrit Hoogenboom, and Ronald W. McClendon. Artificial neural networks for automated year-round temperature prediction. *Computers and Electronics in Agriculture*, 68(1):52–61, aug 2009.
- [26] Data Tables. World Urbanization Prospects The 2009 Revision Highlights. *Prospects*, 2009(February):1–56, 2009.

- [27] Imran Tasadduq and Shafiqur Rehman. Application of neural networks for the prediction of hourly mean surface temperatures in Saudi Arabia. *Renewable Energy*, 25(767):545–554, 2002.
- [28] P J Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.
- [29] Prybutok V R Yi J. A neural network model forecasting for prediction of daily maximum ozone concentration in an industrialized urban area. *Environmental Pollution*, 92(3):349–357, 1996.