Application Of Neural Networks In The Determination Of The Effective Service Cell Area In Wireless Cellular Systems

By
Chrisanthi Lytra

MSc Dissertation

Department Of Electronic And Computer Engineering
Technical University Of Crete
September, 2003

INTRODUCTION **4**

CHAPTER 1 .... **5**

CELLULAR CONCEPT- THE CELL COVERAGE PROBLEM **5**

1.1 EVOLUTION OF MOBILE COMMUNICATIONS 5
1.2 THE CELLULAR CONCEPT 6
1.3 CELLULAR NETWORK ARCHITECTURES 10
1.4 THE CELL COVERAGE ESTIMATION PROBLEM 11
1.5 DIFFICULTIES IN DETERMINING THE EFFECTIVE CELL AREA 12
1.6 EVOLUTION IN COVERAGE ESTIMATION TECHNIQUES 13

CHAPTER 2 .... **15**

NEURAL NETWORKS **15**

2.1 BASIC THEORY AND OPERATION 15
2.2 NEURAL NETWORK ARCHITECTURES AND TRAINING ALGORITHMS 17

CHAPTER 3 .... **23**

APPLICATION OF NEURAL NETWORKS IN MOBILE COMMUNICATION SYSTEMS **23**

3.1 NEURAL NETWORKS IN SIGNAL DETECTION 23
3.2 NEURAL NETWORKS IN ADAPTIVE EQUALIZATION FOR DIGITAL MOBILE RADIO CHANNELS 25
3.3 ADAPTIVE ANTENNA ARRAY BEAMFORMING AND DIRECTION OF ARRIVAL ESTIMATION USING NEURAL NETWORKS 29
3.4 NEURAL NETWORKS IN RADIO RESOURCE MANAGEMENT 34
3.6 NEURAL FRAUD DETECTION IN MOBILE-PHONE SYSTEMS 37
3.8 MOBILE STATION LOCATION USING NEURAL NETWORKS 44
3.9 NEURAL NETWORKS IN CELL PLANNING 48

CHAPTER 4 .... **52**

APPLICATION OF NEURAL NETWORKS TO THE CELL COVERAGE ESTIMATION PROBLEM IN A CELLULAR COMMUNICATION SYSTEM **52**

4.1 EXPERIMENTAL PROCEDURE 52
4.2 THE EXAMINATION AREA 54
4.3 ASSUMPTIONS 55
4.4 NEURAL NETWORK SIMULATION 56
CONCLUSIONS 64
REFERENCES 64

Introduction

In recent years with the rapid growth and need for high quality and high capacity cellular networks, estimating coverage accurately has become extremely important. The cell coverage estimation problem is of great importance to the cellular network designer in order to provide effective cell planning. Traditional estimation techniques are based on propagation models. The generic RF propagation prediction algorithms based on computer databases or empirical results give only approximate coverage, and are not suitable for detailed network design. To more accurately design the coverage of modern cellular networks, signal strength measurements must be taken in the service area. Taking signal strength measurements is an expensive and a time consuming task.

The need for more powerful tools that take into account real signal strength measurements and are able to provide quick and accurate cell coverage information has become increasingly important.

More recently, biologically inspired (bio-inspired) techniques such as fuzzy logic, neural networks, simulated annealing, genetic algorithms, evolutionary computing models, and other bio-inspired techniques have been used to solve problems in a number of key areas in communications. Most of the bio-inspired methods are considered to be ``intelligent'' because of their capability in adapting to changes in the environment that were not predicted in advance. In the last years, neural networks have been applied to many mobile communications aspects.

This dissertation addresses the issue of estimating the effective cell coverage in a cellular communication system, by using an appropriately trained neural classifier. Pattern classification is a very powerful means used in the classification of known situations and the prediction of the category (class) in which a new, 'a priori' unknown situation should be assigned at. The purpose is to define the actual service area of each BS using such previous knowledge based on signal strength measurements, obtained in certain locations for an urban environment, which is passed through pattern classifiers.

The thesis consists of four chapters. The first chapter contains a short description of the cellular concept and the cell coverage estimation problem. In the second chapter a brief presentation of the most popular neural network architectures takes place as well as a reference to their main advantages and disadvantages. Applications of these neural networks mostly found in the cellular mobile communication networks literature are reviewed in the third chapter. In the last chapter the application of neural classifiers to the cell coverage estimation problem is presented. The training procedure of the neural networks is described and results are presented.

Chapter 1
Cellular Concept- The cell Coverage Problem

This chapter begins with a brief reference to the evolution of mobile communications. Then the cellular concept is described, and a reference to the basic cellular network follows. Consequently, the cell coverage estimation problem is defined by presenting the difficulties in determining the effective cell area. A short description of the evolution in cell coverage estimation techniques follows. Finally the goal of this dissertation is presented.

1.1 Evolution of Mobile Communications

The objective of early land to mobile radio systems was to achieve a large coverage area by using a single, high powered transmitter with an antenna mounted on a tall tower.

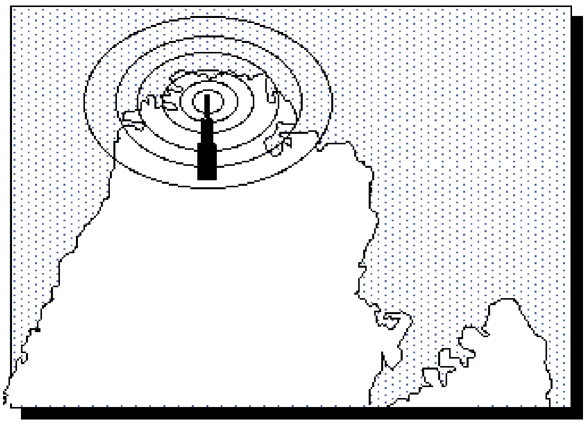

Figure 1.1: Early mobile telephone system architecture

This was done by selecting several channels from a specific frequency allocation for use in autonomous geographic zones. The communications coverage area of each zone was usually planned to be as large as possible. The number of channels that could be obtained from the allocated spectrum was limited. There was generally no in system interference as the same frequencies were reused in the next service area which used to be several hundred miles away. Some of the drawbacks of the early land to mobile systems were limited service capability, high blocking probabilities, inefficient frequency spectrum utilization. If there were two contiguous service areas then call had to be terminated and reinitiated in the next service area. There was no concept of handoff.

Large scale integrated circuit technology reduced the size of mobile transceivers to one that could easily fit into the standard automobile. Another factor was the reduction in price of the mobile telephone unit. Technology, feasibility and service affordability caused the transition from early land to mobile systems to the cellular systems.

Mobile communications is currently at its fastest growth period in history, due to enabling technologies which permit wide spread deployment. Historically, growth in the mobile communications field has come slowly, and has been linked to technological advancements. The tremendous growth in the mobile communications is primarily due to development of highly reliable, miniature solid state devices and the development of the cellular concept which was conceived in Bell Laboratories in the 1960s and 1970s. The future growth of consumer-based mobile and portable communication systems will depend on radio spectrum allocations, regulatory decisions, adoption of common standards, consumer needs and technology advances in the signal processing, access, and integration of voice and data networks.

1.2 The Cellular Concept

The cellular concept was a major breakthrough in solving the problem of spectral congestion and user capacity. It offered high capacity with a limited spectrum allocation without any major technological changes. The cellular concept is a system level idea in which a single, high power transmitter (large cell) is replaced with many low power transmitters (small cells).

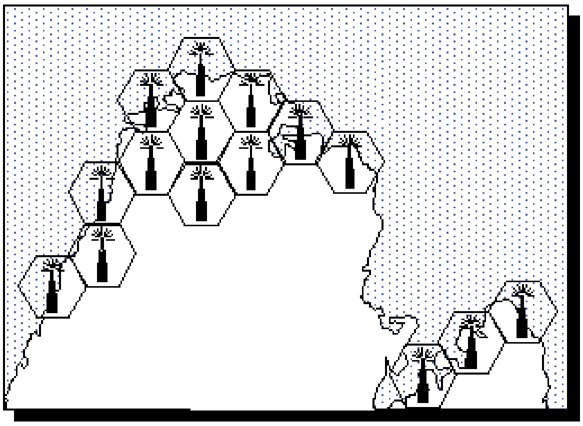

Figure 1.2 : Mobile telephone system using a cellular architecture

The area serviced by a transmitter is called a cell. Each small powered transmitter, also called a base station provides coverage to only a small portion of the service area. Base stations close to one another are assigned different groups of channels so that all the available channels are assigned to a relatively small number of neighboring base stations. Neighboring base stations are assigned different groups of channels so that the interference between base stations is minimized. By symmetrically spacing base stations and their channel groups throughout a service area, the available channels are distributed throughout the geographic region and may be reused as many times as necessary, so long as the interference between co-channel stations is kept below acceptable levels.

The hexagonal cell shape shown in fig. 1.3b is conceptual and is a simplistic model of the coverage for each base station. The hexagon has been universally adopted since the hexagon permits easy and manageable analysis of a cellular system. Also considering geometric shapes which cover an entire region without overlap and

with equal area, hexagon has the largest area considering the distance between the center of a polygon and its farthest perimeter points.
The real coverage pattern in which a given transmitter serves the mobiles successfully, is not symmetrical in any way (Figure 1.3).

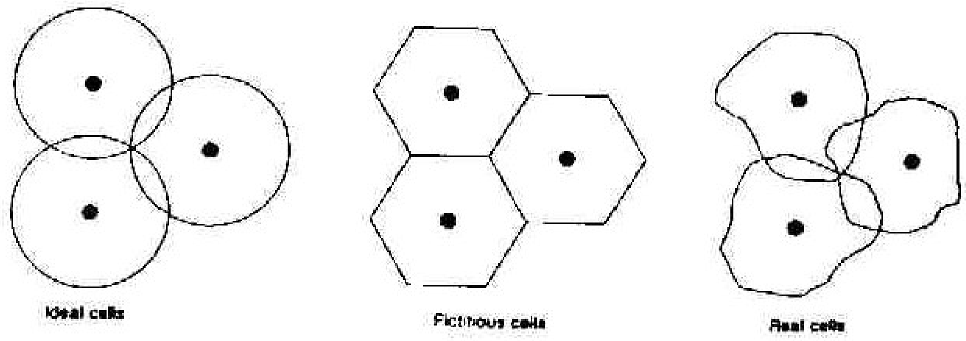

Figure 1.3 : Depiction of cells in the mobile system. a) ideal cells, b) fictious cells, c) real cells.

As the demand for service increases, the number of base stations may be increased, thereby providing additional capacity with no increase in radio spectrum. This fundamental principle is the foundation for modern mobile communication systems, since it enables a fixed number of channels to serve an arbitrarily large number of subscribers by reusing the channels throughout the region. The concepts of frequency reuse, handoff and system capacity are briefly explained.

Frequency Reuse :The base station antennas are designed to achieve the desired coverage within the particular cell. By limiting the coverage area to within the boundaries of a cell, the same group of channels may be used to cover different cells that are separated from one another by distances large enough to keep interference levels within tolerable limits. The design process of selecting and allocating channel groups for all the cellular base stations within a system is called frequency reuse or frequency planning.
In Fig. 1.4 the cells labeled with the same letter use the same group of channels.

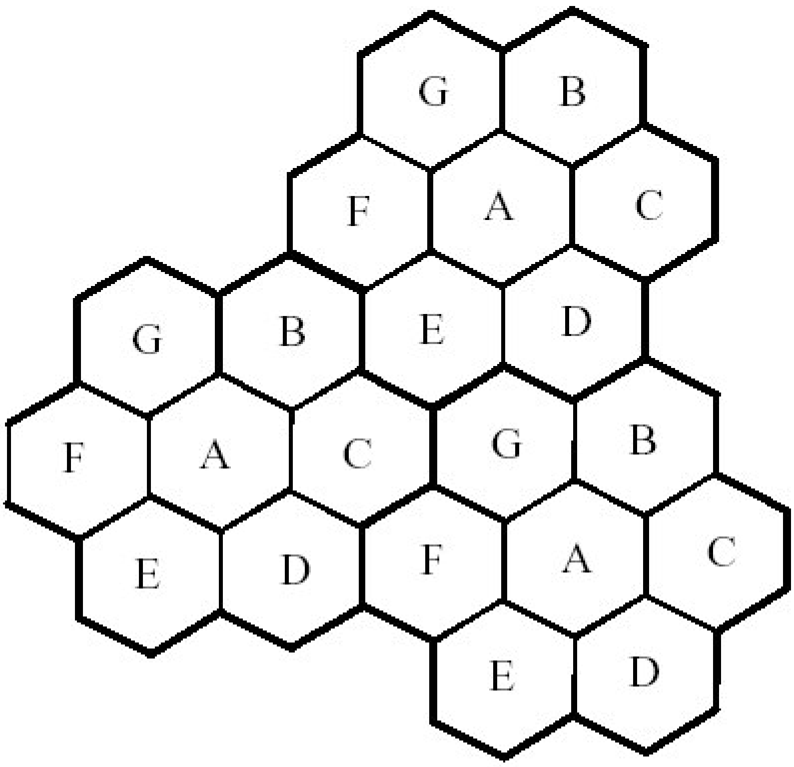

Figure 1.4 : Diagram of cellular frequency reuse.Cells with the same letter use the same set of frequencies. The reuse shown here is for a cluster size of N=7.

Handoff : When a mobile moves into a different cell while a call is in progress, the mobile switching center (MSC) automatically transfers the call to a new channel belonging to the new base station. This handoff operation involves identifying a new base station, assigning a free channel in the new cell to the mobile to change the frequency and transfer the voice circuit to the new base station. Processing handoffs is an important task in any cellular radio system. The handoff process can be performed based on several criteria such as signal strength, bit error rate in digital systems or interference levels. For example, if the signal level is used to trigger the handoff, an optimum signal level at which to initiate handoff is specified which approximately corresponds to the boundary of the cell. Once particular signal level goes below the specified threshold the base station queries the received power from the mobile at the different neighboring base stations, and picks a base station which has a power higher than that seen in the serving base station by a specified margin.

Interference: Interference is the limiting factor in the performance of cellular systems. Sources of interference include another mobile in the same cell, a call in progress in a neighboring cell, or other base stations operating in the same frequency band. Interference on voice channels causes background noise and results in poor voice quality. On control channels, interference leads to missed and blocked calls due to errors in the digital signaling. Interference is the major bottleneck in increasing capacity of a cellular network. The major types of system generated cellular

interference are co-channel interference (cells that use the same set of frequencies) and adjacent channel interference.
The cluster size N determines the overall system capacity, since it represents the number of cells in a cluster. The sets of frequencies used in one cluster are repeated in the rest of the clusters.
Different techniques such as cell splitting, sectoring, power control and smart antennas can be used to increase the system capacity.

Cell splitting

Cell splitting is the process of subdividing a congested cell into smaller cells, each with its own base station and a corresponding reduction in antenna height and transmitter power. Cell splitting increases the capacity of a cellular system since it increases the number of times that channels are reused. By making these the new cells to have smaller radius than the original cells and by installing these smaller cells between the existing cells, capacity increases due to the additional number of channels per unit area. Cell splitting achieves capacity improvement by essentially re-scaling the system

Sectoring

Sectoring is another way to increase capacity. In this approach, capacity improvement is achieved by reducing the number of cells in a cluster and thus increasing the frequency reuse. The co-channel interference in a cellular system may be decreased by replacing a single omni-directional antenna at the base station by several directional antennas, each radiating within a specified sector. By using directional antennas, a given cell will receive interference and transmit with only a fraction of the available co-channel cells. The technique for decreasing co-channel interference and thus increasing system capacity by using directional antennas is called sectoring.

A cell is usually sectored into three sectored configurations. Base Stations are positioned at the intersection of three cells. The sector antennas at the base station radiate with a 120 degree arc into the required cell. In each base station the antennas are used to steer the signal in the required direction. The antennas are mounted high and are physically tilted downwards.
In the figure below the pie shaped pieces represent different cells each of which is served by a directional antenna of 120 degrees arc.

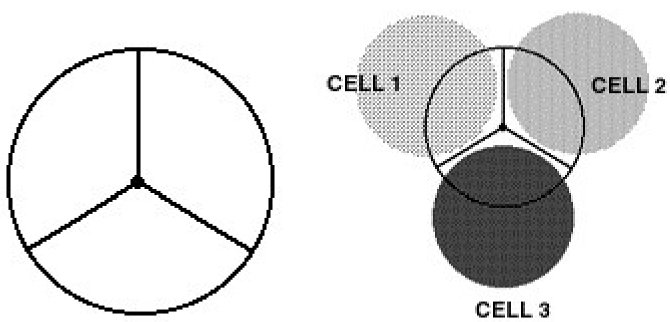

Figure 1.5: The cell-site location

1.3 Cellular network architectures

The radio propagation and related resource management characteristics are different in different cellular structures. Therefore, specific characteristics of cellular networks should be taken into account while examining their performance. Cellular networks can have various architectures the most basic of which are as follows.

Macrocellular networks

Macrocells are mainly used to cover large areas with low traffic densities. These cells have radii several kilometres (usually between 1 and 10 km). They can be classified into large and small macrocells.

Large macrocells have radii between 5 and 10 km and are usually used for rural areas. To ensure good radio coverage, base stations have to be installed in an efficient manner. Macrocellular systems need good planning, based on a combination of field measurements and theoretical modelling.

The radius of ***small cells*** lies between 1 and 5 km. These cells are used if the traffic density in large cells is so high that it will cause blocking of calls. Planning small cells is more difficult since traffic predictions for relatively small areas cannot be easily done. This is because the offered traffic per square kilometre is usually dynamic and will fluctuate more rapidly as the area over which the traffic is offered, become smaller.

Microcellular networks

Microcellular radio networks are used in areas with high traffic density; i.e. urban areas. The cells have radii between 200m and 1 km. In general, microcells increase capacity, but radio resource management becomes more difficult. This is because they are more sensitive to traffic and interference variations than macrocells. Moreover, the shape of the cell is time dynamic (i.e., the shape changes from time to time) due to propagation characteristics.

Microcells are characterised by lower output power and lower antenna placement compared to macrocells. They can be classified as one-, two-, or three dimensional, depending on whether they are along a road or a highway, covering an area (i.e. a number of adjacent roads), or located in multilevel buildings, respectively.

Picocellular networks

Picocells have radii between 10 and 200m. They are used for indoor applications and have three-dimensional structures. The use of fixed cluster sizes, fixed channel allocations, and prediction of traffic densities are difficult for indoor applications. Picocellular systems are used mainly for wireless office communications.

Multilayer networks

To improve capacity and coverage beyond the above referred structures, ***multi-layered networks*** may be used. These are defined as networks where different cells have overlapping coverage. For instance a macrocellular network may contain micro- and picocells.

The addition of smaller cells to give higher capacity for smaller coverage area, i.e., microcells and picocells (mostly used in indoor environments and public structures such as airports and busy railway stations), can provide the capacity and coverage for the network. In such an integrated cellular system (*multi-level cellular system*), the main goal is to provide a balance between maximising the number of users per unit area (favours small cells) and minimising the network control and handover rate (which favours larger cells). Therefore the upper layer of macrocells is used to provide an overall coverage area, and take control of fast moving mobile users. The lower layer of microcells focus on slow moving users moving between high-rise buildings, while the picocells focus mainly on stationary users with high bandwidth requirements.

By using different antenna heights (often on the same building or tower) and different power levels, it is possible to provide different sized cells which are co-located at a single location. This technique is called the *umbrella cell* approach and is used to provide large area coverage to high speed users while providing small area coverage to users travelling at low speed. The next figure presents an umbrella cell which is co-located with some microcells. The umbrella cell ensures that the number of handoffs is minimized for high speed users and provides additional microcell channels for pedestrian users.

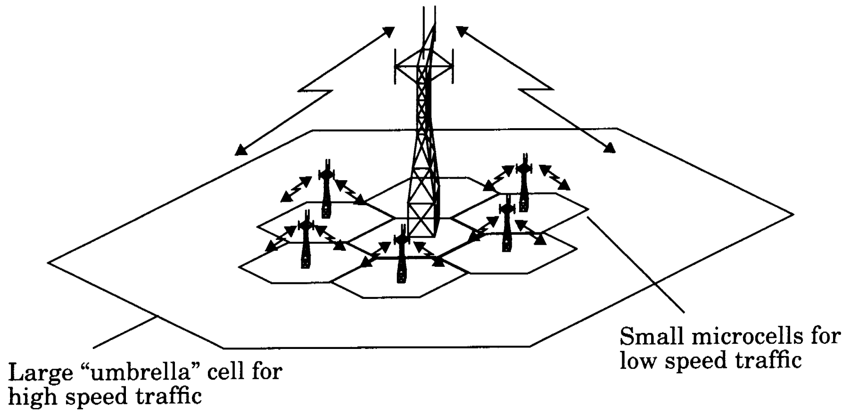

Figure 1.6: The umbrella cell approach

1.4 The cell coverage estimation problem

In recent years with the rapid growth and need for high quality and high capacity cellular networks estimating coverage accurately has become extremely important. Several groups within a network operator use information related to coverage. Those groups range from Engineering, Network Optimization, Customer Care, Marketing, etc.

The coverage map that can be created not only helps in deriving an overview of the mobile network but is necessary in many other ways as well. For example, such estimation can be helpful for the prediction of handover zones in the boundaries of a cell. On the other hand, as the demand for wireless communications is rapidly growing, the cell-splitting and sectoring techniques are applied in an already existing cellular network. In this case, a good estimation of each cell's coverage area is necessary to the radio network designer in order to proceed to the creation of new cells.

The theoretical RF propagation prediction algorithms that are used in cell planning, give only approximate coverage and are not suitable for detailed network design. To more accurately design the coverage of a specific cellular network signal strength measurements must be taken in the service area [2].

The difficulties in the actual determination of the cell coverage area are presented next.

1.5 Difficulties in determining the effective cell area

As shown before, especially in figure 1.4, the shape of the cells in a cellular mobile telephony system, are practically never perfect hexagons.

Additionally, a mobile station (mobile phone) that moves in the coverage area of a particular cell site may be served by an adjacent cell if the level of the signal strength it receives from the second one becomes greater than the one it receives from the first base station (handover). There are many causes that may lead to this situation most (if not all) of which stem from the wireless environment propagation characteristics.

Propagation mechanisms are very complex and diverse. First because of the separation between the receiver and the transmitter, attenuation occurs. In addition, the signal propagates by means of diffraction, scattering, reflection, transmission, refraction, etc.

Diffraction occurs when the direct line-of-sight (LOS) propagation between the transmitter and the receiver is obstructed by an opaque obstacle whose dimensions are considerably larger than the signal wavelength. The diffraction occurs at the obstacle edges where the radio waves are scattered and as a result, they are additionally attenuated. The diffraction mechanism allows the reception of radio signals when the LoS conditions are not satisfied (NloS case), whether in urban or in rural environments.

Scattering occurs when the propagation path contains the obstacles whose dimensions are comparable to the wavelength. The nature of this phenomenon is similar to the diffraction, except that the radio waves are scattered in a greater number of directions. Of all the mentioned effects, scattering is the most difficult to be predicted.

Reflection occurs when the radio wave impinges the obstacle whose dimensions are considerably larger than the wavelength of the incident wave. A reflected wave can either decrease or increase the signal level at the reception point. In cases where many reflected waves exist, the received signal level tends to be very unstable. This phenomenon is commonly referred to as multipath fading, and the signal is often Rayleigh distributed.

Transmission occurs when the radio wave encounters an obstacle that is to some extent transparent for the radio waves. This mechanism allows the reception of radio signals inside buildings in cases where the actual transmission locations are either outdoors or indoors.

Refraction is very important in macrocell radio system design. Due to an inconstant refractive index of the atmosphere, the radio waves do not propagate along a straight line, but rather along a curved one. Therefore, the coverage area of an actual transmitter is usually larger. However, as a result of the fluctuations of the atmosphere parameters, the received signal strength level is fluctuating as well.

Since there is frequently no LoS between the transmitter and the receiver, the received signal is a sum of components that often stem from several previously described phenomena. Therefore, the received signal level is quite variable with respect to time and especially with respect to the receiver and the transmitter displacement. Even a displacement of just a fraction of the wavelength can cause the signal level to change by more than 30db. These fluctuations are known as short-term (or multipath fading). On the other hand, the local average of the signal varies slowly with the displacement. These slow fluctuations depend mostly on environmental characteristics, and they are known as long-term fading.

From the above, it is clear that an accurate prediction of the field strength level, and thus the prediction of the exact cell shape and size, is a very complex and difficult task. To date, various field strength prediction methods have been proposed in the literature. Since the short-term fading of the received signal is almost impossible to predict, all propagation models estimate either the average or median values.

1.6 Evolution in Coverage Estimation Techniques

The primary objective of the first generation land to mobile systems was to provide coverage to the required area using a single tall tower. All the traffic was supported with a single base station. The land to mobile systems were available only in the large cities and the typical radius of coverage was about 40 miles. Same channels were reused in other cities since the co-channel interference was low as the cities were geographically apart. Since the land to mobile systems were coverage limited, stronger signal was better. For the land to mobile systems, the coverage estimation was mostly based on empirical propagation models. The empirical models are developed for different morphologies for some typical cities, and are limited to the ability of the engineer to classify the morphology and only provide an approximate estimate of coverage. Since only one base station was used, the prediction was used to determine the approximate boundary where the subscribers could use the service.

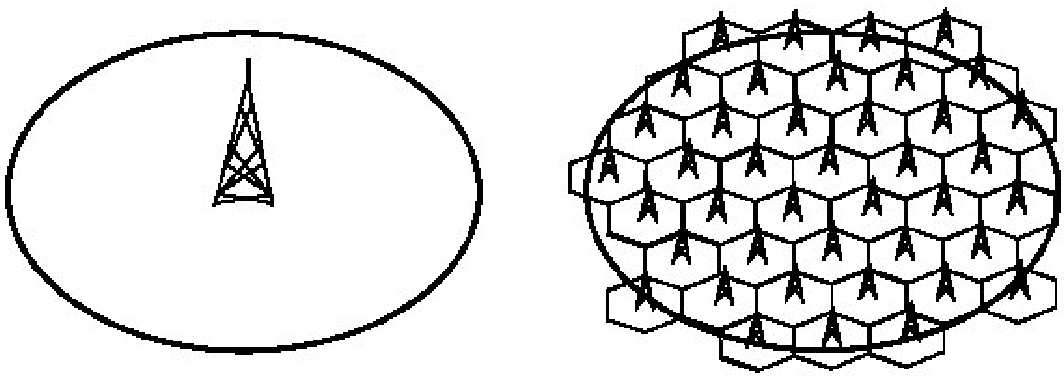

Figure 1.6: Illustration showing the importance of accurate coverage estimation in cellular networks as compared to early land to mobile systems.

The situation in modern cellular system is different. The modern cellular systems are being built to provide high quality of service and high capacity. The typical cell radius in urban areas is less than one kilometer. To minimize the interference (co-channel and adjacent channel interference) it is important to determine the coverage boundary accurately. Estimating inaccurate coverage has severe impact on the network performance. Over estimating coverage results in areas with signal strengths weaker than the minimum required threshold. Under estimating coverage will create coverage overlap, which can result in interference. Thus, accurate estimation of coverage is essential for a good design.

As explained earlier, empirical formulas do not estimate the coverage to the accuracy needed for designing a modern cellular network. Several computer based prediction tools model the physical phenomenon of RF propagation using terrain and clutter (land use) data. The accuracy of coverage estimation using these tools depends on the accuracy and resolution of the available data. Even when accurate and high resolution clutter data is available, the effect of the clutter on the propagation is different in different areas. For example, if the clutter data classifies an area as dense urban, and provides average building height, there is still an ambiguity about the density of the buildings in the area and also the propagation depends on the materials used to construct the buildings. Thus the propagation characteristics in an area classified as dense urban in one country can be very different from that in another country. Also the high resolution clutter data with heights, is extremely expensive to obtain as they have to be obtained by airal photography. Even though computer prediction tools may give better coverage estimates than empirical formulas, they alone are not good enough to be used to design a modern cellular network. The methods of coverage estimation without using any signal strength measurements from the service area are referred in the literature as untuned predictions. From the discussion it is clear that untuned predictions will not provide accurate estimate of coverage for designing modern cellular networks. The common practice now in designing modern cellular networks is to measure the signal strength for a test transmitter in the service area and to tune the propagation model using the measured data. Using this technique, coverage can be more accurately and reliably estimated.

In this thesis, the cell coverage estimation problem has been treated as a classification problem. Pattern classification is a very powerful means used in the classification of known situations and the prediction of the category (class) in which a new, 'a priori' unknown situation should be assigned at. On the other hand, neural networks can be trained to learn a classification task if appropriately selected parameters are applied to their inputs and outputs. Here, neural classifiers have been trained to be able to define the service area of each base station, in an urban environment. The neural classifiers are being supplied with features such as: the co-ordinates (latitude and longitude) of a location within the area of examination and the received signal strength from the serving BS at that particular location. The output layer consists of the number of cells, having each cell-BS represented as one decision class. The training was based on experimental measurements taken in the center of Patras city. Trajan software package has been selected as a tool for simulation. Trajan supports the most practical types of neural networks known for real-world problems-solving today, and includes the latest state-of-the-art techniques for fast training, automatic design and variable selection [5].

Chapter 2
Neural Networks

In this chapter neural networks fundamentals are shortly described, followed by a brief presentation of the main characteristics of the neural network architectures mostly used in cellular communication networks. Finally their main advantages and disadvantages are mentioned.

2.1 Basic Theory and Operation

Neural networks are systems composed of a large number of highly interconnected processing units. They are inspired by the physiology of biological nervous systems. A neural network can be considered as a system that does not have to be programmed- it only needs to be taught a problem area and it will then solve any similar problem.

Neural networks consist of units. Units are the basic building blocks of neural networks (sometimes called nodes or neurons). A unit is a simple processor that uses a function to compute its state of activation and its output from an array of input values it receives. Units can be binary (0,1 or –1,1) or analogue (continuous value range).

Neural networks are made up of units that are interconnected. Connections between units have weights which can be modified. Weights determine how strong a connection between two units is. Connection scan be excitatory (supporting activation) or inhibitory (suppressing activation). Units may be organised into layers (structured networks). A net can have one input layer, one output layer and any number of hidden layers. Hidden units are units that are not used for input or output of the network. They can be seen as holding the network's own dependencies between input and output patterns.

There are two types of network: feed-forward networks and feedback networks. In a feed-forward network there are no cycles i.e. no unit receives input from any unit that its output is connected to (directly or indirectly). In feedback networks cycles are allowed. This makes it possible for a network to have some kind of memory.

The overall behavior of the network depends on its structure and on the strength of connections.

Neural Networks have been used for classification or function approximation. If used for classification, these networks are trained by being presented a sufficient and representative amount of cases from all the different classes that appear in a problem. After they have learned how to assign certain patterns to the classes they belong, they are able to generalize and respond correctly for any unknown pattern that is presented at their input.

Operation

Neural networks are developed as follows: first the network is set up with all its units and connections and it is usually initialized with arbitrary weights. Then it is trained by presenting examples. During the training phase the weights on connections are changed which enables the network to learn. There are different learning algorithms that can be used to modify weights. When the net performs well on all training examples it should be tested on examples that it has not seen before. If the net can deal with these cases too, it is ready to be used.

On a more detailed level, neural networks operate by propagating activation states of units across the net. This is done by letting each unit re-compute its output. Units compute their output using the weights on the connections from their input units, and the activation states of those input units. Some units have a threshold which is used to decide whether the unit has received enough excitatory input to be activated.

An epoch is a presentation of the whole set of training examples to the net. Training times are usually measured in epochs. There are two major strategies for updating weights during training:
to collect the error information and update the weights after an epoch
to update the weights after each training pattern is presented.

Teaching the nets

Probably the most intriguing feature of neural networks is their ability to learn. There is a variety of different learning methods that can be used to train neural networks. There are two major learning techniques:

Supervised learning : here an evaluation of the output which the network produces is available for the training examples. Usually the trainer specifies both the input and the desired output of the learning network. Weights are changed to reduce the error which is the discrepancy between the desired output and the output the network produces itself. In multi-layer networks it is necessary to propagate the error backwards through the net so that the contribution of hidden units towards the overall error is taken into account. A common technique is backpropagation. *Supervised learning*: is used in cases where there is sufficient knowledge about the data that the network will encounter. It must be known what response the network should give for each input pattern, which means that a solution to the problem must already be known.

***Unsupervised learning*:** it is usually competitive, the unit with the biggest total input wins and is activated, all other competing units are de-activated. The winning unit updates its incoming weights so that connections from active units are made stronger and connections from inactive units are made weaker, so that the weight vector is made more similar to the input vector. Only the winning unit learns, the rest of the network remains unchanged. *Unsupervised learning* divides a set of input patterns into disjunct classes. Each cluster contains patterns with similar features. Therefore these networks can be used to detect regularities in input patterns. They can determine which features are relevant for classification, reduce dimensionality and enhance contrasts.

2.2 Neural Network architectures and training algorithms

The main elements of a NN are:

- A set of processing units (neurons).
- An activation state y_j for neuron j , which is the actual output of the neuron.
- Connections among the neurons. In general every connection (link) is defined by a weight w_{ji} which determines the impact of the signal of neuron i to the neuron j.
- A propagation rule which determines the actual input u_j of a neuron from the outer inputs.
- An activation function $\phi_j(\cdot)$ which determines the new activation level based on the actual input and the running activation state.
- An outside bias input b_j for unit j.
- A training method.
- An environment it has to work in and which offers the input signals and the error signals (if needed).

Perceptrons

Perceptrons are the simplest network architecture. There are no hidden units and only one output unit which is connected to all input units. Connections are excitatory and have modifiable weights. Input units have continuous state values whereas the state of the output unit is binary (1,0 or –1,1). The output unit usually has a hard delimiter nonlinearity and a threshold. The input units are used solely to present the input pattern to the net.

The perceptron is built around a nonlinear neuron, the McCulloch-Pitts model of a neuron. Such a model consists of a linear combiner followed by a hard limiter which performs the signum function. The summing node of the neuronal model computes a linear combination of the inputs applied to its units, and also incorporates an externally applied bias. The resulting sum, that is the induced local field is applied to a hard limiter. The neuron produces an output equal to +1 if the hard limiter input is positive and –1 if it's negative.

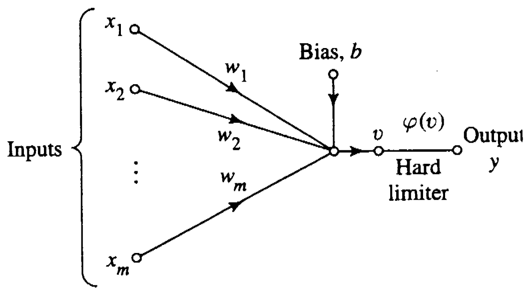

Figure 2.1: Signal-flow graph of the perceptron

The most important feature of the MCP model is that the inputs to the neuron are weighted, that is they are multiplied by a number (called weight). This way, the effect of each input to the decision rule depends on its weight. In the figure, $w_1, w_2,...w_m$ are the weights of the perceptron, while $x_1, x_2,...x_m$ are the inputs applied to it. The external bias is denoted by b. The hard limiter input or induced local field is :

$$u = \sum_{i=1}^{m} w_i x_i + b \qquad (2,1)$$

The goal of the perceptron is to correctly classify the input set to class C_1 if it's output y is +1 and to class C_2 if y is –1.

If a map of the decision regions in the m-dimensional space spanned by the m input variables $x_1, x_2,...x_m$ is plotted, then the two decision regions are separated by a hyperplane defined by :

$$\sum_{i=1}^{m} w_i x_i + b = 0 \qquad (2,2)$$

For the perceptron to function properly the two classes must be linearly separable. This means that the patterns to be classified must be sufficiently separated from each other to ensure that the decision surface consists of a hyperplane. Besides that, the perceptron built around a single neuron can perform pattern classification only with two classes. By expanding the output layer to include more than one neuron we may form classification with more than two classes.

Initialized with arbitrary weights, perceptrons are trained by supervised learning: an input pattern is presented to the net, the output is calculated and compared to the desired output. The weights are changed according to the output error. Then the next training example is learned. This procedure is repeated until the weights converge and do not need to be changed any more. After that the net can be used for unknown input patterns for which it will generate the appropriate output. The output is calculated by computing the weighted sum of the inputs, subtracting the unit's threshold and applying the hard delimiter nonlinearity. The weights are updated using a very simple algorithm: if the calculated output is correct no weights are changed. Otherwise the weights of connections to active input units are changed to support the desired response. If the output unit should be active the weights are increased, if it should be inactive the weights are decreased.

Multi-layer perceptrons

Multilayer perceptrons are feed-forward nets like perceptrons but with one or more hidden layers. To give the net a wider range of application, multiple ouput units are used together with sigmoidal nonlinearities. The output units can now have continuous states which leads to smooth decision regions. Learning in multi-layer perceptrons can be supervised or self-supervised.

They are more powerful than perceptrons, but also more difficult to train. They are trained using supervised learning with the help of backpropagation. During training every trial consists of two phases: a forward sweep for propagation of activation and a backward sweep for error propagation by changing weights. The training patterns (together with the desired outputs) are presented repeatedly until the weights converge and the error of the net is reduced to an acceptable level.

Multilayer Perceptrons (MLPs) constitute an important and very popular class of neural networks. They represent a generalization of the single-layer perceptron. Typically the network consists of a set of sensory units that constitute the input layer, one or more hidden layers of computation nodes and an output layer of computation nodes. The input pattern propagates through the network in a forward direction, on a layer-by-layer basis.

An MLP has three distinct characteristics:

1. Each neuron in the network includes a nonlinear activation function. The nonlinearity is smooth (i.e. differentiable everywhere), opposed to the hard-limiter of the single-layer perceptron. Usually, the sigmoidal nonlinearity is used which is defined by the logistic (sigmoid) function:

$$y_i = \frac{1}{1+\exp(-u_i)}, \qquad (2,3)$$

where u_i is the induced local field (the weighted sum of all inputs plus the bias) of neuron j, and y_j is the output of the neuron.

2. The network contains one or more hidden layers of hidden neurons that are not part of the input or output of the network. These enable the network to learn complex tasks by extracting progressively more meaningful features from the input patterns.
3. The network exhibits a high degrees of connectivity, determined by its weights. A change in the connectivity requires a change in the weights.

Through the combination of these characteristics plus the ability to learn from experience through training, make MLP networks very powerful.

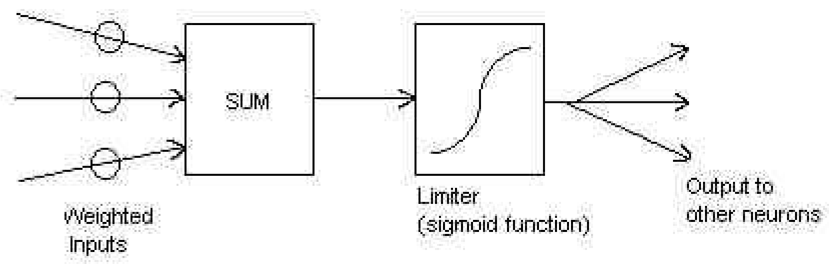

Figure 2.2: One neuron of the MLP network

The Back Propagation algorithm provides a computationally efficient method for the training of multilayer perceptrons.

Back Propagation

The backpropagation algorithm is a generalization of the least mean square algorithm (LMS). It uses gradient descent to minimize the cost function of the net which is the mean square difference between the desired and the actual output.

For output units the error depends on the difference between the actual and the desired output. For hidden units the error is computed using the unit's own activation and the weighted error of all its output units in the layer above. Thus the error is propagated back across the layers of the net and the weights are changed accordingly. The network is initialized with small random weights and the input is presented. When the network has produced its output this is compared to the desired output. Starting with the output units the error of each unit is computed and the weights of the unit's incoming connections are changed according to the error.

The back propagation algorithm works by iteratively training the network using the training data available. On each iteration known as epoch, the entire training set is presented to the network one case at a time. The inputs are presented to the network which is executed to produce output values. The output values are compared with the desired outputs present in the data set, and the error between the desired and actual outcome is used to adjust the weights in the network so that the error is likely to be lower. The algorithm must compromise between the various cases, attempting to alter the weights so that the overall error across the whole training set is reduced.

Radial Basis Function Neural Networks (RBF-NN)

The Radial Basis Function Neural Network is comprised of two layers the hidden layer and the output layer. The hidden layer is composed of an array of hidden neurons each of which contains a parameter vector called a center. The neurons compute the distance between the center and the network input vector. The squared distance is subsequently divided by a parameter called the width which is the spread of the corresponding center, and the result is passed through a nonlinear activation function. The output layer of the network is a linear combiner with a set of connection weights. The output response of an RBF network is a mapping fr:

$$f_r(y) = \sum_{i-1}^{n} w_i \phi\left(\frac{\|y - c_i\|^2}{p_i}\right) \quad , \qquad (2,4)$$

where n is the number of computing nodes, the c_i are the RBF centers, the p_i are the center spread parameters, $\phi(.)$ is the basis function and w_i are the weights.

As in MLP model the space is divided in hyperplanes , in RBF networks the space is divided in circles (hyperspheres). A hypersphere is characterized by its center and radius. In an RBF units respond to the distance of points from the "center" represented by the radial unit.

In contrast to the MLP network, in RBF each unit measures the square of the distance of the input vector from the weight vector. This distance is then multiplied by the threshold (a deviation) before being passed through the activation function. Thus radial basis function networks work by dividing up pattern space using hyperspheres. An RBF always has three layers the input layer, the hidden layer which contains radial units and a linear output layer. The idea is to pick centers which lie at the heart of clusters of the training data with deviations selected to reflect the density of the data.

Hopfield Nets

Hopfield Nets are unstructured nets. Their units are not organized into layers. The connections between units are symmetric: if there is a connection from unit a to unit b then there is also a connection from unit b to unit a and the two connections have the same weights. Connections between units can be excitatory or inhibitory. Units usually have discrete states 1 and 0.

These nets are trained by self-supervised learning. The weights are set so that each example pattern is represented by a stable state. This can be done by calculating the weights on all connections using the set of examples to be learned. Given a new input pattern the net will then settle into the (previously learned) stable state closest to the given input pattern. An input pattern is presented to the net by setting the states of the net's units (active/inactive). In a Hopfield net the two stages of training and retrieval are distinct. They are not adaptive nets. The learning phase is always terminated before the retrieval phase begins. The net changes its overall state by asynchronous iteration, updating the states of its units by computing their activation function, which is simply the weighted sum of the incoming connections. With each iteration the states of units are changed, which influences other units' states at the next iteration. Through these iterations the net converges, i.e. it reaches an overall stable state where the states of units stop changing. During training the weights are assigned to connections according to the elements of the training samples. If two units are often active simultaneously the connections between them are strongly excitatory; if mostly one unit is active while the other is not then the connections are strongly inhibitory; and if two units are mostly independent the connections are weak.

Kohonen's self organizing maps

Kohonen's self organizing feature maps are associative networks. The nets are organized in a two-dimensional array. Units are connected so that there are excitatory connections between nearby units and inhibitory connections between distant units.
Initially all units respond randomly to the input parameter. The network is trained with an adapted form of unsupervised learning. An input pattern is presented to the net and the unit with the highest response to that pattern is located and its weights are updated. Additionaly the unit's neighbours are identified, i.e. those units that are located in some region around the winning unit. These neighbouring units also have their weights updated. This way the formation of neighbourhouds is encouraged so that nearby units respond similarly to inputs.

This architecture and learning process results in fast and reliable organization of networks. Usually the system will start organizing into large regions and focusing into smaller regions during training.

Recurrent Neural Networks

Recurrent Neural Networks are highly nonlinear and excibit a rich and complex behavior. They are the most general case of neural networks since every neuron is connected to every other neuron. The network incorporates feedback and as a result its architecture becomes inherently dynamic. In general, an RNN has m external inputs and n fully interconnected neurons. The output of a neuron at time t+1 depends not only on the external inputs to the network $x_l^{net}[t], l = 1,...,m$ at the previous time instant but also on the previous outputs of the neurons $y_l[t], l = 1,...,n$.

This dynamical behavior is described by the following equations:

$$s_k[t+1] = \sum_{l-1}^{n} w_{kl}[t] y_l[t] + \sum_{l-1}^{m} w_{k,l+n}[t] x_l^{net}[t] \quad (2,5)$$

$$y_k[t+1] = \phi(s_k[t+1]) \quad (2,6)$$

where the w_{ij} are the weights of the connection from the jth neuron to the ith neuron at time t. The activation function $\phi(.)$, may be any real differentiable function with respect to its argument. Usually the hyperbolic tangent function is used as activation function. As can be seen from the equations every output depends on all the previous outputs of the network. The most widely known training algorithm for RNN is the Real Time Recurrent Learning algorithm. The RTRL algorithm is based on the minimization of the MSE by a gradient descent procedure and is used to update the weights of the RNN during the training period.

2.3 Advantages and disadvantages

A short reference to the general advantages and disadvantages of neural networks is following:

Advantages of NNs

- generality : They are learning systems and which problem they solve depends on what they are taught. They can be applied to a variety of applications since mostly they are problem independent. They are easy to implement in hardware and software
- adaptivity: NNs learn to produce a response for a given input. When the network is confronted with new inputs from the environment, it can learn to produce the right responses for those inputs. This makes NNs autonomous systems that can perform well in a complex changing environment without being reprogrammed
- tolerance
- feature selection
- self programming

Disadvantages of NNs

- lack of theoretical foundations
- selection and representation of inputs
- network architecture
- representative training examples
- selection of an appropriate training algorithm
- selection of learning parameters
- interpretation of results
- interpretation of operation
- confidence in solutions
- comparative assessment of performance
- scaling

Chapter 3
Application of Neural Networks in Mobile Communication Systems

In the last years, the application of neural networks to mobile communication networks has been the focus of several studies and many simulations have given interesting results. In this chapter, some of the most popular applications of neural networks in certain areas of mobile communications like:

- signal detection
- adaptive equalization for digital mobile radio channels
- Adaptive antenna array beamforming. A special reference to the direction of arrival problem and the neural network solution is included
- radio resource management
- channel assignment in mobile radio systems
- fraud detection in mobile communication networks
- path loss prediction in cellular networks (field strength prediction)
- mobile station positioning.
- cell planning

are presented.

3.1 Neural Networks in signal detection

Multipath propagation is experienced in cellular mobile communications as well as in many types of communication systems. A multipath transmission is generally experienced if there are two or more paths from transmitter to receiver. This can be due to atmospheric conditions or to obstacles near the surface of the earth or in underwater environments. When considering mobile cellular radio transmissions a signal can be reflected from several intervening surfaces such as buildings and terrain. Consequently, several copies of the signal may arrive at the receiver at various levels of intensity and time delay. These signals arrive constructively and destructively, sometimes reducing the signal-to-noise ratio to unacceptable levels. Severe multipath propagation usually results in a significant decrease in signal quality and intelligibility, thereby reducing throughput. In order to exploit the full system capacity of code division multiple access (CDMA) systems, multiuser detection (MUD) is applied. Various linear and nonlinear multiuser detectors (MUD) have been proposed. The major disadvantage of MUD, especially of optimum MUD, is the increased complexity, compared to single user detection (SUD).

A popular method employed today to determine the nature of the transmitted signal is the RAKE receiver.

In [6] it is stated that in order to achieve a better Eb/N0 (bit energy to noise ratio) than Rake receiver, a better estimation of the radio channel is needed. Artificial neural networks are proposed to reach this goal, which are used for detecting all

multipaths in a direct sequence spread spectrum signal. The output of the adaptive neural net is an estimation of the channel response.
As it is reported in [6], to increase the Eb/N0 of direct sequence spread spectrum signals, Rake receivers consist of up to 6 demodulators, which demodulate the strongest few echoes of the signal. Their output signals are collected. The result is a higher stability against noise. The presented work is also based on using not only one path for reconstructing the transmitted bit stream but nearly all paths.

The received PN (pseudo noise) radio signal is sampled with a rate of about four times the chip rate of the PN signal and passed through a despreader that correlates it with the original PN sequence. In contrast to a conventional receiver, the correlation is made for every delay, realized in a correlator bank whose output is given to a neural network. As neural net, a perceptron with backpropagation is used, with the task to detect all the multipaths in the correlated signal. With conventional filters the problem of detecting only the multipath peaks and not the peaks due to crosscorrelation of a PN sequence with an inverted PN sequence - occuring whenever a bit change from 1 to -1 or inverse - is nearly impossible to handle. The output of the net is the estimated response of the actual channel. With this information containing the delay and the attenuation of each propagation path, the transmitted bit stream can easily be rebuilt.

The authors report that to get the best possible neural net filter in every environment the training of the net must continuously going on. For this reason a self learning system follows the output of the described system in the last paragraph. The estimated channel response, given by the neural net, is convolved with the same PN sequence used for dataspreading. This signal should be equal to the input of a received bit, if the channel estimation is exact. After being correlated this signal is fed to a neural net equal to that used for detecting the multipaths of the incoming data. The output of the net should be the same as the one of the first net. If it is, the filter is optimal adjusted for the given propagation medium. If not, the difference of the output of the first and second net is an error signal. This error signal is back-propagated through the second net and the weights are updated (continuous weight update). These new weights are copied into the first net. The first net uses the new weights for the next incoming data, which leads to a closer approximation of the real channel.

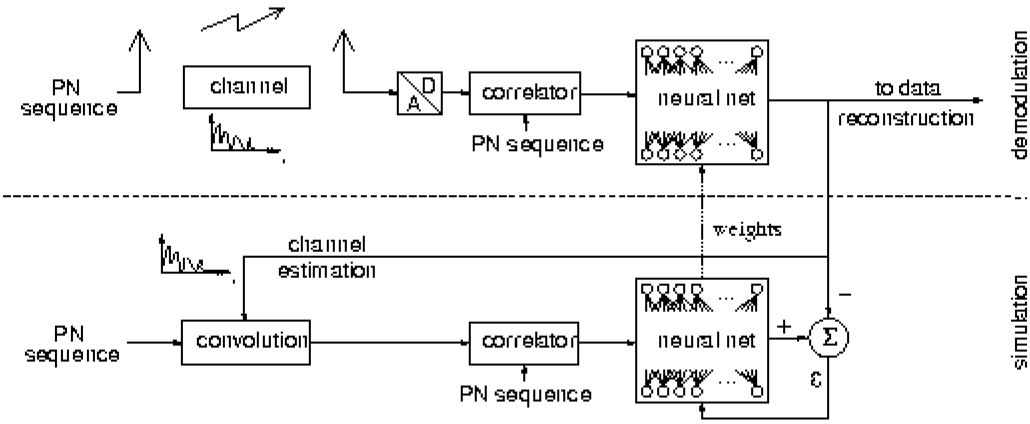

Figure 3.1: Block diagram of the neural net direct sequence spread spectrum receiver, without HF components

In [7], it is also reported that the well-known RAKE Receiver a method employed today to determine the nature of the transmitted signal can be replaced by an appropriately trained neural network.

In [8], the concept of multiuser/multisubchannel detection based on recurrent neural network structures is described. The basic principle of MUD based on RNN structures is an iterated nonlinear feedback of tentative decisions (soft feedback). The nonlinear decision function has been derived for linear modulation schemes with general complex-valued symbol alphabet of the RNN detector. Generally, the algorithm shows an excellent performance close to the statistically optimum MUD, while its complexity remains moderate. Therefore the algorithm is thought to be well suited for many applications, including 3rd generation mobile communication systems.

In [9] the authors present a new multiuser receiver using a neural network-based decision scheme for interference suppression in Direct Sequence Code Division Multiple Access (DS-CDMA) wireless networks. This receiver is mainly made of a decision feedback functional link equalizer (DFFLE), which is the main component of the receiver, combined with an eigenvector network. This structure well approximates a Bayesian receiver and exhibits several advantages when compared to the classical Minimum Mean Square Error (MMSE) receivers. It is also demonstrated that the performance of a multilayer perceptron in this context is comparable to that of the optimum receiver

In [10] linear and non-linear adaptive algorithms are investigated for Space Division Multiple Access (SDMA). SDMA is one of the emerging techniques for multiple access of users in mobile radio, which uses spatial distribution of users for their differentiation. The performance of the linear Square Root Kalman (SRK) algorithm for SDMA is compared to that of the non-linear Recurrent Neural Network (RNN) technique. The proposed SDMA-RNN technique is evaluated over Rician fading channels, and it shows improved Bit Error Rate (BER) performance in comparison with the linear SRK-based technique. The performance of SDMA-RNN is also compared with that of Code Division Multiple Access (CDMA) systems, showing that it could be used as a viable alternative scheme for multiple access of users. Finally, a Hybrid CDMA-SDMA system is proposed combining CDMA and SDMA-RNN systems. Hybrid CDMA-SDMA exhibits a very good potential for increase in the capacity and the performance of mobile communications systems.

3.2 Neural Networks in adaptive equalization for digital mobile radio channels

The worldwide growth of wireless mobile telecommunication services requires the transmission of more and more data at high rates over long distances. This involves the use of non-linear amplifiers to improve the transmission channel efficiency. However, such devices cause severe distortions for the transmitted signal. Channel equalization plays major role in extracting true data from the noisy transmitted data corrupted with intersymbol interference (ISI) and other channel distortions. ISI is one of the main problems arising when a signal is transmitted through a communication channel, due to the fact that signals arrive with different delays due to the presence of different propagation paths.

As stated in [11] equalization refers to any signal processing techniques used at the receiver to combat ISI. In order for an equalizer to mitigate ISI, an estimate of the channel response is needed at the receiver.

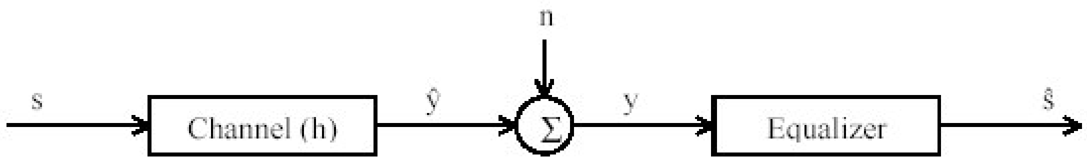

Figure 3.1: Basic communication channel/equalizer model

A simple schematic of the relative position of an equalizer in a data transmission system is provided in the above figure. The input to the channel is s, and the channel output y is corrupted by noise n, which is generally modeled as an additive white Gaussian noise process. The channel is affected by both linear and nonlinear distortions: ISI is mainly responsible for linear distortions while nonlinear distortions are introduced through amplifiers, converters, the propagation environment, etc. If the channel, h, is modeled as linear, the uncorrupted channel output y is simply the convolution of s and h: $y = s * h$.

Conversely, if the channel is modeled as nonlinear, the channel output y cannot be represented as a simple linear convolution.

Conventional methods use linear channel equalization schemes which employ a linear filter with a finite impulse response (FIR) or lattice structure, and non-linear methods like Decision Feedback Equalization (DFE) and Maximum Likelihood (ML) sequence detection schemes. However, linear equalizers do not perform well on channels with deep spectral nulls since the equalizer places a large gain at these frequencies and consequently significantly enhances the additive noise [11]. Noise enhancement can be overcome by using nonlinear equalization techniques such as decision feedback equalization (DFE) or maximum likelihood sequence estimation (MLSE). The difference between linear and nonlinear equalizers is determined by whether the reconstructed symbol at the receiver output is employed in a feed back path to adapt the equalizer for subsequent outputs: linear equalizers do not utilize a feed back path while nonlinear equalizers do. In both linear and nonlinear equalization techniques, however, the channel model is generally assumed to be linear. If the channel suffers from significant nonlinear distortions, the aforementioned equalization techniques exhibit poor performance, and equalization techniques that better combat nonlinear channels are desirable.

Over the last years neural network equalizers have raised much interest. Their non-linear structure and good learning properties make them good candidates to solve the equalization of linear as well as non-linear channels problem.

Equalization as a Classification Problem

The adaptive equalization problem is typically viewed as an inverse filter problem.

Traditional equalizers are designed to approximately track and invert time-varying channel distortions by adjusting filter coefficients while maintaining a prescribed signal to noise ratio (SNR). Tradeoffs between noise enhancement and channel inversion generally render these techniques suboptimal. The output of the equalizer is fed into a decision device which attempts to estimate the transmitted symbol. This configuration of an inverse filter equalizer followed by a decision device results in the partitioning of the output signal space by linear decision boundaries between different symbols.

Equalization can be considered a geometric classification problem rather than an inverse filter problem, where the main objective becomes the separation of the received symbols in the output signal space whose optimal decision region boundaries are generally highly nonlinear. With this viewpoint to equalization, complete channel inversion is unnecessary, and the problem is tackled using classification techniques. Since neural networks are well known for their ability of performing classification tasks by forming complex nonlinear decision boundaries, neural equalizers have been recently receiving considerable attention.

In [11] a review on the use of three neural network structures—the multilayer perceptron, the radial basis function network, and the recurrent neural network in the equalization problem as well as examples of the performance of proposed equalizers using these structures, are presented.

The several delayed versions of the channel output are considered as inputs to the MLP network while the "information" of the channel is stored in the weights, or coefficients, of the neurons, and these weights must be periodically updated to track the time varying channel. The weights are generally updated by using the back-propagation (BP) training algorithm. Since the BP algorithm is a supervised learning algorithm, a set of actual output/desired output pairs train the network to implement the desired mapping. The BP algorithm is iterative and adjusts the weights so as to minimize any differentiable cost function such as the mean square error (MSE). Since the BP algorithm performs a gradient descent to reach a minimum, the search direction is given by the steepest gradient at the current search point, and the weights are adjusted by an amount proportional to local gradients.

When using RBF network the training algorithm is comprised of an unsupervised learning rule for the centers and a supervised learning rule for the weights. The centers are generally updated using the *k*-means clustering algorithm which consists of computing the squared distance between the input vector and the centers, choosing a minimum squared distance, and moving the corresponding center closer to the input vector. In addition, it is also necessary to update the weights of the network. Supervised learning for the weights using the LMS algorithm is generally employed for this purpose. Comparing the RBF network response with the optimal Bayesian equalizer solution shows that the two have identical structures. The channel order, which determines the number of computing neurons, is usually obtained by autocorrelation techniques.

RNN's model nonlinear filters and can accurately realize the inverse of finite memory channels using relatively small numbers of neurons. As shown in[11], small size (i.e. computationally efficient) RNN equalizers outperform traditional equalizers as well as other neural network based equalizers such as the MLP. The small size of RNN equalizers makes them attractive for high speed channel equalization when compared to the complexity associated with other neural equalizer structures.

A different neural network based equalization algorithm is presented in [12] called MRAN (Minimal Resource Allocation Network). That is a minimal radial basis function neural network structure, algorithm that uses on-line learning and has the capability to grow and prune the RBF network's hidden neurons. The network begins with no hidden units. As each input-output training data is received and processed the network is built up based on certain criteria. The algorithm adds hidden units as well as adjusts the existing network according to the data received. When an input to the network does not meet the criteria for a new hidden unit to be added the network parameters are adapted using the Extended Kalman Filter. Results show the superior performance of the MRAN algorithm for two different non-linear channel equalization problems.

The algorithm's performance was evaluated by using it to build up an equalization network for two non-linear channels, along with one non-minimum phase channel. The resulting networks were then tested by comparing their bit error rate (BER) performance to that of the ideal Bayesian equaliser. The results showed that the networks obtained, are comparable in performance to ideal equalizers when suitable training parameters are selected.

In [13], a differentiation between multilayer multi-neuron and multilayer single neuron equalizers takes place. A highly efficient and novel multilayer single neuron neural equalizer structure is developed. It is shown that the performance is quite comparable to that of multilayer multi-neuron network. To adapt the weights of both the neural equalizers, the Back Propagation Algorithm was used. In the simulation, a specially selected channel with certain transfer function is excited by white pseudo-random binary signal and then a white zero-mean gaussian noise is added to the channel output. This distorted and corrupted signal is then passed through the equalizer. The equalizer is a tapped delay filter with 4 feedforward taps and one feedback tap. The output of the equalizer is then compared to the desired signal derived by delaying the input signal by 3 samples. The resulting error is used to update the tap weights and the recursions continue till a minimum mean squared error condition is satisfied. Both neural equalizers have been trained with backpropagation algorithm.

The simulation has been carried out on various communication channels to validate its efficacy. The results clearly indicated that even for highly distorted channels (having deep spectral nulls) the proposed equalizer performs satisfactorily.

A combination of the RBF and Self Organising Map (SOM) neural network based equalizers is proposed in [14], by replacing the RBF k-mean algorithm with a Kohonen algorithm. The RBF equalizer estimates the probability density function of the incoming signal in order to approximate the optimal Bayesian equalizer. The RBF network consists of two layers. Each neuron on the hidden layer computes the Euclidean distance between its center and the input vector and passes the result through a nonlinear function. The centers of the neuron converge to the channel states which are the possible outputs of the noiseless channel. As the number of channel states increases many neurons may be forgotten by the k-mean algorithm. In [14] it is suggested that the neurons of the RBF can be regarded as a SOM and the neurons update equations can follow the Kohonen rule.

In [15], a comparison between three different first-order recurrent neural networks (fully recurrent, partially recurrent and Elman), trained using the real time recurrent learning algorithm and the GSM training sequence ratio (26/114) for digital equalization of 2-ary PAM signals, is presented. All the architectures are single input single output and have N hidden units. The equations describing these architectures are:

Fully Recurrent NN:

$$x[t] = F_{N,N+1}(x[t-1], u[t]);$$
$$y[t] = x_1[t]; \qquad (3,1)$$

Partially recurrent NN:

$$x[t] = F_{N,N+1}(x[t-1], u[t]);$$

$$y[t] = F_{1,N+1}(x[t-1], u[t]); \quad (3,2)$$

Elman NN:

$$x[t] = F_{N,N+1}(x[t-1], u[t]);$$
$$y[t] = F_{1,N}(x[t]); \quad (3,3)$$

where $F_{i,j}$ stands for a single-layer perceptron having i outputs and j inputs, and ij weights and i biases, $u[t]$ is the input, $x_i[t]$ is the state of the ith hidden unit, y[t] is the network output and $d[t]$ the desired output at time t, respectively. The activation function of all units is the hyperbolic tangent.

The real-time recurrent learning algorithm, which is used for the training of these NNs updates weights every time a target or desired output is supplied.

The authors state that recurrent neural networks are in some respects very similar to the Decision Feedback equalizer (DFE) in that outputs are fed back to the classifier to assist in subsequent decisions.

3.3 Adaptive antenna array beamforming and Direction of arrival estimation using neural networks

An application of antenna arrays has been suggested in recent years for mobile communication systems to overcome the problem of limited channel bandwidth, thereby satisfying an ever growing demand for a large number of mobiles on communication channels. It has been shown by many studies that when an array is appropriately used in a mobile communications system, it helps in improving the system performance by increasing channel capacity and spectrum efficiency, extending range coverage, tailoring beam shape, steering multiple beams to track many mobiles, and compensating aperture distortion electronically. It also reduces multipath fading, cochannel interferences, system complexity and cost, BER, and outage probability.

An array of antennas may be used in a variety of ways to improve the performance of a communications system. Perhaps most important is its capability to cancel cochannel interferences. An array works on the premise that the desired signal and unwanted cochannel interferences arrive from different directions. The beam pattern of the array is adjusted to suit the requirements by combining signals from different antennas with appropriate weighting. The scheme needs to differentiate the desired signal from the cochannel interferences and normally requires either the knowledge of a reference signal, a training signal, or the direction of the desired signal source to achieve its desired objectives. There exists a range of schemes to estimate the direction of sources with conflicting demands of accuracy and processing power. Similarly, there are many methods and algorithms to update the array weights, each with its speed of convergence and required processing time. Algorithms also exist that exploit properties of signals to eliminate the need of training signals in some circumstances.

A phased array antenna uses an array of simple antennas, such as omnidirectional antennas, and combines the signal induced on these antennas to form the array output. Each antenna forming the array is known as an element of the array. The direction where the maximum gain would appear is controlled by adjusting the phase (the weights) between the different antennas. The phases of signals induced on various elements are adjusted such that the signals due to a source in the direction where maximum gain is required are added in phase. This results in the gain of the array (or equivalently, the gain of the combined antenna) are equal to the sum of the gains of all individual antennas. The term *adaptive antenna* is used for the phased array when the gain and the phase of the signals induced on various elements are changed before combining to adjust the gain of the array in a dynamic fashion, as required by the system. In a way, the array adapts to the situation, and the adaption process is normally under the control of the system. An optimal antenna is one in which the gain and phase of each antenna element is adjusted to achieve the optimal performance of the array in some sense. For example, to obtain maximum output SNR by canceling unwanted interferences and receiving the desired signal without distortion may be one way of adjusting gains and phases of each element. This arrangement where the gain and phase of each antenna element is adjusted to obtain maximum output SNR sometimes also referred to as signal-to-interference-and noise ratio, SINR) is also referred to as optimal combining in the mobile communications literature.

The signals induced on different elements of an array are combined to form a single output of the array. A plot of the array response as a function of angle is normally referred to as the array pattern or *beam pattern* [17,18].
This process of combining the signals from different elements is known as *beam forming*. Adjusting only the phase of signals from different elements to point a beam in a desired direction is the conventional method of beam pointing or beam forming.

Direction Of Arrival

Locating and tracking the mobiles at the base station is an important task in order to adapt the system parameters to meet the traffic requirements. In mobile communications, scattering in the vicinity of a mobile causes a spreading of the source and the signal arrives at the base station in the form of multipath, as if there were many radiating sources of varied power in the neighborhood of the mobile. The Direction Of Arrival problem deals with locating the mobile station by processing of the signals the antenna array at the base station receives from this specific mobile station.

Neural Network approach to beamforming and DOA problem

Currently, superresolution algorithms such as the multiple signal classification MUSIC algorithm and ESPRIT have been used to perform the direction finding or angle of arrival of signals from mobile users. One drawback of these algorithms is the difficulty of implementing them in real time because of their intensive computational complexity. Neural networks on the other hand, due to their high-speed computational capability, can yield results in real time, as it is reported in [18]. This leads to the accurate estimation of the mobile location in a few characteristic time constants of the circuit, normally, on the order of 100s of nanoseconds. This will enable the system to estimate the directions of multiple users even as the mobile users move. In addition, neural networks can yield fast convergence rates for the adaptive beamforming mechanism since the weights of the adaptive array antennas can now be computed in real time.

Moreover, conventional beamformers require highly calibrated antennas with identical element properties. Performance degradation often occurs due to the fact that these algorithms poorly adapt to element failure or other sources of errors. Neural Network-based array antennas do not suffer from this shortcoming. The antenna behavior can be incorporated in the training of the neural network under different circumstances and scenarios. The network being able to generalize can then be used to predict the aperture behavior at all points.

Neural Networks in Adaptive beamforming

As stated in [18], an antenna array consists of sensors separated in space whose output is fed into a weighting or beamforming network.
In an M-element linear array the elements' output of the array has the form of the M-dimensional vector

$$X = [x_1 \quad x_2 \quad x_M]^T \qquad (3,4)$$

and the weights of the element outputs can be represented in the M-dimensional vector

$$W = [w_1 \quad w_2 \quad w_M]^T \qquad (3,5)$$

These weights are the excitations required to feed the array elements to perform the appropriate beam steering. The array output is

$$y(t) = \sum_{i=1}^{M} w_i^* x_i(t) = W^H X(t) \qquad (3,6)$$

where * denotes the conjugate. To derive the optimal weight vector the array output is minimized so that the desired signals are passed with specific gain while minimizing the contributions due to noise and interference. Neural networks can be trained to determine the appropriate weights.

The whole procedure can be shortly described in two steps:

- Generation of the array output vectors
- normalization of each one of these vectors by its norm.
- evaluation of the correlation matrix for each of the array output vectors.
- Calculation of the corresponding weight vectors.
- Presentation of the normalized array output vector at the input layer of the RBF-NN and of the weight vectors at the output layer of the neural network.

The RBF-NN will thus be trained to produce the optimum weights for any antenna array output vector presented at its input layer.

Unlike the LMS, RLS and SMI algorithms, where the optimization is carried out whenever the directions of the desired or interfering signals change, in this approach the weights of the trained vector can be used to produce the optimum weights needed to steer the narrow beams of the adaptive array to the direction of desired users. Knowing that the response time for neural networks is very small, the proposed adaptive beamforming technique will be able to track mobile users as they move.

It is shown that the RBF-NN produced a solution for the beamforming weight vector that is very close to the optimum solution. By further increasing the size of the array it was found that the network was still able to direct beams in the direction of the desired users while nulling the interference.

Neural Networks in the DOA problem

Algorithms such as MUSIC, the minimum variance distortionless response (MVDR) and the minimum norm (MinNorm), have been successfully applied to the problem of direction of arrival estimation to locate and track radiating sources with additive noise, uncorrelated and correlated signals. The superior resolution capability of these techniques rely on powerful computational approaches involving eigendecomposition of the correlation matrix of the sampled data measured by an antenna array and then solving a projection optimisation problem or performing a search for the roots of some polynomial. As a result they are difficult to solve in real time. On the other hand neural beamformers have the advantage of fast convergence rates, and they can easily determine the angle of arrival of a source or target and allow the antenna to track that target in real time.

The Direction Of Arrival problem is approached as a mapping that can be modelled using a suitable neural network trained with input-output pairs.

Using complex signal representation, the received signal at the ith element of an array is :

$$x_i(t) = \sum_{i=1}^{K} s_m(t) e^{-j(i-1)k_m} + n_i(t) \ , \quad i = 1,2,...,M \qquad (3,7)$$

where $s_m(t)$ is the signal of the mth wave, $n_i(t)$ is the noise signal received at ith sensor and

$$k_m = \frac{\omega_o d}{c} \sin(\theta_m) \qquad (3,8)$$

where d is the spacing between elements and c is the speed of light in free space.

Using vector notation:

$$X(t) = AS(t) + N(t) \qquad (3,9)$$

where

$$X(t) = [x_1(t) \quad x_2(t) \quad ... \quad x_M(t)]^T \qquad (3,10)$$

$$N(t) = [n_1(t) \quad n_2(t) \quad ... \quad n_M(t)]^T \qquad (3,11)$$

$$S(t) = [s_1(t) \quad s_2(t) \quad ... \quad s_M(t)]^T \qquad (3,12)$$

$$\mathrm{A} = [\alpha(\theta_1) \quad \alpha(\theta_2) \quad ... \quad a(\theta_\kappa)] \qquad (3,13)$$

with $\alpha(\theta_\iota)$ being the steering vector of the array toward the direction θ_ι.

The antenna array performs the mapping : $G: R^K \rightarrow C^M$ from the space of DOA $\{\theta = [\theta_1, \theta_2, ..., \theta_K]\}$ to the space of sensor output $\{s = [s_1, s_2, ..., s_K]\}$. That is for each incident signal s_m, an angle θ_m is associated with it.

In [18] a neural network is introduced to perform the inverse mapping $F: C^M \rightarrow R^K$. The network is to be trained by N patterns so that it can associate the output vectors s(1),s(2),…s(N) with the corresponding DOA vectors $\theta_1, \theta_2, ..., \theta_N$

Neural Network Training

First the array output vectors are generated, then transformed into appropriate input vectors to be presented to the network. Assuming that the noise signals received at the different sensors are statistically independent white noise signals the author derives the correlation matrix R of the received noisy signals.

All the cross-correlated terms between signals as well as the initial phase that contain no information about the direction of the incoming signals are eliminated from the training data. So R is transformed into a new input vector b which is normalized by its norm in the vector z.

An appropriate training procedure is employed on an RBF neural network to learn the training set z [18].

The main advantage of using an RBF-NN over other approaches is that it does not require training the network with all possible combinations of input vectors. For the network to generalize, it is sufficient to perform the training with vectors that span the expected range of input data.

For uncorrelated signals the RBF-NN method proved to approach the MUSIC algorithm. In the case of correlated or coherent (perfectly correlated signals being received by the array the RBF-NN outperformed the conventional MUSIC algorithm yielding smaller error. The RBF-NN in this case just takes into account the correlation between incoming signals when the correlation matrix R is generated for training.

Direction Of Arrival for multiple sources using multilayer neural networks.

The neural multiple source tracking (NMUST) is introduced where an arbitrary number of mobile users can be tracked and no prior knowledge on the number of mobile users is required. There are two stages of the RBF-NNs. The first is the detection stage which is divided into neural network sectors. The entire angular spectrum (field of view of the antenna array) is divided into these neural network sectors. An RBF-NN is trained in each sector to determine if one or more signals exist within this particular sector. If there are any signals present the neural network will give the value 1 for an answer, otherwise it will give zero as its output value. This information is then passed to the second stage, the DOA stage which operates as described previously. The main advantage of this technique is a dramatic reduction in the size of the training set since much fewer possibilities need to be considered by sectoring the antenna field of view.

In [19] a fast neuro-beamformer is presented also based on the radial-basis function network. To meet real-time requirement, the authors customized the basis function for fast computation, and applied recursive least square learning rule to speed up the network training. By comparing the effects of center location and distribution, they achieved a minimum network for recalling. The network recalling does not require the knowledge of direction-of-arrival, and thus the method is reported as a blind method.

3.4 Neural Networks in radio resource management

In [20] the authors propose a new method for increasing network capacity by introducing an adaptive radio resource management system into a typical GSM/GPRS network. The adaptive radio resource management system predicts future radio resource requirements in terms of channel demand, for both circuit switched GSM calls and packet switched GPRS sessions using neural networks. Frequency assignment is then performed using a genetic algorithm. Neural network schemes for radio resource management operate in real-time, with each cell requiring information regarding channel usage in neighbouring cells.

The authors show that resource predictors based on multi-layered feed forward neural networks (MFNNs) can make accurate predictions when trained with sufficient amount of historical data. The system proposed considers a MFNN for each type of traffic at each cell. Each MFNN contains three layers with a total of 49 neurons. The back-propagation learning algorithm and non-linear sigmoid activation function are used in the learning process. The training and prediction of the resource predictors proceeds as follows:

1. Hourly radio resource demand statistics for GSM calls (new and handover) and GPRS sessions are collected.
2. Then it is recorded whether the demand occurs on a weekday or weekend (day statistic). Then the time is recorded (time statistic). These statistics constitute the initial training data set.
3. The MFNNs are trained using the data arising from step 2.
4. Once the MFNNs are trained, the channel demand for the next hour in each cell is predicted using the demand statistics from the previous 10 hours, the day and time statistics.
5. The predicted number of frequencies for each traffic type is assigned to each base station.
6. The training set of 8 weeks is updated to contain the statistics for the current hour (assuming the network gathers statistics at least every 60 minutes).
7. Each MFNN is retrained every 24 hours to maintain accurate predictions.

Finally it is reported that MFNN can be implemented within the Operating Maintenance Center of a GSM network which makes it a viable proposal for a more flexible management of resources for 2.5G networks. Simulation results have shown resource gains of up to 20.7% and 12.5% for GSM and GPRS traffic respectively. The authors conclude that their results exhibit the less resource requirements than existing fixed channel allocation networks and performance that is comparable to recently proposed dynamic resource allocation schemes, but with the advantage of significantly less complexity and no additional network signalling load.

3.5 Neural networks applied to the channel assignment problem

The fact that the electromagnetic spectrum available for cellular mobile communication systems is a limited resource places severe limitations on the size and performance of such systems. Careful design of a network is necessary to ensure efficient use of the available spectrum.

The term channel is normally used to denote a frequency in the FDMA system, a time slot in the TDMA system and a code in the CDMA system or a combination of these in a mixed system. Two channels are different if they use different combinations of these at the same place. Channel assignment is a complex process where a finite number of channels are assigned to various base stations and mobile phones. In a system with fixed channel assignment (FCA) channels are assigned to different cells during the planning stage and this assignment rarely changes to reflect the traffic needs. A channel is assigned to the mobile at the initiation of a call and the MS communicates with the BS using this channel until it remains in the cell.

Dynamic channel allocation (DCA) is an efficient way of channel usage in a multiple-user environment. In this allocation scheme, a channel with the minimum interference is found before assignment. The interference level of all the channels used and unused is monitored, and during the call the channel assignment may be changed from one with high interference to one with low interference.

There are three kinds of interference: another caller within some range using the same channel (a co-channel interference); another caller in an adjacent region using an adjacent channel in the frequency domain (an adjacent channel interference); and another caller within the same region using another channel within some range (a co-site interference). The channel assignment problem is then to assign the required number of channels to each region in such a way that interference is precluded and the frequency spectrum is used efficiently.

It has been shown that the FCA problem is a generalised graph colouring problem and is therefore NP-hard.The introduction of artificial Neural Networks to implement channel assignment schemes enables retaining flexibility without compromising on speed. Hopfield neural networks have been used to solve the FCA problem

When channel assignment problem is examined in the frequencies point of view it is referred as frequency assignment problem (FAP).

In [21] a short review on the literature concerning the use of neural networks in the channel assignment problem is presented.

It is reported that the basic frequency assignment problem consists of assignment constraints, interference constraints and an objective function. A convenient representation of interference is by means of a graph $G = (V, E)$, called the interference graph or the constraint graph. Each antenna is represented by a vertex $v \in V$. Two vertices v and w for which the corresponding signals may interfere for at least one pair of transmitting frequencies, are connected by an edge $\{v, w\} \in E$. If the available channels (frequencies) are denoted by $F = \{1,..., N\}$, then the channels available for a connection or antenna u form a subset $F(u)$ of F. For each pair of frequencies $f \in F(v)$ and $g \in F(w)$ the combined choice is penalized by a measure depending on the interference level. In most models this penalty has a specific structure: it depends only on v and w and the distance between the frequencies $|f - g|$. FAPs with this structure are called distance FAPs.

It is stated that a standard way to define the neural network for FAP is the following: associate a neuron V_{if} with each pair (i, f) where $i \in V$ and $f \in F_u$. Two neurons are coupled if the corresponding vertices are adjacent in the interference graph. The energy function is the weighted sum of several terms representing different types of interference constraints (co-cell, co-site,…), demand constraints (number of required frequencies) and sometimes instance specific requirements.

An objective function that minimizes the sum of the penalties incurred by the frequency choices is used in the so called Minimum Interference Frequency Assignment Problem. (MI-FAP)

In [22] a MI-FAP model is presented. The coupling weight between two distinct neurons V_{if} and V_{jr} depends on the type of interference relation (co-cell, co-site,…) between the corresponding two assignments. Computational experiments were performed on random hexagonal networks.

In [23] the model is distance MI-FAP. The local updating rule consists of two terms. The first term is proportional to the demand deficiency while the second is proportional to the distance violations.

In [24] the dynamic MI-FAP is considered. At each iteration the current assignment is updated to take into account new requests of connection. Only those vertices involved in new connections are re-optimized. The energy function is the weighted sum of several terms to handle interference level, unsatisfied demand, number of distinct frequency assigned. Computational experiments are performed on hexagonal networks with non-uniform traffic distribution.

In [25] the neurons can only assume binary values. The energy function takes into account several types of interference constraints and the level of unsatisfied demand. This demand is translated into an additional input to each neuron which forces the assignment of new frequencies to vertices with unsatisfied demand.

In [26] the problem has been modeled as a 0-1 quadratic optimization problem with linear constraints. Interference is dealt with as a soft constraint and the number of demanded channels as a hard constraint. In order to minimize the interference two kinds of tools have been used: neural networks and genetic algorithms. The neural networks used were the Hopfield and the self-organising neural networks.

The problem is characterized by a number of cells cel, and a number of channels ch. The solution of the problem is represented by matrix X. Each element of X is defined as:

$$x_{ij} = \begin{cases} 1, & ifchannel \quad j \quad is \quad assigned \quad to \quad cell \quad i \\ 0, otherwise \end{cases} \qquad (3,14)$$

The interference matrix C expresses all kinds of interference effects. Each element of this matrix represents the distance between two channels assigned to cells i and j. The number of channels demanded to the system is given by vector D, where each d_i is equal to the number of required channels in cell i. The objective function has two terms: an interference term and a penalty term. The first forces the algorithms to minimize the interference and the second forces to minimize the demanded channels to be assigned.

The objective function becomes:

$$F(x) = c^T x + x^T Qx, \quad com \quad c^T = 0 \qquad (3,15)$$

where matrix X has been transformed into x and matrix Q represents the interference as explained in [26].

The Hopfield neural network due to its inherent capability of minimizing an associated Liapunov energy function is a very good tool for solving problems involving function minimization. In [26] the issue is to minimize interference. The energy function for a Hopfield NN in its matrix form is:

$$E = -\frac{1}{2}x^T W x - i^T x \qquad (3,16)$$

It is shown though that the similarity between the last two equations allows the mapping of the channel assignment problem onto a Hopfield network.

In the case where the SOM NN is used, the goal remains the minimization of interference.The inputs of the SOM NN are the lines of a permutation matrix formed from the demanded channel vector D. All the values of these inputs are zero except for the cell where a new channel is required which is one. The network evaluates the cost of assigning the new demanded channel to every available channel on the system and assigns it to the one with the lesser cost. The weights W are updated to reflect this new reality and the matrix W is the output of the algorithm.

In [27] two channel assignment algorithms are introduced that make use of the concept of re-use groups. Re-use groups take care of the interference constraints that are enforced to ensure clear reception. A Hopfield neural network is used which is called HNN-DCA and a Dynamic Channel Assignment (DCA) algorithm named MaxAvail. The same model is used as a building block for the HNN-LCR, an assembly of neural networks, which carries out channel assignments with the option of rearrangement according to a Limited Channel Rearrangement (LCR) algorithm named Remax1. A detailed description of both algorithms can be found in the paper.

In [28], the authors also suggest that the channel allocation problem can be reduced to the search for a global minimum if the system is represented with an objective function whose minimum is associated to a good configuration and the various constraints appear as penalty terms in the function. Several heuristics can be used such as Hopfield neural network, which is also used in [29].

3.6 Neural fraud detection in mobile-phone systems

Fraud in communication networks refers to the illegal access to the network and the use of its services. It is estimated that a mobile phone network operator may lose as much as million dollars a day due to fraudulent usage of mobile phones. A typical example of fraud is the subscription fraud, where a fraudster acquires a subscription to the mobile network under a false identity and starts reselling the use of his phone to unscrupulous customers (typically for international calls to foreign countries) at a rate lower than the regular tariff. The fraudster accumulates a large number of expensive calls, but disappears before the bill can be collected.

Intelligent data analysis methods are used to solve the fraud detection problem. Neural networks have been successfully applied to the fraud detection procedure in mobile communication networks.

In [30] fraud is defined from the unobserved intentions of the mobile phone users. These intentions are reflected in the observed call data which is used in describing behavioral patterns of users. The task is to use the call data to learn models of calling behavior so that these models make inferences about users‘ intentions. Neural networks and probabilistic models are employed in learning the usage patterns from call data. Learning here means adaptation of the parameterized models so that the inherent problem structure is coded in the model. Since there is no specific sequence of calls that would be fraudulent with absolute certainty, uncertainty in modeling the problem is needed which is embodied in the framework of probabilistic models.

A short review on the fraud detection techniques based on neural networks takes place in [30].

In [31], the authors report their first experiments detecting fraud in a database of simulated calls. They use a supervised feed-forward neural network to detect anomalous use. Six different user types are simulated stochastically according to the users' calling patterns. Two types of features are derived from this data, one set describing the recent use and the other set describing the longer term behavior. Both are accumulated statistics of call data over time windows of different lengths. This data is used as input to the neural network. The performance of their classifier is estimated to be 92.5 % on the test data.

In [32], work on fraud detection based on supervised feed-forward neural network techniques is reported. The authors criticize thresholding techniques by detecting excessive usage, since these might be the very best customers if these are legitimate users. In order to use supervised learning techniques, they manually label the estimated user profiles of longer term and recent use, similar to those in [33,34] into fraudulent and non-fraudulent and train their neural network on these user profiles.

In [32], they report having classified test data with detection probabilities in the range of 80-90 % and false alarm probabilities in the range of 2-5 %.
Some work in fraud detection is based on detecting changes in geographical spread of call destinations under fraudulent activity. In [35] the authors use neural networks in classification of such kind of call data.

In [36], first, feed-forward neural network based on supervised learning is used to learn a non-linear discriminative function between classes fraud and non-fraud. Secondly, density estimation with Gaussian mixture models is applied to modeling the past behavior of each subscriber and detecting any abnormalities from the past behavior. Lastly Bayesian networks are used to define probabilistic models under the assumptions fraud and non-fraud.

As long as the neural-network method is concerned, the feed-forward network consisted of five hidden units and one binary output. It was trained using Quasi-Newton optimization. The features used were average and standard deviation of the duration and the number of calls made during the day, maximum duration and number of calls per day during the observed time period. The data set contained time series of toll tickets, which are call records stored for billing purposes, from both fraudulent and legitimate users. The toll tickets are created for each phone call made and include information like the identification of the caller, starting time of the call, duration of the call, the called party number etc. The features considered were summary statistics over the whole observed time period. The output of the neural network was interpreted as the posterior probabilities of fraud given the inputs.

The neural-network method detected over 85% of fraud cases without causing false alarms.

The AspeCT Project-BRUTUS

In 1996 the ASPeCT project partners, namely the Katholieke Universiteit of Leuven, Siemens Munich, Vodafone UK, Panafon GR and Royal Holloway University of London, introduced new fraud detection concepts that have now been implemented in three separate intelligent techniques - four stand –alone tools, which were then combined into a powerful fraud detection engine (called BRUTUS).

Separate tests were performed, on the three approaches adopted, to ensure each of the techniques, both neural network based and rule based, were capable of performing the task at hand. The demonstrations produced positive results.

In [37], BRUTUS is referred as a hybrid detection tool utilising both rule-based and neural network technologies that enable the profiling of both network subscribers and network traffic. The main idea here as well is that in mobile phone fraud detection when fraud occurs, there will nearly always be an observable change in the behaviour of the mobile phone. This fundamental principle not only applies to mobile phones but applies to network surveillance in general.

The researchers tried to maintain histories of usage information, relating to the entity, over differing time periods. They refer to the short term past behaviour as the Current Behaviour Profile (CBP) and the long term past behaviour as the Behaviour Profile History (BPH). The detection engine has to determine if a significant change in behaviour has occurred. This is known as performing a differential analysis. When the CBP exceeds predetermined thresholds for acceptable network usage over the lifetime of the CBP, alarms can also be raised. This is known as performing an absolute analysis

In the case of detecting fraud on mobile telecommunications networks, as stated before, behavior profiles are built from Toll Tickets. A toll ticket is a bill issued by the network after each call, which contains all relevant information about the call. The information they used is the International Mobile Subscriber identity (which identifies a user uniquely, the starting date of the call, the starting time of the call, the duration of the call, number that was called, the type of the call (national or international using features.

One of the profiling techniques used was an unsupervised neural network to develop prototypes of call records in order to build statistical behavior profiles maintained as probability distributions. The strength of this representation is in detecting new fraud scenarios, important for surveying new applications.

The authors report that if cell sites are considered as the network entity to profile, a CBP for each cell can easily be built. If the system is operating in real time, BRUTUS will detect sudden drops in the activity of cells and be able to warn the appropriate bodies that there is a physical problem on the network.

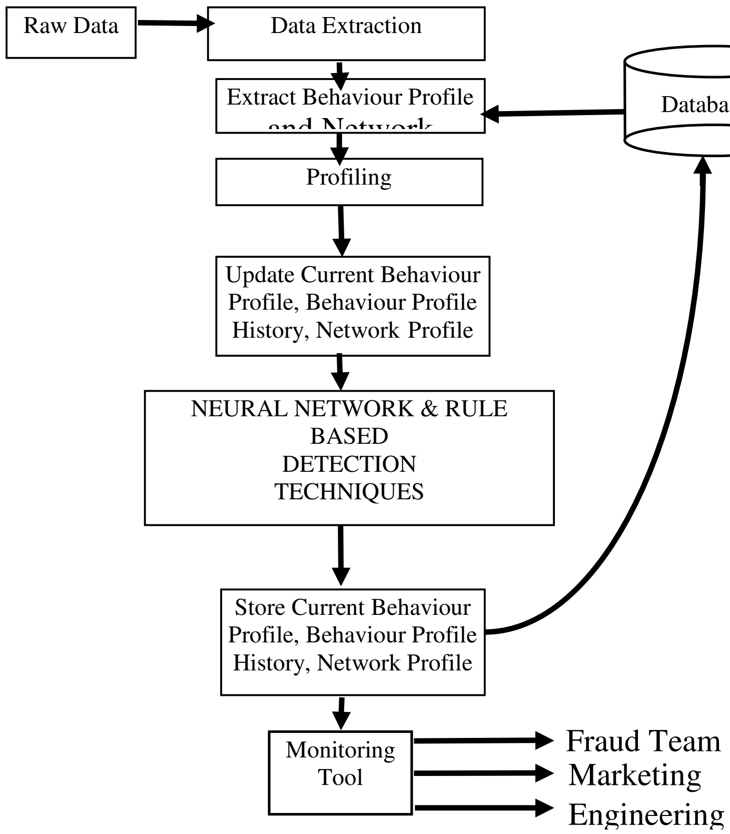

Figure 3.2 : BRUTUS engine description

The detection components of BRUTUS were introduced in detail in [38].

The fraud detection engine BRUTUS consists of :

1. A rule-based tool, allows detecting the definite frauds with a low rate of false alarms. It can easily provide reasons for an alarm being raised. The rules for triggering of an alarm are designed manually by an expert.
2. A supervised neural network: The classifier used in BRUTUS is a multilayer perceptron with a hidden layer of logistic-sigmoid neurons. The problem was considered to be a two-class one for which the authors used a weighted squared error as cost function. The minimization was performed using the Levenberg-Marquardt algorithm. After having repeated these procedures for different architectures of the neural network they used the optimal one on top of the front-end and it would produce an alarm value between 0 and 1 each time a toll ticket was presented to the fraud detection tool.
3. Two unsupervised neural networks were used to look at how a user's behavior changes over time. These don't need prior knowledge of fraud. Two profile records are generated for each user by considering two different time spans over the toll tickets. The unsupervised neural network is based on a prototyping technique reminiscent of vector quantization or clustering. Prototyping is a method of forming an optimal discrete representation of a naturally continuous variable.

Under BRUTUS the various detection components are able to feed information to each other sharing profiling techniques and detection results. For example the supervised learning system uses the Hellinger distance between the elements of the CBP and BPH produced by the unsupervised neural network. The Rule based system may use the detection results of the other two components to add evidence to its analysis. The grand finale of the whole process is the forwarding of alert status' from the merged detection components to an intelligent monitoring tool capable of combining the alerts and determining what action should be taken.

Performance

For the unsupervised neural network tool, mainly low alarm values are excited. The performance of the supervised neural network is proved to be higher than that of the unsupervised neural network.

Since the unsupervised neural network is working with no prior knowledge of the fraud scenarios it would encounter and thus, as expected, produced slightly inferior results. The strength of unsupervised learning is in the development of profiles that maximise information entropy, given discretisation restrictions, in order to facilitate the detection of new fraud scenarios

In [39], the authors present an on-line security system for fraud detection of impostors and improper use of mobile phone operations based on a RBF neural network classifier. The neural network acts solely on the recent information and past history of the mobile phone owner activities, and classifies the telephone users into classes according to their usage logs. Such logs contain the relevant characteristics for every call made by the user. As soon as the system identifies a fraud, it notifies both the carrier telecom and the victim about it immediately and not at the end of the monthly bill cycle. In their implementation, they used the radial basis function (RBF) model because of its simplicity and its flexibility to adapt to pattern changes. The reason was that by learning, a RBF NN can discover some regular patterns and the relation across them, and organize itself for making these associations. Their results indicate that their system reduces significantly the telecom carriers's profit losses as well as the damage that might be passed to the clients.

3.7 Neural Networks in Path Loss Prediction

Most cellular radio systems operate in urban areas where there is no direct line-of-sight path between the transmitter and the receiver, and where the presence of high buildings causes severe diffraction loss. Due to multiple reflections from various objects, the electromagnetic waves travel along different paths of varying wave lengths. The interaction between these waves causes multipath fading at a specific location and the strengths of the waves decrease as the distance between the transmitter and the receiver increases. So as the MS moves away from the transmitter over much larger distances the local average received signal will gradually decrease.

The path loss, which represents signal attenuation as a positive quantity measured in dB, is defined as the difference in dB, between the effective transmitted power and the received power.

$$PL(dB) = 10\log\frac{P_t}{P_r} \qquad (3,17)$$

where P_t is the transmitted power of the BS's antenna and P_r is the received power by the MS's antenna.

The classical path loss prediction models are

- empirical : are described by equations derived from statistical analysis of a large number of measurements
- deterministic: are based on the application of well-known electromagnetic techniques to a site-specific description of the environment (buildings and terrains)
- semi-empirical or semi-deterministic: are based on the equations derived from the application of deterministic methods to generic urban or indoor models.

In literature, neural networks are examined as an alternative approach to the classical path loss prediction models. Their advantage is the flexibility to adapt to arbitrary environments, high speed processing and the ability to process a high quantity of data.

On the other hand, one of the main parameters determining the quality of the communication link is the received electric field strength from a central base station. This information as showed previously is necessary in the path loss prediction problem. A number of models for the prediction of the field strength is given in literature, but all models have their individual advantages and disadvantages. There is nostandard model, especially for the prediction of the electric field strength inside buildings. Neural networks have successfully been applied in this field as well.

In [40] the authors consider a function approximation problem that consists of a non-linear mapping from a set of input variables containing information about potential receiver onto a single output variable representing the error between measured path loss and the path loss obtained by applying the modified COST231-Walfisch-Ikegami (a semi-deterministic) model.

The General regression neural network is a neural network architecture that can solve any function approximation problem. The proposed neural network is trained with physical data, which include the distance between transmitter and receiver, the width of the streets, the height of the buildings, the building separation and the distance between base station and roof top height. The GRNN model has a single output which represents the normalized propagation path loss. The training set is from real measurements a in greek city of Kavala. Comparing the neural network approach with the empirical models: Walfisch- Bertoni, the single slope model and the modified COST231-Walfisch-Ikegami model, there is a significant improvement in the prediction made by neural models due to their generalization property. Another advantage is the fact that they are trained with measurements, so the included propagation effects are more realistic.

Indoor Propagation Models

Eventhough, the field strength prediction in indoor environments seems be to easier than the outdoor prediction, at a specific location, the electric field of the indoor environment is formed by a much larger number of indirect components than in the case of the outdoor environment. Therefore, the indoor signal level is more fluctuating than the outdoor signal level, and thus it is more difficult to predict. The problem of the indoor field level prediction can be considered statistically or theoretically. While almost all statistical (empirical) models are based on the same general model, there are several distinguished theoretical models of which ray-tracing models and Finite-Difference Time-Domain (FDTD) models are the most popular. Some important disadvantages of both empirical and theoretical models can be overcome by an appropriate artificial neural network model.

In [41] a field strength prediction model is presented based on neural networks, trained with measurements. The author bases his research on the prediction of neighbouring points. That is, based on the field strength at a central point C a neural network is used to predict the field strength at the four neighbouring points N1, N2, N3 and N4.

After this initial prediction the field strength of the neighbouring pixels of N1, N2, N3 and N4 is computed depending on the field strength of N1, N2, N3 or N4. So the former neighbour pixel becomes the new center pixel.

The neural network used is a feed forward perceptron, trained with the Resilient Backpropagation Algorithm. Eight input parameters are used for the neural prediction model:

1. Field strength at centre point
2. Distance between transmitter and receiver
3. Visibility LOS { Line of sight, OLOS { Obstructed line of sight (same room, but no line of sight), NLOS { Non line of sight (different room))
4. Orientation : Neighbour point nearer to the transmitter as central point or vice versa
5. Shape of the room Corridor, small room, hall
6. Transmission: Transmission loss of a wall between neighbour and centre point
7. Immunity Improbability of time-variant effects
8. Waveguiding : Number and distance of walls, parallel to the line between centre and neighbouring point

For optimum performance each input parameter is normalized with different functions to a range between -1 and +1.The network contains one simple output, only representing the field strength in one of the neighbouring points.

After the training of the network, predictions were made in buildings and with transmitting positions not trained with the measurements, to show the generalization capability of the neural network prediction model. Good agreement between measurement and prediction was obtained.

In [42], neural networks are applied to both outdoor and indoor field strength prediction. More specifically:

1) Neural Networks Outdoor Model

It is reported that the ability of feed-forward neural networks with sigmoidal activation functions to demonstrate very good performance in solving problems with mild nonlinearities on the set of noisy data, fully corresponds to the problem of the field strength prediction. The data obtained by measurements are always noisy. Another key feature of neural networks is the intrinsic parallelism allowing for a fast evaluation of solutions. The process of learning may last for a couple of hours, but the process of field strength prediction is fast.

The neural network model proposed here, is based on a multilayer perceptron feed-forward neural network architecture. The proposed neural network has three groups of inputs. The first group consists of an input only and it is the normalized distance from the transmitter to the receiver. The second group of inputs (4 inputs) is based on the terrain profile analysis. These inputs are: 1) portion through the terrain;

2) and 3) modified "clearance angle" factors for both the transmitter and the receiver sites, respectively; (this angle should be a representative of those angles in the reception area measured between the horizontal line at the receiving antenna, and that which clears all obstacles within 16km in the direction of the transmitter and 4) the rolling factor. The third group of input parameters is based on the land category analysis along the straight line drawn between the transmitter and the receiver. There is a single input for each defined land use category. The network has one output and it is a normalized electric field level. The implementation of the proposed neural network model requires two databases. The first is the standard digital terrain elevation database; the other is the ground cover (i.e., land usage) database. The author concludes that, in comparison to other popular prediction models, the neural network model demonstrated very good performance. The effects of urbanization are considered more subtly in the proposed model than in standard empirical models providing greater accuracy. On the other hand, the neural network model is not computationally as extensive as deterministic models. The neural network model has been realized and used in 450MHz and 900MHz frequency bands for the purpose of TETRA and GSM system design, respectively.

2) Neural Network Indoor Model

Similar to the case of the neural network macrocell model, this model is based on multilayer perceptron feedforward neural networks. The implementation of a neural network model requires a database of the floor plan in which all particular locations are classified into several environmental categories, for example, wall, corridor, classroom, window, etc. The easiest way to do this is to make a color picture over the scanned floor plan. The neural network model proposed is a multilayer perceptron with three hidden layers. There are several inputs based on the number of previously defined environmental categories. One of the inputs is the normalized distance from the transmitter to the receiver. The remaining inputs are based on the analysis of the straight line drawn between the transmitter and the receiver with respect to environmental categories, e.g, how many doors, what percentage of the line passes through the classrooms, etc. The model has an output that is a normalized field level.

Determining the parameters of the neural network model is very simple. Statistical analysis is unnecessary. The neural network should just be trained with the measured data. Computationally the training process is very extensive, but it is done only once. In implementation, it was shown that the accuracy of the neuralnetwork model is very high.

3.8 Mobile Station Location using Neural Networks

Location management is one of the key issues in mobile networks to provide efficient and low-cost services.

Future mobile networks will have to support large population of mobile users and have to provide efficient and low-cost services under diversified characteristics of network architecture, services,and user types, i.e. different cell sizes, multimedia services, different mobility users, etc. One of the most important issues in the mobile networks is the location management. L*ocation management* is the process by which the current location of a mobile station (MS) is determined. It can be divided into two different services: *mobile tracking* and *locating*. *Mobile tracking* is the process by which the network keeps track of the current location of the mobile station, whereas *Mobile locating* deals with the process of finding the current location of the mobile station for the delivery of an incoming call. One way of searching an MS to identify its current location is to broadcast a search information, called *paging,* in each and every cell in the whole geographical area. However, the amount of channel bandwidth consumed by these numerous broadcast signals can be extremely high. The other way of searching is to store the current location of each MS by a *location update* signal from the MSs and maintaining the location information database for each MS, which is more expensive and has high signalling traffic.

In [43] the authors propose a prediction-based location management using multilayer neural network (MNN). The MNN is trained with respect to the data obtained from the history of movement pattern of a mobile station for making predictions.

The authors define m*ovement pattern* (*Pn*) as the history of movement of a mobile host recorded for a period of time *Tn*, where *n* is the number of regular time intervals at which the mobile station movements are recorded. Then the movement pattern *Pn* = {*p*1, *p*2, ...,*pn*} recorded for a mobile MS, where *Pi* indicates the movement of a mobile station during time *ti*, the movement is defined in terms of direction and distance travelled by an MS during the time interval *ti*. Then *Pi* is represented by a pair (*di, dsi*).

- *di* is the possible direction in which an MS moves at *i th* time interval. *di* $\in$ *(North, East, South, West*}.
- *dsi* is the distance travelled by a mobile station at *i th* time interval. Here, the distance traveled may be number of cells, kilometers, meters, etc.

For example, if a mobile movement pattern is recorded for three time intervals (*n* =3) with direction of movements, North, East, East and the distance travelled is 2, 1, and 3 units, then the movement pattern is, *P*3 = *p*1, *p*2, *p*3} = {(*d*1, *ds*1), (*d*2, *ds*2), (*d*3, *ds*3)} = {(North, 2), (East, 1), (East, 3)}.

As training data set they use the set of subpatterns obtained from the movement pattern *pn* by partitioning it into $n-k$ subpatterns, where *k* + 1 is the size of each subpattern (*k* << *n*). The subpattern is a training data pair with mobile movements for *k* time intervals as input and the movement for the next time interval as a desired output. For example, the first training subpattern is *p1, p2, ..pk* as input and *pk*+1 as the desired output. The parameter *k* is the prediction order or time window, which is chosen based on the movement characteristics of a mobile station and the size of the recorded movement pattern. Similarly, the *i th* training subpattern contains *pi, pi*+1, ...,*pi*+$\tilde{k}$1 as the inputs and *pi*+*k* as the desired output.

The multilayer neural network devised for mobile movement prediction is based on the back-propagation learning algorithm. The role of the neural network in this application is to capture the unknown relation between the past and the future values of the movement pattern. This helps in predicting the future location of a mobile station.

The number of input neurons is considered as an important parameter since it corresponds to the length of the subpatterns used to discover the underlying features in a given movement data.

To predict the future movement of a mobile station, single or multiple move prediction were defined.

Single move prediction

This predicts the movement (p_{n+1}) of a mobile host at time interval t_{n+1} for the given movement pattern p_n. To carry out single move prediction, the subpatten $\{p_{n-k+1}, p_{n-k+2}, .., p_n\}$ is applied to the input of the MNN and the output obtained is the predicted movement p_{n+1} for time interval t_{n+1}, which is the direction d_{n+1} and the distance ds_{n+1} travelled by a mobile station.

Multiple Move Prediction

This predicts the movement of a MS after several time intervals from t_n time interval, i.e. to predict the movement (p_{n+m}) of a MS, where m>1.A recursive method has been designed where the predicted output p_{n+1} is inserted as one of the inputs in subpattern at the extreme right by shifting the entire subpattern to left by one time interval to predict the next movement p_{n+2}.

The mobile movements were recorded in terms of cell number at every time interval. The recorded mobile movements were preprocessed to obtain the movement pattern. For example, let an MS move in direction *x* at *i th* time interval from the current position, then *di* = *x* where *x* ∈ *D* = {*North, NorthEast, East, SouthEast, South, SouthWest, West, NorthWest*} = *N, NE, E, SE, S, SW, W, NW*}, a set of possible directions, and if an MS is moving at a certain speed, then the distance travelled at *i th* time interval is *dsi* = *y*, where *y* is the number of cells crossed by an MS during the *i th* time interval. The movement pattern *Pn* for a mobile station is obtained To determine the direction and distances moved by an MS the author uses an adjacency matrix corresponding to the neighbours of a cell.

The rationality behind the prediction of location of a mobile host is based on the following. In general, the movement of mobile hosts shows some pattern according to their movement behaviour. For example, users living in urban areas move daily from home to office and back home. If the movement patterns for the previous several hours, days or months are investigated, some periodicity for the patterns will be exhibited. This periodicity is a key to predicting the future location of a mobile host. But majority of mobile users have some regular daily movement patterns and follow these patterns more or less every day of the week except during weekends or holidays when the movement is random; hence the movement patterns are stochastic in nature.

The actual algorithm introduced in [44] is the following:

Algorithm: Prediction-based location management

BEGIN

Step 1: Movement pattern *Pn* for an MS is recorded for a time duration [0, *Tn*);

Step 2: The direction and distance from the movement pattern is derived using adjacency matrix;

Step 3: *Subpatterns* for the movement pattern are derived as a training data set for training MNN;

Step 4: MNN is trained

Step 5: IF MS is to be located at time *Tc* > *Tn* THEN

Compute [m] =
IF (m > 1) THEN
Carry out *multiple move prediction* to predict the movement p_{n+m};
ELSE
IF (m= 1) THEN
Carry out *single move prediction* to predict the movement pn+1;
Step 6: Using adjacency matrix, the location of an MS is found from the predicted direction and distance, given the current location of the MS;
END

Where ,Pn = Recorded movement pattern for time duration [0, Tn), Tn = Time, up to which the movement pattern is recorded, Tc = Time at which the location of a mobile station is to be predicted, Δt = Time interval, at which mobile movements are recorded, m = Number of time intervals ahead of the prediction of location of a mobile station

The performance of the method has been verified for prediction accuracy by considering different movement patterns of a mobile station and learning accuracy of the MNN model. Simulation is also carried out for different movement patterns (i.e. regular, uniform, random) to predict the future location of a mobile station. The average prediction accuracy was measured and achieved up to 93% accuracy for uniform patterns, 40% to 70% for regular patterns and 2% to 30% for random movement patterns.

In [44], the authors used only constant time intervals to build the user profile. A mobile network with 20 location areas was chosen, numbered from 1 to 20. That means that they registered each location update generated from the MS when changing the location area. The user had to be observed to store his movement patterns as a function of time m(t). In the next step the suitable neural network was trained with the observed motion pattern m(t). The contents of the input and output vectors depended on the neural network type. Feedback neural networks are able to remember the order in which the input patterns are presented during the training, because of their backward connections. So, it was only necessary to give the last known location area number as input vector.

Feed-forward networks are memory-less, their output is independent of the previous inputs. The input vector consisted of the N last known locations and the absolute time (day,hour) the change to the location was observed.
The output information was the location area number of the next movement. A second information in the output vector was the probability that the subscriber was located in the predicted location areas.

For the feedback type, the authors chose Jordan, Elman and Hierarchical Elman networks. The learning algorithms studied were backpropagation, quick prop and resilient propagation.
First results seemed to be promising; they showed correct prediction probabilities around 80%

A comparison with conventional methods is also presented in [44].

In [45], a novel approach to mobile station positioning using a GSM mobile phone is described. The approach is based on the use of an inherent feature of the GSM cellular system (the mobile phone continuously measures radio signal strengths from a number of the nearest base stations (antennas)) and on the use of this information to estimate the phone's location. The current values of the signal strengths are processed by a trained neural network executed at the computer attached to the mobile phone to estimate the position of the mobile station in real time. The neural network configuration is obtained by using a genetic algorithm that searches the space of specific neural network types and determines which one provides the best location estimation results. Two general methods are explored: the first is based on using a neural network for classification and the second uses function approximation. A detailed description of the neural network can be found in [45].

Apart from the location prediction of the Mobile Station, its motion prediction (mobile tracking), which involves studying the motion pattern of the MS, is also important in the mobile communication networks. In [46], several motion prediction methods have been presented most of which are based on an aggregate history of hand-offs observed in each cell. The behavior of a MS is assumed to be probabilistically similar to the MSs which came from the same previous cell and are now in the current cell. Simulation shows these approaches work fairly well in general geographical areas. However, their adaptability to new environment and graphical situation is not so ideal. In [46], the researchers used Adaptive Neural-Fuzzy Inference System (ANFIS) and statistical methods to solve the problem. In the simulation process, they generated mobile station motion pattern data with specified geographic information which was much closer to the real case compared with probabilistic models. ANFIS integrated neural network and fuzzy inference system together constructing data model with fuzzy inference techniques and choosing model parameter via adaptive neural network method. The adaptive capability of ANFIS makes it almost directly applicable to adaptive control and learning control such as the motion prediction problem in mobile communication systems. The authors conclude that the neural network solution is a promising direction.

3.9 Neural Networks in Cell Planning

The increasing demand for mobile communications leads mobile service providers to look for ways to improve the quality of service and to support increasing numbers of users in their systems. Currently cellular system design is challenged by the need for a better quality of service and the need for serving an increased number of subscribers. Cell planning becomes a key issue in the mobile communication networks which motivates the search for intelligent techniques which may considerably alleviate planning efforts and associated costs.

The main goals of cell planning are:

Coverage: The radio signal coverage must be guaranteed. The received signal strength must always be higher than a threshold according to the receiver sensitivity, and holes in the coverage area should be avoided.

Capacity: In each cell a sufficient number of channels must be available in order to meet its traffic demand for new calls and handoffs.

Transmission quality: The carrier to interference (C/I) ratio of radio channels must satisfy the requirement of transmission quality

Cost: Since the cost and complexity of a mobile network is closely related to base stations the number of BSs must be minimized so as to reduce the cost and the time of the system deployment.

System growth capability: New cells to be added must be integrated with the existing infrastructure.

In [47] an intelligent network planning solution, based on neural networks is presented. The network planning solution consists of 1)a statistical analyzer applied to the original demographic mobile user data (DMUD) representing the network traffic, 2) predicting the proper number of the required clusters/cells based on the statistical nature of the DMUD and the network allocated budget, 3) grouping this data into distinct groups (clusters/cells) based on how they are closely related to each other, 4)testing how efficient the proposed clustering configuration is and 5) network reconfiguration by repositioning the mobile BTSs locations.

The clustering component targets grouping mobile users into clusters so that the degree of association is strong between members of the same cluster and weak between members of different clusters. The idea is to cluster the data in two stages 1) run the Kohonen Self Organizing Map (SOM) algorithm implemented by neural networks (because SOM is widely used for dimensionality reduction and clustering) and 2) measure the distances between each pair of the reference vectors assigned to each neuron in the map. The clusters can be visualized by displaying the distances between each neuron's reference vectors. In clustered map zones, reference vectors are close to each other; however they are sparse in the inter clusters areas. BTSs components are located at the optimal local minima positions of the clustering objective function, which are the clusters' centroids.

F-test is used to measure the clustering separability (the clustering accuracy), in a sense of determining if there exists any significant correlation between different cluster boundaries. F-test is measured by observing differences among the variances of two samples:

$$F = \frac{s_1^2}{s_2^2} \qquad (3,18)$$

As it is showed in the paper, F value is far from 1, which means that net cluster boundaries are more separable.

A detailed analysis of the rest of the components can be found in [47].

In [48] two new methods to find cells in two radio access network scenarios have been presented. In the first, a lower level SOM which represents general mobile cell model is built. Histograms of the states of the base stations are built using clusters of lower level SOM. The same cells can be used later to find out histograms of new data. Thus, the operational mode of each cell and the whole network can be monitored. In the second method, lower level SOMs of one variable are first built. Covariance matrices of the component planes of these SOMs are then used to train another map, which reorders the mobile cells. States of one mobile cell and groups of similar mobile cells are found using agglomerative hierarchical clustering techniques on SOM codebook vectors. It has been shown, that the SOM provides powerful means to move from time consuming and ineffective per cell optimization to cell cluster optimization.

The first method classifies cells using class frequencies as models of cell behavior. The distributions describe how much a particular cell differs from a general cell model, which has been built using as much data as possible. The data which is used to build the lower level SOM in this method should be selected carefully so that it represents well all the possible states of the cells.

The second method is based on covariance matrices of SOM component planes. It uses two level SOMs. In the previous the data was used to build a model of one base station.The same data can also be used to build models of the network. At first SOMs of one variable are built. Each of these SOMs is a model of the network. Next, the component planes of the SOMs are processed. The covariance matrices of the component planes are computed. Covariances of one or more variables are concatenated to be used as profiles of mobile cells. These profiles are the data to a second level SOM. The outputs of the second SOM are the clusters of mobile cells

In [49], the authors divide the network planning procedure into the following steps:

- estimation of cell count
- location optimization of the base stations
- optimization of maximum transmitting powers
- adaption of the antenna patterns

The purpose of the optimization is to suggest the most appropriate location, the transmitting power and the antenna patterns for the base stations with regard to coverage for a CDMA system. Self Organizing Maps perform the optimization process. This kind of network is as well applied to the optimization of base station sites as to the antenna pattern optimization.

The coverage in the area is predicted by a semi-deterministic model based on a propagation neural network.

As each cell has limited capacity one has to estimate the cell count for the area A to be planned first. The required base stations can be determined by the number of users and the offered traffic per user in the microcell. The cost function used to calculate the cell number is :

$$user\_per\_cell = \frac{SE\left[\frac{kbps}{MHz \cdot cell}\right] \cdot B[MHz]}{DR[kbps] \cdot \frac{E[erl]}{user}} \cdot L[\%] \qquad (3,19)$$

$$cell\_number = (\text{int})\left(\frac{\frac{A}{user\_per\_cell}}{\frac{user}{km^2}} + 1 \right) \qquad (3,20)$$

where for simplicity a constant number of user per km^2 and Erlang E per user is assumed, spectrum efficiency SE which is known for a given system, data rate DR, bandwidth B and load L.

The purpose of the optimization is to minimize the number of points with a received signal strength either lower (under-supplied) or higher (over-supplied) than an appropriately chosen threshold. Over-supplied points increase the interference in the adjoining cells in the CDMA system and play therefore also a negative role. The BSs calculated from the previous equation are considered as neurons of the SOM with weights $w = (x, y, z)$. By training the neural with the covered points, the units should be moved in such a way that the over- and under-supplied points in the area concerned decrease. In order to move the BSs, represented by the neurons, towards, the under-supplied points and away from the over-supplied points, there are mainly two possibilities: one can select and present under-supplied points more often to the net or one can increase the influence of the over-supplied points by changing the learning rate. For the optimization of the BS location the learning rate is changed as:

$$n'(t, j) = n(t) \cdot (1 - \frac{F(j)}{\max}) \qquad (3,21)$$

where $F(j)$ is the fieldstrength at point j and max is the maximum fieldstrength in the area.

All coordinates with the appropriate learning rate $n'(t, j)$ are presented for the same number of times to the net. The initial positions of the BSs can be set randomly. For the optimization of the maximum transmitting power the BSs are also represented by the neurons of the net but the transmitting powers are represented in the weight vector w, and not the coordinates.

In micro cells where many BSs are installed in a small area and where the wave propagation is characterized by wave-guiding, antenna patterns play a great role. For the optimization of antenna patterns in the horizontal plane a SOM with the dimension m=1 is used. The weight vectors $w = (\phi, G)$ of the neurons represent the angle ϕ and the gain G in the direction ϕ of the antenna. For the SOM the input vectors $X_j = (\phi_j, G_j)$ are determined from:

$$G_j = T_H \cdot \frac{P_t \cdot G_{pj}}{P_{pj}} \qquad (3,21)$$

where $G_{p,j}$ is the actual gain of the transmitter in the direction ϕ_j, where the path of the highest contribution to the overall fieldstrength of point j is launched. P_t is the actual transmitting power of the respective transmitter and P_{pj} the actual power at point j.

The results presented in [50] indicate that optimizing the location, the transmitting power and the antenna patterns with the SOMs described, is highly suitable for radio network planning. Thanks to the use of neural networks the optimization process requires only a few minutes.

Chapter 4

Application Of neural networks to the cell coverage estimation problem in a cellular communication system

This chapter presents the application of neural classifiers to the cell coverage estimation problem. First the procedure followed and the device used for the collection of the real measurements from Patras is described. From the entire set of measurements a subset has been used to train the neural networks. A presentation of the assumptions made and the preprocessing applied to these data, follows. The two approaches considered are described. Finally, the performance of the best neural networks found is presented.

4.1 Experimental procedure

After determining that the coverage of every base station in the examination area is going to be considered as a separate class, a representative set of data was used to train the neural networks. Three variables were used to feed the networks:

- Distance of the specified spot from the equador meridian = latitude (degrees North)
- Distance from the Greenwich meridian = longitude (degrees East), and
- Power received from the serving Base Station, at the particular spot.

Certain values for the above variables were taken from a special device shown in figure 4.1.

A satisfying number of measurements has been gathered in the center of the city of Patras during the hours 11.00 a.m. and 14.00 a.m. in the first week of December 2000.

The measurements were conducted in sites, from main streets in the centre of Patras. In order to characterise propagation in different directions relative to the streets for urban environment, selected test routes were chosen. The routes involved measurements in LOS (Line-of-sight) and NLOS (no Line- of-sight) paths, parallel and perpendicular streets and zig-zag and staircase streets. That kind of measurement campaigns have taken place in many different areas (urban, suburban etc) around the world.

The measurement set-up includes the mobile test equipment. A general overview of the measurement device that was used is shown in Figure 4.1.

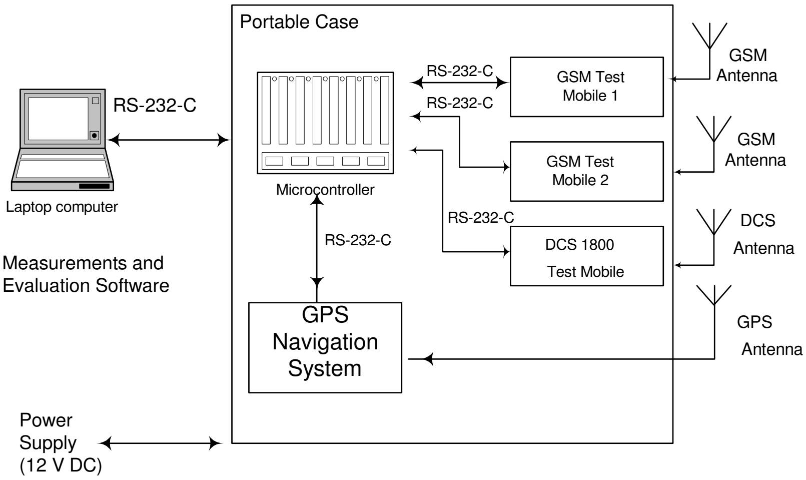

Figure 4.1 : Block diagram of the experimental system

The system shown, uses the intelligence of the used radiotelephones. This means, that they automatically find the operating frequencies of the radio service. Some of the standard measurement functions of the system are the automatic call set up and hang up with separate phone number for each test mobile, variable call duration settings and idle time jointly adjustable for all mobiles (120 sec in our experiment). In our case only one test mobile was used operating at 900 MHz (GSM) which was programmed to initiate a call to a certain fixed telephone number, sited in the centre of Patras as well, every 15 seconds. The most important function of the system is the recording of the peak receiving power. In addition, it is able to record the power received from all six neighbour-BSs. For all the BSs the BCCH-BSIC pair that is used for the identification of each one, is also recorded. In our example, a new power measurement was recorded every second. The micro-controller is responsible for the manipulation of both the test mobiles and the GPS (Global Positioning System) receiver. It allows data to be pre-processed before they are stored in an IBM-compatible laptop. The final software program enables us to observe at any time the signal fading on the computer monitor during the measurement procedure and to have a realistic representation of the followed path. It is also shown when a handoff procedure is initiated and terminated helping us realize when the serving BS (the cell) changes.

The used receiving omni-directional antenna with gain 1 dBi was located at the center of the car's roof at a 1.5 m height above the ground.
The GPS receiver calculates the coordinates of the device from the satellite signals which contain the known positions of the satellites. At any given site it gives the distance of this site from the equador meridian and from the Greenwich meridian in degrees. In this case,

- Distance from equador meridian = latitude (degrees North)
- Distance from the Greenwich meridian = longitude (degrees East)

The GPS error depends on how fast the device is moving. For a stopped car the GPS error is very small (almost 10-15 metres) and it is getting larger as the velocity of the car increases.

4.2 The examination area

The measurements have been gathered along an area that covers a great part of the center of the city of Patras. In this area we can observe 3 BSs which correspond to 3 major macro-small cells, 1 BS operating as an umbrella cell (it is placed on the top of a hill behind all the main streets of the city) and 3 BSs corresponding to 3 micro-cells. As it is shown in figure 4.2, each cell site consists of three directional antennas radiating at 120 degrees arc each. Each directional antenna serves a cell area.

So, in the specified examination area, shown in figure 4.2, we have 3 macro-small cell sites and those cells of them that are noted in the table 4.1

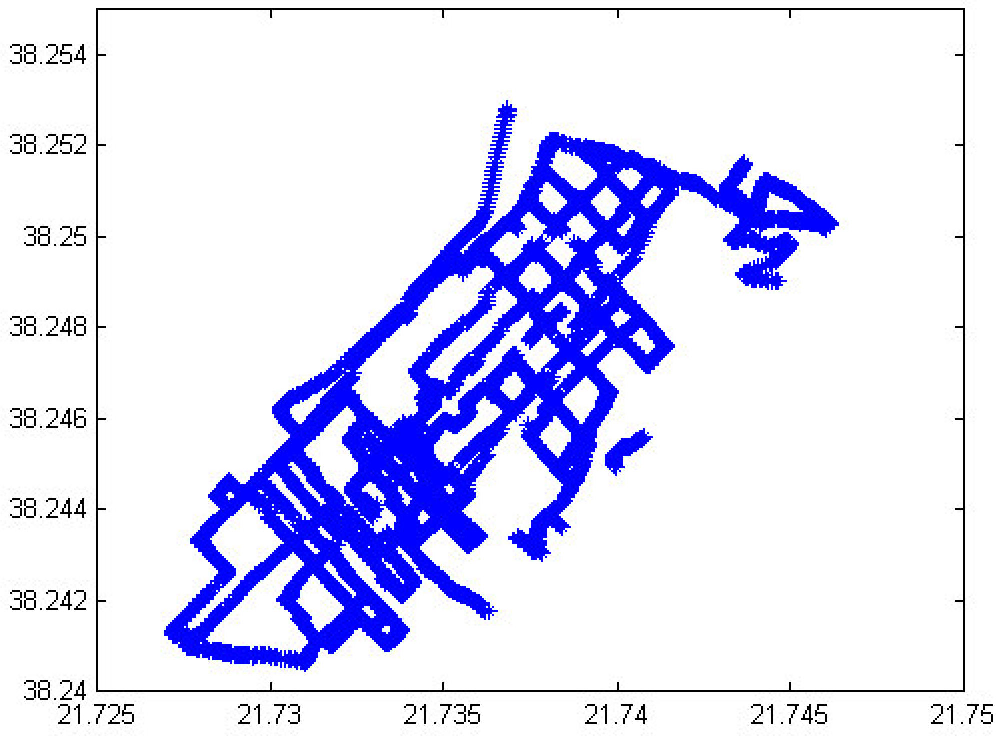

Figure 4.2: The examination area

On the x and y axis of the above figure, we can see the longitude and latitude coordinates as they are given by the GPS receiver.

Table 4.1 : The BSs and cells of the examination area.

CELL	BCCH	BSIC	Colour
Georgiou Sqr.	84	4	'*' black
	82	3	'*' magenta
	100	2	'*' blue
Hotel Adonis	93	4	'o' red
	99	2	'*' yellow
	88	1	'*' green
St. Andrew Church	79	1	'*' red

The next figure shows the examination area coloured with the corresponding classes measured in it.

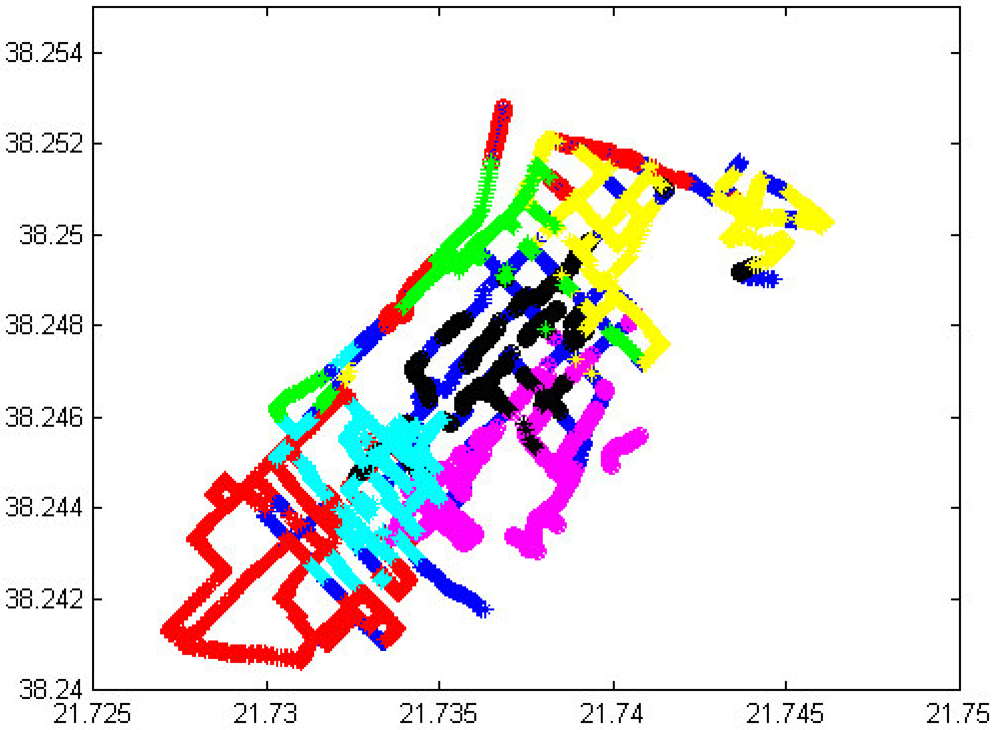

Figure 4.3: The examination area with classes. All dark blue sites are not part of any class for reasons explained in the following section.

4.3 Assumptions

Since real-time measurements are used, some assumptions were made before the use of the classifier, in order for the problem to be defined more accurately.

While for the small cells mentioned, the different measurements obtained would vary from 300-1300 for each cell, only a few points (10-35) have been measured to belong to unknown cells (cells whose id (BCCH-BSIC pair) was not given by the GSM provider). These points may have been served at that particular instant by BSs from Aitoloakarnania e.g. and have been extracted from the input data set.

- Only the macro-small cells layer of the multilayer structure of GSM is being examined, and thus three micro cells that have been found (not shown in the figure 4.3) have been treated this way:

- The measurements whose first neighbor BS (N1) is one of the macro-small cells have been extracted from the micro cell data set and have been added to the measurements of the corresponding macro-small cell.

- The rest of the micro cell measurements whose first neighbor is one of the micro cells as well, have not been taken in mind at all.

- Correspondingly, measurements that have been found to belong to the only one umbrella cell, have been extracted from the input pattern.

- Due to the time-varying wireless environment, where the wave propagation is affected by a variety of factors, not easily determined, as it has already been shown, there are points that have been measured to belong to a certain cell eventhough geographically they lie quite far from the majority of the measurements referring to that cell. These points, observed in groups of 3-5, have been treated as outliers and have been extracted from the input pattern.

Notice: This has happened only for the groups of 3-5 points and never else.

- The GPS error is considered very small since the car has been traveling at low speed and has been forced to stop quite often due to the existence of quite many lightening signs in the main streets of Patras.

- In the case of more than one measurements for the same site (stopped car), the latitude-longitude pair has been kept only once in the final data set where the maximum received signal strength has been replaced by the average value of all the receiving signal strengths from the serving BS at that site.

As it has been stated before, the automatic recognition of the cell/sector a randomly selected site (latitude, longitude, Pmax) belongs to, has been considered as a pattern classification problem.

4.4 Neural network simulation

Trajan software package has been selected as a tool for simulation. Trajan supports the most practical types of neural networks known for real-world problems-solving today, and includes the latest state-of-the-art techniques for fast training, automatic design and variable selection. It uses genetic algorithms to search for the most appropriate neural network.

The neural network that has been proved to give better results in our case is a multilayer perceptron trained with the back propagation and conjugate gradient algorithms. The back propagation algorithm was described in chapter2.

Conjugate gradient descent is an advanced method of training multilayer perceptrons. It is recommended for any network with a large number of weights (more than a few hundred) and/or multiple output units. Conjugate gradient descent is a batch update algorithm: whereas back propagation adjusts the network weights after each case, conjugate gradient descent works out the average gradient of the error surface across all cases, before updating the weights once at the end of the epoch. The algorithm assumes that the error surface is quadratic, which is not always true. If the algorithm discovers that the current line search direction isn't actually downhill, it simply calculates the line of steepest descent and restarts the search in that direction. Once a point close to a minimum is found, the quadratic assumption holds true and the minimum can be located very quickly.

4.5 Preprocessing and training

There have been two approaches in an effort to find the best performing MLP network for the cell-area determination problem.

Preprocessing Techniques

In half of the classes that were finally kept for the classification there were almost double measurements than those for the rest of them. In order to have almost equal amount of measurements for all classes, and since the distance between successive points of measurement is very small, every 2^{nd} or 3^{rd} sample has been kept, so as to have:

Table 4.2: Final data that were introduced to the MLP network.

Classes	BCCH	BSIC	Initial samples	Choosing procedure	Samples kept
Class1	100	2	885	Every 2	443
Class2	79	1	1318	Every 3	440
Class3	82	3	806	Every 2	403
Class4	84	4	336	As it is	336
Class5	88	1	327	As it is	327
Class6	93	4	269	As it is	267
Class7	99	2	868	Every 2	434
TOTAL					2652

In both approaches, latitude and longitude were transformed from degrees to latitudinal and longitudinal distance from a reference site (called Akti Dymeon), which lies southern-western from all the measurements. The reason that a southern-western site has been chosen is that this way the calculated distances would all be positive numbers. The transformation algorithm used is an empirical one, which has been proven to give small deviation from the true values.

<u>Transformation Algorithm</u>

1 degree latitude (North) = 111109 m   1 degree longitude (East) = 90381 m

1 degree = 60 min = 3600 sec

If (x_1, y_1) are the coordinates in degrees of a measurement site and (x_r, y_r) are those of the reference site, then :

Lattitudinal distance between sites: $111109 * |x_1 - x_r|$, in meters

Longitudinal distance between sites: $90381 * |y_1 - y_r|$, in meters

The values of Preceived were in the range [-95,-47]. In order to have all the inputs of the NN positive Preceived was shifted in the range [0, 48].

Normalization/Denormalization

Sometimes, it is critical for proper operation of the neural network to provide some external normalization of the values presented to it.

The objective of data normalization is to ensure that the statistical distribution of values for each input and output is roughly uniform. The network can accommodate some range in values and adjust parameters to compensate. If this is not done and an input with a normal distribution and a small variance is used, then the net will only see a small number of occurrences of facts away from the central tendency.
For example, the neural network is expected to fail badly if it is presented with two inputs (latitudinal, longitudinal distances) in the range [300, 1800] metres and the third (Preceived) in the range [0,48] dBms. The overall constants can't be changed fast enough to prevent the network from effectively ignoring the P input.

There's no given prescription for scaling the inputs but, usually it is preferred to be scaled to match the range of the input neurons, typically [-1,1] and [0,1]. The scaling function involves finding the minimum and maximum values in the data set, setting these equal to the lower and upper values of the desired range, and scaling all intervening values based on the proportional change. This method scales input data into the appropriate range but does not increase its uniformity.
A different normalization method normalizes the mean and standard deviation of the training set. Actually, it normalizes the inputs and targets so that they will have zero mean and unity standard deviation.

It is possible either to normalize both the inputs and targets of the network, or just the inputs. In our example, only the inputs have been normalized since the outputs are nominals indicating the class to which each input vector is assigned.
If the training data set is preprocessed this way, then whenever the trained network is used with new inputs, they should be preprocessed with the minimums and maximums, in the first case, and the means and standard deviations, in the second case, that were computed for the training set.

In this example, all values of latitudinal distance and longitudinal distance have been scaled by dividing with 100. The MLP network has been tried to be trained with the values of all three features being normalized, sequentially, in the ranges:[-1,1], [0,1] and by using (mean,std) normalization.
In the last case, the training data have been normalized using (mean, std) normalization and the resulting samples were shuffled. This means that the order of the presentation of the cases was changed within each epoch. Although this procedure adds some noise to the training process (the error may oscillate slightly), the algorithm is less likely to get stuck, and overall performance is usually improved [5].

The simulation

There have been two approaches to find the best performance of the MLP network for the particular case. The total data set was divided into three parts: the training set, the verification set and the test set. The verification set has been used in order for the network to avoid overfitting.

Verification set:
Without cross verification, a network with a large number of weights can overfit the training data - learning, as it were, the noise present in the data rather than the underlying structure. The ability of a network not only to learn the training data, but to perform well on previously-unseen data, is known as generalization.
The network can be checked if it generalizes properly by observing whether the verification error is reasonably low. The network is trained using the training set, but is also tested after each epoch using the verification set.
There have been found MLP networks with one or two hidden layers, performing well in our occasion. The logistic function was used as the activation function in all layers of the network.

TrainingError, VerificationError and TestError
The error when the network is executed on the training, verification and test subsets respectively. This is the root mean square (RMS) of the errors on each individual case, where the error on each individual case is measured by the network's error function. The training algorithms attempt to minimise the training error. The test error gives a final check of the performance of the network.
The training error is almost entirely irrelevant as an indicator of the ability of the model to make predictions given new data, while the test error can be used to diagnose training problems.
It is quite common for the training error to be much lower than the other two. If this occurs, it is preferred to select networks with less hidden units, as these are less prone to over-fitting.
If the verification error is much lower than the test error, that is of much concern. By conducting a large number of experiments, it may be possible to get a very low verification error "by luck." This is a particular danger if we had relatively few cases. However, since we have an adequate number of measurements for the tested area, a much low verification error could be due to unrepresentative distribution of cases. To test for this occasion, in the first approach, I have tried shuffling the assignment of the training, verification and test cases. In contrast, if the verification and test error figures are very close together, this is strong empirical evidence that the network has learned to generalise reliably.

TrainingPerformance, VerificationPerformance and TestPerformance
These figures show the performance of the network. The meaning of the measurement depends on the nature of the network.
In our example, where the network performs classification, the performance measure indicates the proportion of cases which are correctly classified.

Two approaches have been implemented in order to find the best performing NN.

1st approach
In the first approach all seven classes have been presented in one MLP network.
The best network obtained has 2 hidden layers of 38 hidden units each. The verification error is not much larger than the training error, something that proves good generalization. The overall performance is 93.085% while the overall RMS error is 0.1289. The training set contains 1900 cases, the verification set 376 cases and the test set 376 cases.

In table 4.3, the best networks obtained for different kinds of normalization of the input data set, are presented.

Table 4.3: Different normalization techniques used for the MLP, 2-layer training.

No of runs	**Normalization used**	**Best Network**	**Performance**	**Training Error**	**Verification Error**	**Testing Error**
1st	Without normalization	25:25	88.56%	0.1378	0.1526	0.144
2nd	Normalized [0,1]	25:24	89,62%	0.1427	0.1568	0.1617
3rd	Normalized [-1,1]	16:16	89.89%	0.1398	0.1483	0.1478
4th	Normalized (mean,std)	16:15	90,42%	0.1496	0.1518	0.1681
5th	Normalized (mean,std) +shuffled samples	38:38	93%	0.1	0.1289	0.16

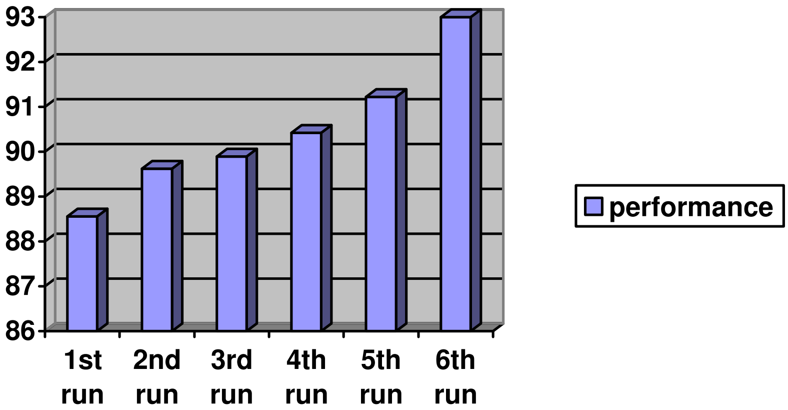

Figure 4.4: The performances obtained in the six different runs of the MLP network with the CG algorithm. The 6th run gave the best overall performance.

2nd approach

In an effort to improve the overall performance, the use of a modular neural network has been tried. The first module is an MLP network with 2 hidden layers of 16 hidden units each. The input data set contains samples from the following 4 classeis:

Table 4.4: Classes for the first NN of the second approach

Class-name	**BCCH**	**BSIC**	**Choosing procedure**	**Samples Kept**
Class3- magenta	82	3	As it is	806
Class7- yellow	99	2	As it is	868
Class2-red	79	1	Every 2 +200 of the rest	859
other	*100*	2	Every 2	*443*
	84	*4*	From the boundaries with classes 3, 7 and 2	*146*
	93	*4*	Every 2	*134*
	88	*1*	From the boundaries with classes 7 and 2	*164*
	rest			*97*
	Total other			984
TOTAL				3517

In the above table the class 'other' contains input cases (vectors) which represent the area among the first three classes of the table and the rest of the examination area. They are selected measurements from the next four classes of the second MLP network. The 'rest' part refers to the sites among the classes: 82-3, 99-2 and 88-1 (black, yellow and green, respectively).

From the 3517 samples, 2300 have been chosen to be the training set, 608 the verification set and 609 the test set. The only preprocessing applied was the division of lattitudinal and longitudinal distances by 100 and the shifting of Pmean at [0,47.75]. The best performance found for this network is 93.6% and the overall RMS error is 0.1588.

Table 4.5: *Performance of various neural networks*

Type of NN	**Hidden layers**	**Training Error**	**Verification Error**	**Test Error**	**Training Algorithm**
RBF	5:0	0.2853025	0.2863851	0.2881996	KM,KN
MLP	3:3	0.2163339	0.2229124	0.220004	BP50,CG470b
RBF	96:0	0.1868095	0.1970553	0.1968252	KM,KN
RBF	111:0	0.1817744	0.1930616	0.1936773	KM,KN
MLP	12:0	0.1632393	0.1736049	0.1763741	BP50,CG700b
MLP	14:9	0.1602559	0.1693025	0.1677691	BP50,CG351b
MLP	16:16	0.1397165	0.1546856	0.1537884	BP50,CG601b
MLP	38:38	0.1	0.1289	0.16	BP50,CG50

KM: K-means algorithm
KN: K-nearest neighbor
BPx: Back Propagation with x epochs
CGx: Conjugate Gradient Descent with x epochs

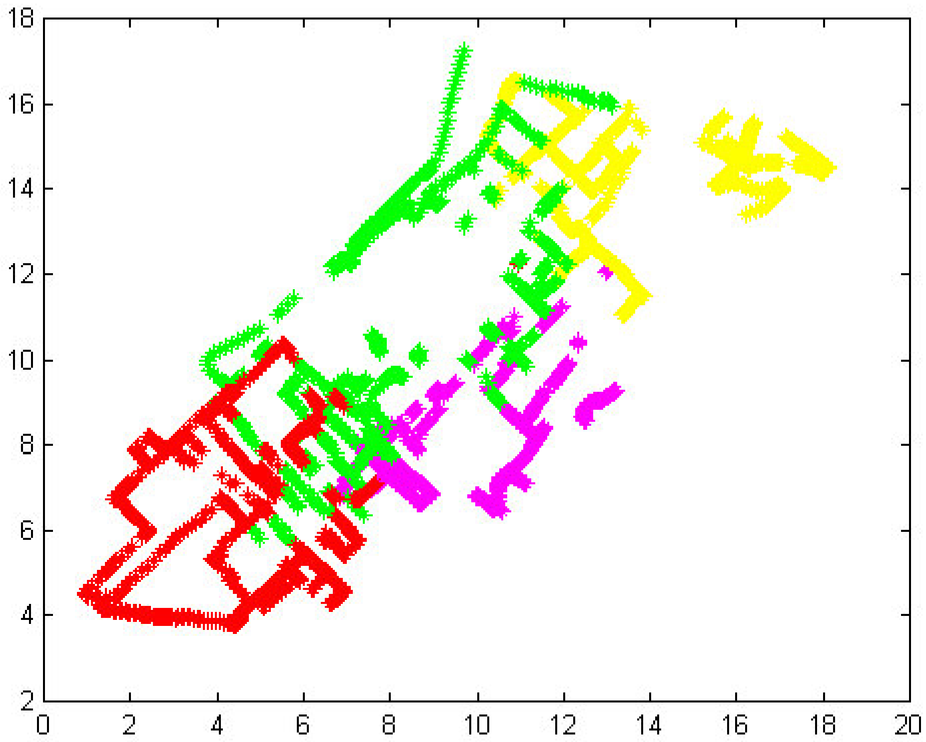

Figure 4.5: Classes in the first NN. The 'vother' class is shown in green

The second module is an MLP network as well, with 2 hidden layers of 16 hidden units and 15 hidden units each. The input data set for the second MLP contains samples from the rest of the classes.

Table 4.6: Classes for the second NN

Class-name	BCCH	BSIC	Choosing procedure	Samples Kept
Class4 – black	84	4	As it is	336
Class5 - green	88	1	As it is	327
Class6 – red ‘o’	93	4	As it is	267
Class1 - blue	100	2	Every 2	443
TOTAL				1373

From the 1373 samples, 916 samples have been used as the training set, 229 as the verification set and 230 as the test set. The inputs of this network have also been normalized to [-1,1].
The best performance found for this network was 98.68% and the overall RMS error is 0.077.

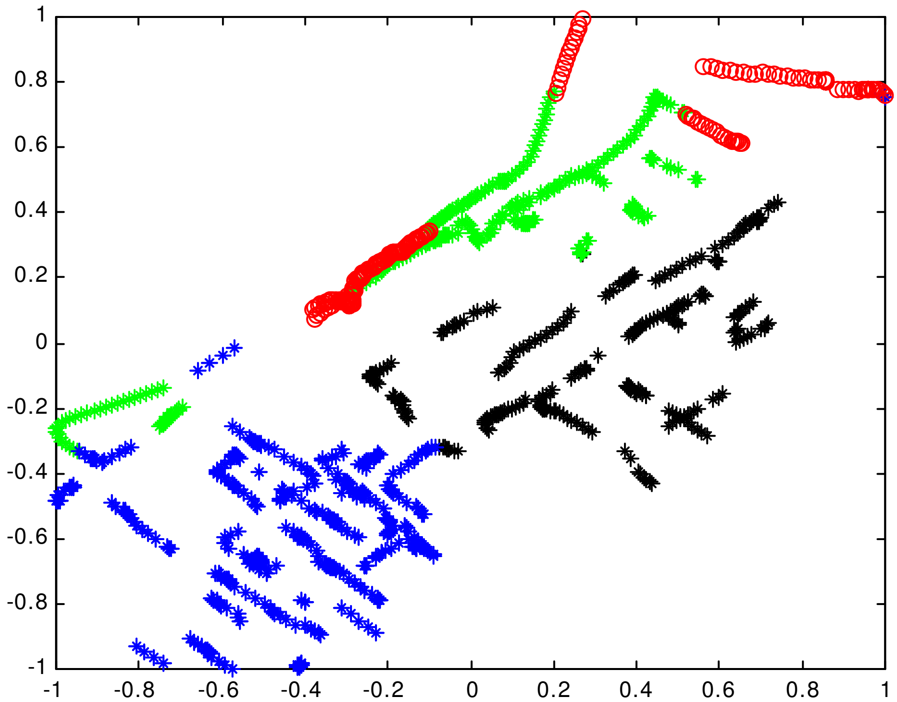

Figure 4.6: Classes in the second NN.

A new unknown input case is firstly presented to the first network and if it is decided that this case belongs to class ‘other’, then it is forwarded to the second network, which decides to which one from the ‘other’ classes it belongs to.

The two modules are less complex, than the MLP network used in the first approach since they have 16:16 hidden units each. The lower the complexity of an MLP network is the best it is able to generalize correctly. That's why the NN of the first approach eventhough it has lower overall RMS error, it has lower performance than the first NN of the second approach. On the other hand, the verification error, in the first NN of the second approach, is closer to the test error than is the verification error of the NN of the first approach to the test error. This is another sign that the second approach has given a model that generalizes better.

Table 4.7: Results for the best NNs found in the second approach

NNs	Hidden Layers	Training Error	Verification Error	Testing Error
1st NN	16:16	0.16	0.1588	0.18
2nd NN	16:15	0.066	0.077	0.11

Conclusions

In this dissertation the cell coverage estimation problem has been treated as a classification problem. Neural classifiers have been trained with real field strength measurements to be able to decide the cell each location, in the map of the examined area, belongs at. The cell coverage map that is created this way, can be very useful to the radio network designer as it is explained in the first chapter. The major advantage of the proposed method is that it is based on real on-site field strength measurements. The MLP neural network trained with the back propagation and conjugate gradient algorithms, was found to give the best performance. The trained neural networks can be further used to indicate the exact positions for the installation of new base stations in order to increase the performance of the cellular system under examination.

References

[1] "*Wireless Communications, Principles and Practice",* by Theodore S. Rappaport, Prentice Hall, 1996

[2] "*Coverage estimation for mobile cellular network from signal strength measurements*", Kangalu Manoj, dissertation for PhD in Electrical Engineering, University of Texas at Dallas, April 1999.

[3] "*Applying Neural Networks*", Judi Stader, The University of Edinburgh, August 1992

[4] "*Neural Networks, A Comprehensive Foundation"*, by Simon haykin, Prentice Hall Second Edition 1999

[5] "*Introduction to Trajan Neural Networks"*, Trajan Software 1999

[6] "*Adaptiv Multipath Detection with Neural Networks in Direct Sequence Spread Spectrum Systems*", Daniel Lutziger and Urs A. Mueller Electronics Laboratory, Swiss Federal Institute of Technology CH-8092 Zurich, Switzerland

[7] "*Signal Estimation with Neural Networks for multipath mobile communications*",T.L. Hemminger, *IEEE International Conference on Neural Networks,* Houston, TX June 1997.

[8]" *InMultiuser/Multisubchannel DetectionBased on Recurrent Neural Network Structures for Linear Modulation Schemes with General Complex-valued Symbol Alphabet*", Ch. Sgraja, W. G. Teich, A. Engelhart, J. Lindner, Proc. COST262 Workshop Multiuser Detection in Spread Spectrum Communications, Reisensburg/Germany 2001, pp. 45-52

[9] "*Multiuser Detection for DS-CDMA Transmission Systems using Neural Network Techniques*", Ninth DSP (DSP 2000) Workshop+First Signal Processing Education Workshop, Two Workshops in One, October 15 - 18, 2000, Hunt, Texas

[10] "*Application of a Recurrent Neural Network to Space Deversity in SDMA and CDMA Mobile Communication Systems*" , Neural Computing & Applications , Springer-Verlag London Ltd, Volume 10, Number 2 , May 2001, pp. 136 – 147

[11] " Adaptive Channel Equalization Using Neural Networks", Amita Pugalia, Final report on project EE359 in Department of Electrical Engineering , University of Stanford, Fall 2002

[12] "*Non-linear channel equalization using minimal radial basis function neural networks*", P. Chandra Kumar, P. Sarachandran and N. Sundararajan, ICASSP 98

[13] "*Development of a novel digital communication channel equalizer using real-time implementable multilayer neural structure*", Jitendriya K. Satapathy, Ganapati Panda and Susmita Das

[14] "*Equalization of Satellite Mobile Communication Channels Using Combined Self-Organizing Maps and RBF Networks*", S Bouchired, M. Ibnkahla, D. Roviras, F. Castanie, International Conference on Acoustics, Speech, and Signal Processing, May 1998, Seattle, Washington, USA

[15] "*A comparison between recurrent neural network architectures for digital equalization*", Ortiz-Fuentes J., Forcada M. (Univ. d'Alacant, Spain.),ICASSP International Conference on Acoustics, Speech, and Signal Processing, April 21-24, 1997, Munich, Germany, *Vol. 4, pp. 3281*

[16] "*Application of Antenna Arrays to Mobile Communications, Part I : Performance Improvement, Feasibility and System Considerations*", LAL C Godara, Proccedings of the IEEE, vol. 85, no 8, July 1997

[17] "*Applications of Antenna Arrays to Mobile Communications Part II: Beam-Forming and Direction Of Arrival Considerations*", LAL C Godara, Proccedings of the IEEE, vol. 85, no 8, Aug 1997

[18] "*Applications of Neural Networks in Electromagnetis*", C. Christodoulou and M. Georgiopoulos, Artech House, Nov. 2000

[19] "*A Fast Neural Beamformer For Antenna Arrays*", K.-L. Du, K.K.M. Cheng, and M.N.S. Swamy, Proc. IEEE International Conference on Communications, New York City, NY, 28, April - 2 May, 2002, pp. 139-144

[20] "*Adaptive Radio Resource Management for GSM/GPRS Networks*", K. Murray and D. Pesch , First Joint IEI/IEE Symposium on Telecommunications Systems Research , 27th November 2001, Dublin

[21] "*Models and Solution Techniques for Frequency Assignment Problems*", Aardel,Karen, Hoesel,Stan,van Koster,Arie Mannino,Carlo Sassano,Antonio, METEOR The Faculty's Graduate research school METEOR (Maastricht Economic Research School on Technology and Organisation), University of Maastricht, 2002

[22] "Channel assignment for cellular radio using neural networks", D. Kunz, IEEE Transactions on Vehicular Technology, 40 1991, 188-193

[23] "*A neural network parallel algorithm for channel assignment problems in cellular radio networks*", N. Funabiki, Y Takefuji, IEEE Transactions on Vehicular Technology 41 1992, 430-437

[24] "*A synamic channel allocation technique based on hopfield neural networks*", E. Del Re, R. Fantacci and L. Ronga, IEEE Transactions on Vehicular Technology 45 1996, 26-32

[25] "*Cellular radio channel assignment using a modified hopfield network*", J-S Kim, S. Park, P. Dowd and N. Nasrabadi, IEEE Transactions on Vehicular Technology 46 1996, no. 4, 957-967

[26] "*Solving the channel assignment problem using neural networks and genetic algorithms*",Telmo R. C. Fernandes and Henrique J. A. da Silva,Proceedings of 3rd Conference on Telecommunications (conftele'2001), Figueira da Foz, Portugal, 23-24 April, 2001, Instituto de Telecomunicações, pp. 297-301

[27] "*Dymanic Channel Assignment in Cellular Mobile Radio Systems using Hopfield Neural Networks*", Dushy Tissainayagam, December 1996, The University of Melbourne**,** Department of Electrical & Electronic Engineering, Seminars

[28] "*Cellular Channel Assignment Comparing and Simplifying Heuristics*", Roberto Battiti, Alan A. Bertossi, Mauro Brunato, IEEE/ACM Workshop Dial M for Mobility, (Discrete algorithms and methods for mobile computing and communications), Budapest, Oct 1, 1997.

[29] "*Channel Assignment in Mobile Networks using Neural Networks*", Fernandes, T.R.; H. J. A. Silva; Proc Matemαtica em Telecomunicaηυes - Que Problemas? - MTQP , Coimbra , Portugal , Vol. 01 , pp. 20 - 23 , September , 1998 .

[30] "*User Profiling and classification for fraud detection in mobile communication networks*", Jaakko Hollmen, Dissertation for the degree of Doctor of Science in Technology, Helsinki University of Technology, 2000

[31] "*The detection of fraud in mobile networks*", Barson, S. Field, N. Davey G. McAskie and R. Frank, Neural Network World 6(4), 477-484

[32]"*Fraud detection in mobile communication networks using supervised neural networks*", Moreau Y., and J. Vandewalle, Proceedings of SNN '97, Europe's Best Neural Networks Practice. World Scientific.

[33] "*Frameworks for fraud detection in mobile telecommunications networks*", Burge, P. and J. Shawe-Taylor, In Proceedings of the fourth annual mobile and personal communications seminar, University of Limerick, 1996

[34]"*Detecting cellular fraud using adaptive prototypes*", Burge, P. and J. Shawe-Taylor, in Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection & Risk Management, pp. 9-13, AAAI Press.

[35]"*Neural-network detection of fraudulent calling card patterns*" , Connor, J. T. L. R. Brothers and J. Alspector, In proceedings of the International Workshop on Applications of Neural Networks to Telecommunications, 2,pp.363-370. Laurence Erlbaum Associates.

[36] "*Fraud Detection in Communications Networks using Neural and Probabilistic Methods*", Olli Simula, Esa Alhoniemi, Jaakko Hollmén, and Juha Vesanto, Proceedings of the 1998 IEEE International Conference in Acoustics, Speech and Signal Processing (ICASSP'98), Volume 2, pages 1241-1244, May 1998.

[37]"*BRUTUS - A Hybrid Detection Tool*", .P.Burge, J.Shawe-Taylor, Y.Moreau, H.Verrelst, C.Stoermann, P.Gosset, ACTS Mobile Summit 97, Proceedings of ACTS Mobile Summit 97, pp. 722-727, Aalborg, 07/10/97.

[38] "*A rule-based and neural network system for fraud detection in mobile communications*", Herman Verrelst, Ellen Lerouge, Yves Moreau, Joos Vandewalle, Christof Stormann, Peter Burge, ACTS Mobile Communications Summit 1999

[39]"*Behavior-Based Intrusion Detection in Mobile Phone Systems*", Azzedine Boukerche, Mirela Sechi Moretti Annoni Notare, Journal of Parallel and Distributed Computing 62(9):1476-1490 (2002)

[40]"*Applications of general regression neural networks for path loss prediction*" , Ileana Popescu, Athanasios Kanatas, Philip Constantinou, Ioan Nafornita, International Workshop On rends and Recent Achievements in Information Technology 16 - 18 May 2002, Cluj Napoca, Romania

[41] "*A Recursive Model For The Field Strength Prediction With Neural Networks*", G. Wölfle and F.M. Landstorfer,10th IEE International Conference on Antennas and Propagation (ICAP), Edinburgh, U.K., pp. 2.174-2.177, Apr. 1997.

[42] "*Modern Approaches in Modeling of Mobile Radio Systems Propagation Environment*", Aleksandar Neskovic, Natasa Neskovic, and George Paunovic, IEEE Communications Surveys & Tutorials, Third Quarter 2000, Vol. 3, No. 3

[43] "*Prediction-based location management using multilayer neural networks*",BP V IJAY K UMAR AND P. V ENKATARAM, Journal of the Indian Institute of science, vol. 200201, 2002

[44]" *Neural Networks for Location Prediction in Mobile Networks*", Jens Biesterfeld, Elyes Ennigrou, Klaus Jobmann Biesterfeld, J., Ennigrou, E., and Jobmann, K., Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications 1997 (IWANNT'97), Lawrence Erlbaum Associates, Melbourne, Australia.

[45]"*Mobile Station Positioning Using GSM Cellular Phone and Artificial Neural Networks"*, Salcic Zoran and Chan Edwin, Wireless Personal Communications: An International Journal Volume 14 , Issue 3 (September 2000), Pages: 235 - 254

[46] "*Motion Prediction in mobile communication systems from a historical and geographical perspective*", Zhiwei Cen, Hong Chen Xiaomei Liu, CSE812 Group Project ReportComputer Science and Engineering Department, Michigan State University, Fall 2002,

[47] "*An integrated neuro configuration management approach for hybrid mobile communication networks*", Mostafa M. El-Said, Anup Kumar, Adel S. Elmaghraby , University of Louisville, *Seventh International Symposium on Computers and Communications* (ISCC'02) July 01 - 04, 2002, Taormina-Giardini Naxos, Italy p. 315 -320

[48] "*Neural analysis of mobile radio access network*", Kimmo Raivio,Olli Simula, Jaana Laiho, Proceedings of the 2001 IEEE International Conference on Data Mining, 29 November - 2 December 2001, San Jose, California, USA, 457-466

[49] "*Radio Network Planning with neural networks*", Thomas Binzer and Friedrich M. Landstorfer, Institut of Radio Frequency Technology, University of Stuttgard

[50] "*Wireless Communications, Principles and Practice",* by Theodore S. Rappaport, Prentice Hall, 1996

[51] "*Cell Planning for Wireless Communications*", Manuel F. Catedra and Jesus Perez-Arriaga , Artech House, 1999

[52] *'Pattern Recognition Techniques in Handoff and Service Area Determination'*, by H.Maturino-Lozoya, D.Munoz-Rodriguez and H.Tawfik, Proc 44th VTC vol1 1994 pp 96-100.

[53] *'Using RBF algorithm for Landcover or Clutter Classification for Mobile Planning'*, by Liyang Zhang, Boon-Hee Soong, Anthony Tung-Shuen Ho and Jin-Teong Ong, VTC 2000