ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**Τομέας Ηλεκτρικών Βιομηχανικών Διατάξεων και Συστημάτων Αποφάσεων**

**Μονάδα Συστημάτων Πρόβλεψης Και Προοπτικής**

**Διπλωματική Εργασία του**

**Γιαννέλου Κωνσταντίνου**

# Πληροφοριακό Σύστημα Υποστήριξης Κριτικών Επεμβάσεων Σε Στατιστικές Προβλέψεις

Καθηγητής:: Prof. Β.Ασημακόπουλος
Επιβλέπων: Dr. Κ.Νικολόπουλος

Αθήνα, Ιούλιος 2004

ΜΟΝΑΔΑ ΣΥΣΤΗΜΑΤΩΝ ΠΡΟΒΛΕΨΗΣ ΚΑΙ ΠΡΟΟΠΤΙΚΗΣ

# Πληροφοριακό Σύστημα Υποστήριξης Κρπικών Επεμβάσεων Σε Σταπιστικές Προβλέψεις

**Γιαννέλος Κωνσταντίνος**

Καθηγητής:: Prof. Β.Ασημακόπουλος
Επιβλέπων: Dr. Κ.Νικολόπουλος

Αθήνα, Ιούλιος 2004

# Πρόλογος

Η διπλωματική αυτή εργασία εκπονήθηκε στα πλαίσια των ερευνητικών δραστηριοτήτων της Μονάδας Συστημάτων Πρόβλεψης και Προοπτικής κατά το χειμερινό εξάμηνο του έτους 2003 – 2004 (Δεκέμβριος 2003 – Ιούνιος 2004). Η μονάδα υπάγεται στον Τομέα Ηλεκτρικών Βιομηχανικών Διατάξεων και Συστημάτων Αποφάσεων, της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, του Εθνικού Μετσοβίου Πολυτεχνείου.

Η Μονάδα Συστημάτων Πρόβλεψης και Προοπτικής λειτουργεί υπό την επίβλεψη του καθηγητή κ. Βασίλη Ασημακόπουλου και απαρτίζεται από άρτια καταρτισμένο επιστημονικό προσωπικό. Στα πλαίσια της διπλωματικής αυτής εργασίας, θα ήθελα να ευχαριστήσω όλα τα μέλη του εργαστηρίου για την βοήθεια που μου παρείχαν σε θέματα τεχνικής φύσεως και πολύ περισσότερο για το φιλικό περιβάλλον εργασίας μέσα στο οποίο εργάστηκα. Ιδιαίτερα όμως θα ήθελα να ευχαριστήσω τον κ. Dr. Κωνσταντίνο Νικολόπουλο, ο οποίος είχε αναλάβει την επίβλεψη του όλου έργου, για την άριστη συνεργασία μαζί του σε όλη τη διάρκεια του εξαμήνου. Οι συμβουλές του και η καθοδήγησή του με βοήθησαν να κατανοήσω από την αρχή σε βάθος το αντικείμενο εργασίας μου, καθώς και να ξεπεράσω όποιες τεχνικές δυσκολίες παρουσιάστηκαν.

Τέλος θα ήθελα να ευχαριστήσω τον καθηγητή κ. Βασίλη Ασημακόπουλο για την ευκαιρία που μου έδωσε να ασχοληθώ με το συγκεκριμένο θέμα και να διευρύνω το γνωστικό μου πεδίο με ένα εντελώς νέο για μένα επιστημονικό αντικείμενο.

Αθήνα, Ιούλιος 2004

# Περιεχόμενα

## Κεφάλαιο 7 : Οδηγίες Εγκατάστασης



## Κεφάλαιο 8 : Βιβλιογραφία

Κεφάλαιο 1

# ΕΙΣΑΓΩΓΗ

## 1.1 Εισαγωγή

Οι συνήθεις στατιστικές μέθοδοι μας επιτρέπουν να επεκτείνουμε αναγνωρισμένα πρότυπα και / ή υπάρχουσες σχέσεις με σκοπό να προβλέψουμε την συνέχεια τους, υποθέτοντας ότι αυτά τα πρότυπα ή σχέσεις δεν θα αλλάξουν για τον χρονικό ορίζοντα της πρόβλεψης. Την ίδια στιγμή, επειδή είναι δυνατόν να συμβούν αλλαγές και κατά πάσα πιθανότητα θα συμβούν, πρέπει να τις εντοπίζουμε όσο το δυνατόν νωρίτερα ώστε να αποφεύγουμε τα μεγάλα και συνήθως δαπανηρά λάθη στις προβλέψεις.

Ωστόσο, όταν εντοπιστούν οι αλλαγές ή αν γνωρίζουμε πότε θα συμβούν, η ανθρώπινη κρίση είναι η μόνη εφαρμόσιμη εναλλακτική για την πρόβλεψη τόσο του μεγέθους των αλλαγών όσο και των επιπτώσεων που αυτές θα επιφέρουν στην πρόβλεψη. Η ανθρώπινη κρίση είναι επίσης απαραίτητη στην ενσωμάτωση εσωτερικών πληροφοριών και γνώσης, με την εμπειρία των διευθυντικών στελεχών (managers) για το μέλλον. Πριν χρησιμοποιήσουμε την κρίση μας για να βελτιώσουμε τις προβλέψεις θα πρέπει να αναγνωρίσουμε τις μεροληψίες της και τους περιορισμούς της μαζί με τα μεγάλα πλεονεκτήματα της. Αν το κάνουμε αυτό, μας δίνεται η δυνατότητα να συνδυάσουμε τις πληροφορίες που λαμβάνουμε από τις στατιστικές προβλέψεις με αυτές που μας παρέχει η κρίση μας αποφεύγοντας όλα τα μειονεκτήματα.

## 1.2 Η ορθότητα των Κριτικών προβλέψεων

Αν και καθημερινά πραγματοποιούμε αναρίθμητες προβλέψεις, ξοδεύουμε ελάχιστη ενέργεια για την αποτίμηση τους και για την αναζήτηση τρόπων να βελτιώσουμε την ορθότητα τους. Ο λόγος είναι απλός: δεν θέλουμε να αναλάβουμε την ευθύνη στην περίπτωση που οι προβλέψεις μας είναι λανθασμένες. Ωστόσο, αν δεν ελέγξουμε την ορθότητα τους τότε είναι πολύ πιθανό ότι δεν θα καταφέρουμε να βελτιώσουμε την απόδοση μας όταν πραγματοποιούμε παρόμοιες προβλέψεις στο μέλλον. Εφόσον οι υποκειμενικές – κριτικές προβλέψεις είναι πολύ πιο κοινές από τις στατιστικές, όχι μόνο δεν μπορούμε να τις αγνοήσουμε αλλά πρέπει να δεχτούμε ότι δεν μπορούμε να αποφύγουμε εντελώς τα σφάλματα σε αυτές. Θα είναι πολύ καλύτερα αν αποδεχτούμε αυτά τα σφάλματα και προσπαθήσουμε να διδαχθούμε από αυτά έτσι ώστε να βελτιώσουμε την ικανότητα μας να προβλέπουμε με μεγαλύτερη ορθότητα.

Η ορθότητα των υποκειμενικών προβλέψεων είναι, κατά μέσο όρο, κατώτερη από αυτή των στατιστικών προβλέψεων. Ο λόγος που συμβαίνει αυτό είναι ότι η κρίση μας συχνά χαρακτηρίζεται από μεροληψίες και περιορισμούς. Θα δείξουμε παρακάτω ότι η εγκυρότητα των υποκειμενικών μας προβλέψεων μπορεί να βελτιωθεί με την χρήση κάποιων απλών στατιστικών μοντέλων. Το κόστος αυτών των μοντέλων είναι κατά πολύ μικρότερο από αυτό των ανθρώπων που πραγματοποιούν τις προβλέψεις. Η πρόκληση για αυτούς που κάνουν προβλέψεις είναι να αποφύγουν τους περιορισμούς και τις μεροληψίες τους, συνδυάζοντας τα καλύτερα στοιχεία της κρίσης τους με απλές στατιστικές μεθόδους.

## 1.3 Προβλέψεις μη επενδυτικού τύπου

Αν και δεν υπάρχουν άφθονα εμπειρικά δεδομένα σχετικά με κριτικές προβλέψεις για άλλους τομείς πέρα από τις επενδύσεις, το συμπέρασμα είναι παρόμοιο : η ακρίβεια αυτών των προβλέψεων είναι κατά μέσο όρο, μικρότερη από αυτή των στατιστικών προβλέψεων. Σε αυτό το τμήμα παρουσιάζονται στοιχεία προβλέψεων από πωλητές, διοίκηση (management) και ειδικούς.

### 1.3.1 Προβλέψεις πωλητών

Οι προβλέψεις των πωλητών ήταν κάποτε ιδιαίτερα δημοφιλείς καθώς, λόγω της καθημερινής συναναστροφής τους με τους πελάτες, οι πωλητές θεωρητικά είναι σε θέση να γνωρίζουν τις επερχόμενες αλλαγές στην αγορά. Η μελέτη των αποτελεσμάτων, όμως, δείχνει ότι στην πράξη οι προβλέψεις τους είναι πολύ λανθασμένες. Το χειρότερο είναι ότι οι προβλέψεις τους παρουσιάζουν μεγάλες διακυμάνσεις, ανάλογα με την διάθεση τους και ανάλογα με τις πωλήσεις που έχουν πετύχει. Επιπλέον, οι πωλητές συνήθως επιβραβεύονται αν επιτύχουν πωλήσεις μεγαλύτερες από κάποιον στόχο που έχει τεθεί (ο στόχος καθορίζεται με κάποιου είδους πρόβλεψη). Έτσι τους συμφέρει να θέτουν χαμηλούς στόχους, τους οποίους μπορούν να επιτύχουν σχετικά εύκολα ώστε να εισπράττουν το bonus. Την ίδια στιγμή, οι διευθυντές πωλήσεων (sales managers) επιθυμούν να θέτουν υψηλούς στόχους ώστε να έχουν κίνητρο οι πωλητές. Έτσι, προσαρμόζουν την εκτίμηση πωλήσεων προς τα επάνω, δημιουργώντας σύγχυση ανάμεσα στις αντικειμενικές προβλέψεις και την επίτευξη επιθυμητών στόχων. Τέλος, οι πωλητές προσδιορίζουν τις παρούσες πωλήσεις στην περιφέρεια τους, παρά το πώς αυτές θα εξελιχθούν στο μέλλον όταν οι συνθήκες μπορεί να αλλάξουν. Για όλους αυτούς τους λόγους, οι κριτικές προβλέψεις τους δεν είναι ο καλύτερος τρόπος για να προσδιοριστούν οι μελλοντικές πωλήσεις μίας επιχείρησης.

### 1.3.2 Προβλέψεις της διοίκησης

Οι διευθυντές, σε αντίθεση με τους πωλητές, έχουν μία πιο γενική εικόνα της εταιρείας, της αγοράς αλλά και της οικονομίας. Ωστόσο, συχνά παρουσιάζονται ιδιαίτερα αισιόδοξοι για το μέλλον της εταιρείας ή του προϊόντος, για το οποίο είναι υπεύθυνοι. Για παράδειγμα, τα διευθυντικά στελέχη (managers) σπάνια προβλέπουν μειωμένες πωλήσεις ή ότι κάποιο προϊόν θα αποτύχει. Επίσης δεν είναι οι καταλληλότεροι για να εκτιμήσουν την απειλή από τους ανταγωνιστές ή τις συνέπειες της νέας τεχνολογίας στα προϊόντα τους. Έτσι, οι προβλέψεις τους, συνήθως, δεν είναι πιο έγκυρες από αυτές των στατιστικών μεθόδων, οι οποίες έχουν και ως μεγάλο πλεονέκτημα την αντικειμενικότητα. Αυτό δεν σημαίνει ότι τα διευθυντικά στελέχη (managers) και οι πωλητές δεν είναι ικανοί να πραγματοποιούν έγκυρες προβλέψεις ή ότι δεν κατέχουν χρήσιμες πληροφορίες, οι οποίες θα μπορούσαν να βελτιώσουν της ικανότητα της επιχείρησης να προβλέπει

μελλοντικές καταστάσεις. Απλώς είναι συχνά αισιόδοξοι στις προβλέψεις τους και δεν διαχωρίζουν εύκολα το προσωπικό τους συμφέρον από αυτό της εταιρείας.

### 1.3.3 Προβλέψεις ειδικών

Υπάρχουν σημαντικά δεδομένα που συγκρίνουν τις προβλέψεις «ειδικών» με αυτές των στατιστικών μεθόδων. Σχεδόν σε όλες τις περιπτώσεις στις οποίες τα δεδομένα μπορούν να ποσοτικοποιηθούν, οι προβλέψεις των μαθηματικών μοντέλων είναι ανώτερες από αυτές των ειδικών. Οι άνθρωποι, είμαστε κακοί στις προβλέψεις και συχνά για καλό λόγο (φανταστείτε τον manager ενός νέου προϊόντος να προβλέπει ότι το προϊόν θα αποτύχει). Η κρίση μας συχνά επηρεάζεται από μεροληψίες και από άλλους περιορισμούς που μειώνουν την εγκυρότητα των προβλέψεων μας. Αυτές οι μεροληψίες και οι περιορισμοί περιγράφονται στη συνέχεια.

## 1.4 Η φύση των κριτικών μεροληψιών και των περιορισμών

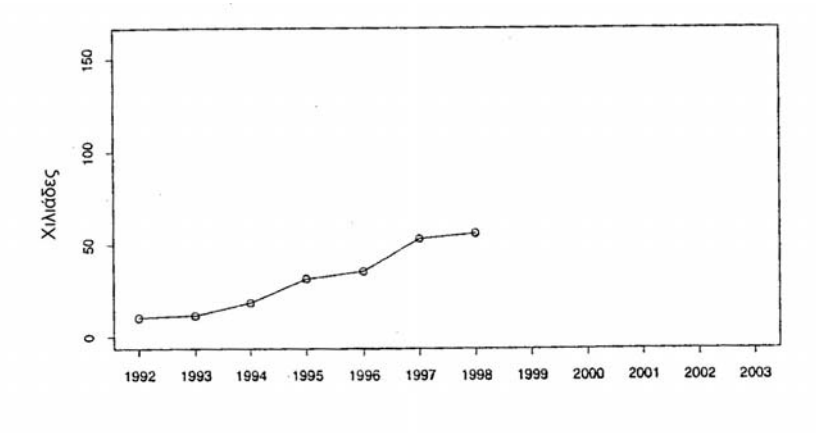
Οι περισσότεροι από εμάς γνωρίζουμε τους περιορισμούς της μνήμης μας. Γνωρίζουμε ότι δεν μπορούμε να θυμόμαστε τα πάντα και έτσι λαμβάνουμε τα μέτρα μας ώστε να αποφεύγουμε τις αρνητικές συνέπειες. Για παράδειγμά, χρησιμοποιούμε μια ατζέντα στην οποία γράφουμε με αλφαβητική σειρά τα ονόματα ανθρώπων και επιχειρήσεων μαζί με τα αντίστοιχα τηλέφωνα και τις διευθύνσεις ώστε να μπορούμε να τα βρούμε με ευκολία στο μέλλον. Κάνοντας αυτό δεν σημαίνει ότι η μνήμη μας είναι ανεπαρκής. Αντιθέτως, η ανθρώπινη μνήμη έχει τεράστια αξία. Όμως, η τεράστια χωρητικότητα της θα γέμιζε σε λίγες μέρες αν καταγράφαμε τα πάντα, χωρίς καμία διάκριση. Έτσι, είναι ιδιαίτερα σημαντικό να καθορίζουμε τι πρέπει να θυμόμαστε και τι όχι. Όσο περισσότερο απαλλάσσουμε τη μνήμη μας από το βάρος των άχρηστων πληροφοριών τόσο περισσότερο αυξάνεται η χωρητικότητα της και η ευκολία με την οποία ανακαλεί χρήσιμες πληροφορίες.

Ωστόσο, αν και αναγνωρίζουμε τις ανεπάρκειες και τους περιορισμούς της μνήμης μας και πράττουμε αναλόγως, σπάνια κάνουμε κάτι για να αντιμετωπίσουμε τις ανεπάρκειες της κρίσης μας, κυρίως γιατί δεν είμαστε πρόθυμοι να δεχτούμε ότι η κρίση μας μπορεί να είναι λανθασμένη και να έχει

μεροληψίες. Επειδή σπάνια παραδεχόμαστε ότι υπάρχουν κριτικές μεροληψίες είναι πολύ σημαντικό να τις ξεσκεπάσουμε: τα εμπειρικά δεδομένα αποδεικνύουν ξεκάθαρα την ύπαρξη τους και τις αρνητικές συνέπειες που έχουν. Οι μεροληψίες αυτές είναι αποτέλεσμα του τρόπου με τον οποίο λειτουργεί το μυαλό μας και προσπαθεί να ενοποιήσει ασυμβίβαστους στόχους.

## 1.5 Κριτικές μεροληψίες στις προβλέψεις

Υποθέστε ότι εργάζεστε για την "Elco Electronics" μία εταιρεία με περίπου 2000 εργαζόμενους, και είστε υπεύθυνος για τους εκτυπωτές λέιζερ. Αυτοί οι εκτυπωτές μπορούν επίσης να χρησιμοποιηθούν και ως φαξ, scanners και φωτοαντιγραφικά, και είναι υπεύθυνοι για το 40%, περίπου, των εσόδων της εταιρείας. Ο ανταγωνισμός είναι πολύ ισχυρός στην αγορά και στο εσωτερικό αλλά και από εισαγωγές κυρίως από τις χώρες του Ειρηνικού. Ως υπεύθυνος για το προϊόν πρέπει να αποφασίσετε πόσα κομμάτια από καθένα από τα 10 μοντέλα εκτυπωτών να κατασκευαστούν και να μεταφερθούν στους διανομείς, καθώς επίσης και αν πρέπει να γίνουν επενδύσεις σε νέο εργοστάσιο και καινούργιο εξοπλισμό. Για να ληφθούν αυτές οι αποφάσεις είναι απαραίτητο να γίνουν προβλέψεις.




**Σχήμα 1:** Πωλήσεις εκτυπωτών λέιζερ της Elco: 1992-1998

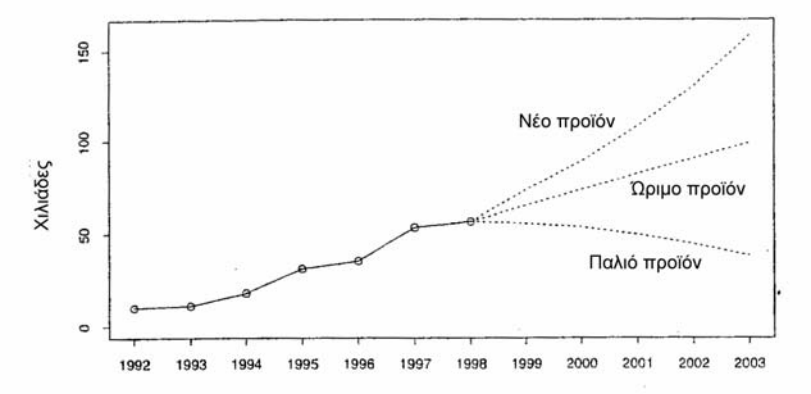
Στο σχήμα 1 φαίνονται οι πωλήσεις εκτυπωτών της εταιρείας από το 1992. Όπως φαίνεται υπάρχει μία αύξηση στις πωλήσεις μέχρι τώρα, όμως το μέλλον σας απασχολεί. Θα πρέπει να κάνετε πρόβλεψη πωλήσεων ( μαζί με πρόβλεψη τιμής, ανταγωνισμού κ.ο.κ.) για την επόμενη χρονιά και για πέντε χρόνια από τώρα. Ποια κατά, κατά τη γνώμη σας, θα είναι η πρόβλεψη για το 1999 και το 2003, συμπεριλαμβάνοντας ένα απαισιόδοξο σενάριο χαμηλών πωλήσεων και ένα αισιόδοξο, υψηλών πωλήσεων;

| **Έτος** | **Χαμηλή (απαισιόδοξη) πρόβλεψη** | **Πιο πιθανή πρόβλεψη** | **Υψηλή(αισιόδοξη) πρόβλεψη** |
|---|---|---|---|
| 1999 | _____ | _____ | _____ |
| 2003 | _____ | _____ | _____ |

**Πίνακας 1:** προβλέψτε τις πωλήσεις λέιζερ εκτυπωτών της Elco για το 1999 και 2003

Στην πραγματικότητα, δεν υπάρχει σωστή ή λάθος πρόβλεψη για τις πωλήσεις εκτυπωτών λέιζερ. Αν ζητηθεί από κάποιον τρίτο (όπως εσείς) να προβλέψει τις πωλήσεις της Elco, το πιο πιθανό είναι ότι θα προβλέψει μία ευθεία επέκταση από το παρελθόν στο μέλλον. Έτσι η πιο πιθανή πρόβλεψη, γύρω από την οποία θα βρίσκονται οι περισσότερες απαντήσεις, θα είναι περίπου 65.000 μονάδες για το 1999 και 99.000 μονάδες για το 2003. Αυτό γιατί θεωρείτε πως ότι συνέβη στο παρελθόν θα συνεχίσει να συμβαίνει και στο μέλλον. Δηλαδή, οι διάφοροι παράγοντες που είτε αυξάνουν είτε μειώνουν τις πωλήσεις θα ακυρώσουν ο ένας τον άλλο, καθώς η Elco και οι ανταγωνιστές της θα κάνουν κινήσεις οι οποίες θα εξουδετερώνουν με επιτυχία η μία την άλλη. Αυτή είναι η προσέγγιση που χρησιμοποιούν όσοι κάνουν κριτικές προβλέψεις όταν δεν είναι διαθέσιμες επιπλέον εσωτερικές πληροφορίες και όταν δεν διακυβεύονται προσωπικά ή πολιτικά συμφέροντα. Στην πράξη μία τέτοια προσέγγιση, αν και μηχανική, φαίνεται να δουλεύει καλύτερα από κάποιες εναλλακτικές. Αυτό που είναι ενδιαφέρον είναι ότι οι προβλέψεις με στατιστικές μεθόδους επέκτασης της τάσης (trend extrapolation) δίνουν παρόμοια νούμερα με τις προβλέψεις τρίτων , από τους οποίους ζητείται να κάνουν κριτική πρόβλεψη των πωλήσεων της Elco.

Αυτό που είναι σημαντικό είναι το πόσο μεταβάλλονται οι προβλέψεις αν πούμε στους ανθρώπους ότι τα δεδομένα του σχήματος 1 αφορούν ένα νέο προϊόν (ή ένα ώριμο ή ένα παλιό προϊόν). Όπως φαίνεται στο σχήμα 2 οι απαντήσεις διαφέρουν αρκετά, γεγονός το οποίο αποδεικνύει ότι συχνά αγνοούμε σημαντικά δεδομένα και κάνουμε προβλέψεις χρησιμοποιώντας κάποια στερεότυπα (για παράδειγμα, οι πωλήσεις ενός νέου προϊόντος θα πρέπει να αυξάνονται, ενώ ενός παλιού να μειώνονται). Στην πραγματικότητα, οι προβλέψεις για νέο, ώριμο ή παλιό προϊόν διαφέρουν τόσο πολύ (ειδικά για το 2003) καθιστώντας επιτακτική την ανάγκη να αποβάλουμε τις μεροληψίες από τη διαδικασία μιας πρόβλεψης. Θα ήταν λάθος να θεωρήσουμε αυτόματα ότι οι πωλήσεις ενός προϊόντος θα αυξάνονται και μάλιστα περισσότερο από μία ευθεία αύξηση, μόνο και μόνο επειδή το προϊόν είναι καινούργιο – ειδικά στην αγορά των προσωπικών υπολογιστών όπου ο κύκλος ζωής ενός προϊόντος σπάνια ξεπερνάει τα δύο χρόνια καθιστώντας ένα προϊόν επταετίας ιδιαίτερα παλιό. Όμως αυτού του είδους τα λάθη γίνονται συχνά από ανθρώπους όταν τους ζητείται να προβλέψουν τις πωλήσεις της Elco.




**Σχήμα 2:** Πωλήσεις εκτυπωτών της Elco και προβλέψεις

Σε ένα πείραμα, τα δεδομένα του σχήματος 1 δόθηκαν σε μία ομάδα από διευθυντές προϊόντων (product managers) στους οποίους είπαν ότι πρόκειται για τις προηγούμενες πωλήσεις ενός προϊόντος της εταιρείας. Στην πρόβλεψη τους για το μέλλον παρουσίασαν μία σημαντική αύξηση παρόμοια με αυτή του «νέου προϊόντος» στο σχήμα 2. Ωστόσο, μία άλλη ομάδα διευθυντών προϊόντων (product managers) στην οποία δόθηκαν τα ίδια δεδομένα αλλά τους είπαν ότι πρόκειται για τις πωλήσεις προϊόντος των μεγαλύτερων ανταγωνιστών της εταιρείας τους, έκανε πρόβλεψη για σημαντική μείωση των πωλήσεων

παρόμοια με αυτή του «παλαιού προϊόντος» του σχήματος 2. Είναι προφανές ότι αυτές οι μεγάλες διαφορές στην πρόβλεψη είναι αποτέλεσμα μεροληψιών και απορρέουν είτε από αισιοδοξία για την εταιρία ή από επιθυμία για αποτυχία του ανταγωνισμού και τελικά δεν σχετίζονται με μία αντικειμενική εκτίμηση για τις μελλοντικές συνθήκες.

Τελικά, όλοι όσοι μελέτησαν το σχήμα 1 και έκαναν πρόβλεψη, υποτίμησαν την αβεβαιότητα που περιβάλει το θέμα (ο υπολογισμός του διαστήματος ανάμεσα στην υψηλή και τη χαμηλή πρόβλεψη). Αυτό ήταν αλήθεια ειδικά για το 2003, όπου οι πωλήσεις θα μπορούσαν να είναι πολύ υψηλότερες από την ανώτατη τιμή που δόθηκε και πολύ χαμηλότερες από την κατώτερη τιμή. Πολλά μπορούν να συμβούν μέσα σε πέντε χρόνια, όμως όσοι έκαναν προβλέψεις υποτίμησαν την αβεβαιότητα του μέλλοντος, ανεξάρτητα από τη θέση τους (φοιτητές MBA, διευθυντές) ή το υπόβαθρο τους (MBA, μηχανικοί, φοιτητές καλών τεχνών, κλπ). Υποθέτοντας μία γραμμική τάση, μία ρεαλιστική πρόβλεψη, βασισμένη στην στατιστική θεωρία, για το διάστημα μεταξύ χαμηλών και υψηλών πωλήσεων για το 1999 θα είναι από 50.000 μέχρι 79.000. Για το 2003 τα ίδια νούμερα είναι από 78.000 έως 119.000 μονάδες. Η πλειοψηφία αυτών που έκαναν πρόβλεψη έδωσε διαστήματα μικρότερα του μισού από αυτά που έδωσε η στατιστική θεωρία, η οποία πρέπει να πούμε υπολογίζει τα διαστήματα αυτά θεωρώντας ότι τα πρότυπα του παρόντος δεν θα αλλάξουν.

## 1.6 Αντιμετώπιση των κριτικών μεροληψιών

Για καλύψουμε ολόκληρο το θέμα των κριτικών μεροληψιών θα χρειαζόντουσαν πολλοί τόμοι και, επομένως, δεν είναι δυνατόν να το αναλύσουμε όλο σε αυτό το κεφάλαιο. Έτσι θα εστιάσουμε την προσοχή μας στους πιο σημαντικούς παράγοντες κριτικών μεροληψιών, οι οποίοι επηρεάζουν άμεσα τις προβλέψεις.

Ένα σημαντικό ελάττωμα των ανθρώπων είναι η ασυνέπεια, η τάση να αλλάζουμε τη γνώμη μας ή τις αποφάσεις μας όταν δεν υπάρχει λόγος να το κάνουμε. Θεωρήστε έναν διευθυντή παραγωγής (production manager) ο οποίος πρέπει να προβλέψει πόσα κομμάτια θα πρέπει να παρασκευαστούν από 10 προϊόντα, μέσα στον επόμενο μήνα. Ο Bowman (1963) το 1950 παρατήρησε ότι οι προβλέψεις των διευθυντών παραγωγής (production managers) και οι αποφάσεις τους σχετικά με την παραγωγή προϊόντων, παρουσίαζαν διακυμάνσεις από μήνα σε μήνα χωρίς να υπάρχει ιδιαίτερος λόγος. Όταν παρουσιάστηκε συνέπεια στη λήψη αποφάσεων αμέσως βελτιώθηκε η εγκυρότητα των προβλέψεων και η κερδοφορία. Τα

ευρήματα του Bowman έχουν αναπαραχθεί σε ένα μεγάλο αριθμό από μελέτες (Hogarth και Μακριδάκης, 1981) και το αποτέλεσμα είναι πάντα το ίδιο: οι επαναλαμβανόμενες προβλέψεις ρουτίνας (και αποφάσεις γενικά) μπορούν να βελτιωθούν σημαντικά αν αφαιρεθεί η ασυνέπεια. Οι άνθρωποι, όμως, συχνά δεν θέλουν ή δεν μπορούν να εφαρμόσουν τα ίδια κριτήρια και τις ίδιες διαδικασίες όταν λαμβάνουν παρόμοιες αποφάσεις. Ορισμένες φορές ξεχνούν, άλλες φορές απλώς επηρεάζονται από την διάθεση τους (σκεφτείτε κάποιον που πραγματοποιεί μία πρόβλεψη ένα πρωί ύστερα από μία νύχτα χωρίς ύπνο εξαιτίας ενός βραδινού καυγά με τον / την σύντροφο του). Άλλες φορές οι άνθρωποι απλώς βαριούνται και θέλουν να δοκιμάσουν κάτι καινούργιο. Τέλος, μπορεί να πιστεύουν ότι οι συνθήκες έχουν αλλάξει, ενώ στην πραγματικότητα αυτό δεν έχει συμβεί.

Οι διευθυντές παραγωγής (production managers) δεν είναι οι μόνοι που παρουσιάζουν ασυνέπεια στις αποφάσεις τους. Ο Meehl (1954), σε ένα μικρό αλλά σημαντικό βιβλίο του, κατέληξε στο ότι οι κανόνες αποφάσεων που χρησιμοποιούν έναν μικρό αριθμό μεταβλητών παρέχουν καλύτερες προβλέψεις από τους ανθρώπους, κυρίως γιατί τα μαθηματικά μοντέλα μπορούν με συνέπεια να εφαρμόζουν τα ίδια κριτήρια για τις αποφάσεις, ενώ οι άνθρωποι είναι ασυνεπείς στην επιλογή των παραγόντων επάνω στους οποίους βασίζουν τις αποφάσεις τους. Τα συμπεράσματα του Meehl έχουν επιβεβαιωθεί από εκατοντάδες άλλες μελέτες. Έχει αποδειχθεί ότι οι κανόνες αποφάσεων, με την μορφή απλών στατιστικών μοντέλων, υπερτερούν της κρίσης των ειδικών, όταν απαιτούνται επαναλαμβανόμενες αποφάσεις ρουτίνας. Σε αυτό το είδος αποφάσεων συμπεριλαμβάνονται ιατρικές διαγνώσεις, προβλέψεις ψυχολόγων για τα χαρακτηριστικά της προσωπικότητας ανθρώπων, επιλογή φοιτητών για την εισαγωγή τους σε κολέγια ή πανεπιστήμια, προβλέψεις μελλοντικών εσόδων εταιριών κ.ο.κ. Σε ελάχιστες περιπτώσεις φαίνεται η κρίση των ειδικών να δίνει καλύτερα αποτελέσματα από τους κανόνες αποφάσεων. Προφανώς αυτές οι μελέτες αναφέρονται σε επαναλαμβανόμενες αποφάσεις ρουτίνας, αλλά ακόμα και τότε τα συμπεράσματα προκαλούν έκπληξη, όπως στην περίπτωση των ιατρικών προβλέψεων. Ο Garland (1960), για παράδειγμα, παρέδωσε μία μελέτη, στην οποία έμπειροι ακτινολόγοι απέτυχαν, σε ποσοστό 30%, να διαγνώσουν μία ασθένεια του πνεύμονα η οποία ήταν εμφανής στην ακτινογραφία. Παρόμοιες μελέτες έδειξαν ότι ακτινολόγοι άλλαξαν διάγνωση, σε ποσοστό 20%, όταν τους δόθηκε η ίδια ακτινογραφία σε δύο διαφορετικές περιπτώσεις.

Για να αποφύγουμε την ασυνέπεια θα πρέπει να τυποποιήσουμε την διαδικασία λήψης αποφάσεων (στις μέρες μας αυτό ονομάζεται κατασκευή έμπειρων συστημάτων). Για να γίνει αυτό, πρώτα απαιτείται να καθοριστούν οι παράγοντες που θεωρούνται σημαντικοί στην λήψη μίας συγκεκριμένης επαναλαμβανόμενης απόφασης. Στην συνέχεια θα πρέπει να αξιολογηθούν αυτοί οι παράγοντες (κάποιος μπορεί να είναι πολύ πιο σημαντικός από κάποιον άλλο) και τέλος θα προσδιοριστούν οι στόχοι που επιθυμούμε να

βελτιστοποιήσουμε. Η χρησιμότητα των κανόνων αποφάσεων απορρέει από το γεγονός ότι συμμετέχουν αρκετοί άνθρωποι στον καθορισμό τους, έτσι ώστε είναι δυνατόν να επιλεγούν οι καλύτεροι παράγοντες, η καλύτερη αξιολόγηση τους και οι περισσότερο βιώσιμοι στόχοι. Εφόσον ένας κανόνας θα χρησιμοποιηθεί ξανά και ξανά, είναι λογικό να αφιερωθεί σημαντική ενέργεια για να βρεθεί ο καλύτερος δυνατός. Ο κανόνας μπορεί, στη συνέχεια, να εφαρμοστεί σε καθημερινή βάση εξοικονομώντας, έτσι, σημαντική ανθρώπινη ενέργεια και συνεισφέροντας στην βελτίωση της εγκυρότητας των προβλέψεων και της λήψης αποφάσεων γενικότερα. Σκεφτείτε, για παράδειγμα, αν θα πρέπει να εγκριθεί μία αγορά ενός κατόχου κάρτας Visa. Αν αυτό γινόταν συνέχεια και για κάθε αγορά, θα χρειαζόταν τεράστιο ανθρώπινο δυναμικό και η όλη επιχείρηση θα ήταν πολυδάπανη. Σκεφτείτε τώρα να βρίσκατε όλους τους σημαντικούς παράγοντες, με βάση τους οποίους θα αποφάσιζαν οι υπεύθυνοι να εγκρίνουν ή όχι μία τέτοια αγορά. Εφόσον μπορείτε να συμβουλευθείτε πολλούς ειδικούς για το θέμα και μπορεί να αφιερωθεί μεγάλη ενέργεια σε αυτή τη διαδικασία, είναι δυνατόν να βρεθούν οι πιο σημαντικοί παράγοντες και να συμπεριληφθούν σε ένα στατιστικό μοντέλο, το οποίο θα είναι σε θέση να καθορίσει αν αυτοί οι παράγοντες είναι πραγματικά σημαντικοί και τι είδους βαρύτητα πρέπει να δοθεί στον καθένα. Με αυτόν τον τρόπο θα καθιερωθεί ένας κανόνας απόφασης, ο οποίος θα επιτρέπει σε έναν υπάλληλο να εισάγει τις ζητούμενες πληροφορίες και το μοντέλο θα παρέχει μία απόφαση. Εφόσον ο στόχος είναι να ελαχιστοποιηθούν οι απάτες με πιστωτικές κάρτες, ένα έμπειρο σύστημα μπορεί να κατασκευαστεί και μόνο στις εξαιρετικές περιπτώσεις όπου το σύστημα δεν μπορεί να δώσει απάντηση  θα ζητείται η συμβουλή των ειδικών. Με την εφαρμογή ενός τέτοιου συστήματος απαιτείται μικρότερο ανθρώπινο δυναμικό και οι αποφάσεις παρουσιάζουν συνέπεια και αντικειμενικότητα. Εξίσου σημαντικό είναι ότι οι αποφάσεις μπορούν να αποτιμηθούν στην συνέχεια ώστε να είναι δυνατή η βελτίωση του μοντέλου, αν αυτό κριθεί απαραίτητο.

Με δεδομένη την σημερινή τεχνολογία στον χώρο των ηλεκτρονικών υπολογιστών και των τηλεπικοινωνιών, συστήματα σαν αυτό που περιγράψαμε παραπάνω είναι δυνατόν να αναπτυχθούν με χαμηλό κόστος και να προσφέρουν αυξημένη κερδοφορία. Η Visa έχει, πράγματι, εφαρμόσει τέτοια μοντέλα λήψης αποφάσεων με αποτέλεσμα την αυξημένη αποδοτικότητα και το κέρδος. Παρόμοιοι κανόνες αποφάσεων μπορούν να εφαρμοστούν όταν πραγματοποιούμε κριτικές προβλέψεις.

Προφανώς, οι κανόνες απόφασης δεν μπορούν να χρησιμοποιηθούν αόριστα. Το περιβάλλον αλλάζει, όπως επίσης και ο ανταγωνισμός, ενώ είναι πιθανό να χρειαστεί να καθοριστούν νέοι στόχοι. Έτσι, η αποτελεσματικότητα των κανόνων πρέπει να παρακολουθείται συστηματικά για να είμαστε σίγουροι ότι εξακολουθούν να είναι κατάλληλοι. Αυτό σημαίνει ότι θα πρέπει να εισαχθεί μία διαδικασία μάθησης στα έμπειρα συστήματα, διαφορετικά υπάρχει ο κίνδυνος να εφαρμόζουμε απαρχαιωμένους κανόνες. Η

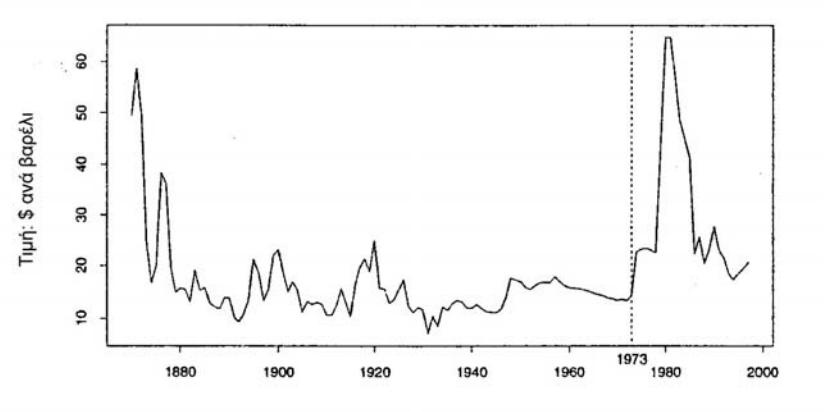
υπερβολική συνέπεια μπορεί να είναι το ίδιο επικίνδυνη με την ασυνέπεια καθώς αποκλείει την μάθηση και οδηγεί στον συντηρητισμό.

Αυτό, ακριβώς, είναι και το πρόβλημα με τις μεροληψίες: μπορεί προσπαθώντας να αποφύγουμε μία να καταλήγουμε σε άλλη. Η μεροληψία σε αυτήν την περίπτωση υπάρχει γιατί στο μυαλό μας προσπαθούμε να πετύχουμε την συνέπεια, όμως πρέπει να αφήσουμε περιθώριο και για μάθηση. Η πρόκληση που αντιμετωπίζουμε όλοι, επομένως, είναι να είμαστε συνεπείς και συγχρόνως να εισάγουμε μηχανισμούς που εξασφαλίζουν μάθηση και τελικά οδηγούν σε αλλαγές των κανόνων, όπου αυτό είναι απαραίτητο, ώστε να προσαρμόζονται σε νέες συνθήκες. Αυτή είναι μία μεγάλη πρόκληση για αυτούς που κάνουν προβλέψεις, ειδικά για τις μακροπρόθεσμες προγνώσεις, όπου οι αλλαγές είναι άφθονες ενώ η μάθηση σπάνια.

Μπορούμε, μήπως, να αποφύγουμε τις μεροληψίες αν τις προβλέψεις τις πραγματοποιούν σύνολα ανθρώπων; Δυστυχώς όχι. Στην πραγματικότητα, τα στοιχεία δείχνουν ότι με αυτόν τον τρόπο ενισχύονται οι μεροληψίες καθώς εμφανίζεται το φαινόμενο της συμμόρφωσης με τις απόψεις της ομάδας. Κατά το φαινόμενο αυτό, τα μέλη μίας ομάδας υποστηρίζουν τον αρχηγό της ή τα υπόλοιπα μέλη, αποφεύγοντας με αυτόν τον τρόπο τις συγκρούσεις και τις διαφωνίες κατά τη διάρκεια των συμβουλίων. Επιπλέον, οι αποφάσεις συνόλου είναι περισσότερο επικίνδυνες καθώς η τελική ευθύνη μίας απόφασης δεν μπορεί να αποδοθεί σε κάποιο συγκεκριμένο άτομο. Στον πίνακα 2 περιγράφονται οι μεροληψίες που εμείς θεωρούμε (από την εμπειρία μας δουλεύοντας με εταιρείες, από την ερευνητική μας δουλειά και από σχετικά ευρήματα στην βιβλιογραφία προβλέψεων) ιδιαίτερα σημαντικές για τις προβλέψεις και γενικότερα για την λήψη αποφάσεων σχετικά με το μέλλον. Στον πίνακα, επίσης, παρέχονται προτάσεις για το πώς να αποφύγουμε ή τουλάχιστον να μετριάσουμε τις συνέπειες αυτών των μεροληψιών.

Ένα χαρακτηριστικό παράδειγμα μίας τέτοιας μεροληψίας – της τάσης να θυμόμαστε πιο ζωντανά τα πρόσφατα γεγονότα με αποτέλεσμα να μας επηρεάζουν περισσότερο από ότι παλαιότερα γεγονότα – αφορά τις τιμές πετρελαίου μεταξύ του 1965 και του 1988. Κατά τη διάρκεια αυτής της περιόδου, βασικά οικονομικά δεδομένα αγνοήθηκαν και έγιναν σημαντικά λάθη, καθώς κυβερνήσεις και οργανισμοί αντέδρασαν υπερβολικά στα τελευταία επίπεδα των τιμών και έλαβαν αποφάσεις θεωρώντας ότι αυτές οι τιμές (ή οι τάσεις που επικρατούσαν) θα κρατήσουν για πάντα. Αυτό αποδείχθηκε λάθος και πριν το 1973, όταν οι τιμές μειώνονταν και μεταξύ 1974 και 1981 όταν αυξάνονταν υπερβολικά. Αν είχαν χρησιμοποιηθεί όλες οι πληροφορίες σχετικά με την τιμή του πετρελαίου (σχήμα 3) θα ήταν φανερό ότι η πραγματική τιμή του πετρελαίου θα παρέμενε σταθερή μακροπρόθεσμα. Αυτό ήταν αλήθεια πριν το 1974 και σήμερα. Έτσι, θα έπρεπε να έχει υποτεθεί ότι οι αποκλίσεις από την μακροπρόθεσμη τάση θα ήταν προσωρινές και ότι η αγορά θα επέστρεφε στην μακροπρόθεσμη ισορροπία που περιλαμβάνεται στην βασική τάση της τιμής του

πετρελαίου. Στην πραγματικότητα αυτό συνέβη, αν και οι τιμές του πετρελαίου εκτοξεύθηκαν από τα 14.2 δολάρια το 1973 στα 64.7 δολάρια επτά χρόνια αργότερα. Όπως δείχνει το σχήμα 3, η τιμή του πετρελαίου το 1997 είναι πολύ κοντά στην μακροπρόθεσμη βασική τάση αυτών των τιμών.




**Σχήμα 3:** Τιμή του πετρελαίου σε δολάρια του 1997. 1870-1997

| Τύπος μεροληψίας | Περιγραφή | Τρόποι αποφυγής ή μείωσης των αρνητικών συνεπειών |
|---|---|---|
| Ασυνέπεια | Η αδυναμία να εφαρμόσουμε τα ίδια κριτήρια αποφάσεων σε παρόμοιες καταστάσεις | -Τυποποίηση της διαδικασίας λήψης αποφάσεων<br>-Δημιουργία κανόνων αποφάσεων |
| Συντηρητισμός | Αδυναμία να αλλάξουμε γνώμη ( ή καθυστέρηση στην αλλαγή) όταν εμφανίζονται νέα στοιχεία | -Παρακολούθηση για αλλαγές στο περιβάλλον και δημιουργία μηχανισμών ώστε να υπάρχει αντίδραση όταν εμφανίζονται αυτές |
| Επιμονή σε πρόσφατα γεγονότα | Τα πιο πρόσφατα γεγονότα μας επηρεάζουν πιο πολύ ενώ τα παλαιότερα αγνοούνται ή υποβαθμίζονται | -Αναγνωρίστε ότι υπάρχουν κύκλοι και όλες οι αυξήσεις ή μειώσεις δεν είναι μόνιμες<br>-Καθορισμός των κυρίων παραγόντων που επηρεάζουν το θέμα που μας ενδιαφέρει |
| «Διαθεσιμότητα» (Availability) | Στηριζόμαστε σε συγκεκριμένα γεγονότα τα οποία θυμόμαστε εύκολα και αποκλείουμε άλλες σχετικές πληροφορίες. | -Παρουσιάστε όλες τις σχετικές πληροφορίες<br>-Παρουσιάστε τις πληροφορίες με τέτοιο τρόπο ώστε να φωτίζονται όλες οι πλευρές της κατάστασης που εξετάζετε |
| «Αγκυροβόληση» | Επηρεαζόμαστε υπερβολικά από την | -Αρχίστε με αντικειμενικές πληροφορίες |

| | | |
|---|---|---|
| (Anchoring) | αρχική πληροφορία στην οποία δίδεται περισσότερο βάρος στην διαδικασία πρόβλεψης | (π.χ. προβλέψεις)<br>-Ζητήστε από ανθρώπους να συζητήσουν τους πιθανούς τύπους αλλαγών. Ρωτήστε τους λόγους για τις πιθανές αλλαγές |
| Λανθασμένες συσχετίσεις | Πιστεύουμε ότι υπάρχουν κάποια πρότυπα και / ή δύο μεταβλητές σχετίζονται αιτιολογικά, ενώ αυτό δεν συμβαίνει | -Επαληθεύστε την στατιστική σημασία των προτύπων<br>-Μοντελοποιήστε τις συσχετίσεις, αν είναι δυνατόν, σχετικά με τις αλλαγές |
| Αναζήτηση συγκεκριμένων στοιχείων | Αναζητούμε δεδομένα τα οποία οδηγούν σε συγκεκριμένα συμπεράσματα και υποστηρίζουν κάποια άποψη, παραβλέποντας άλλα στοιχεία τα οποία αντιτίθενται σε αυτή | -Συμπεριλάβετε στοιχεία που διαψεύδουν αυτή την άποψη<br>-Εισάγετε τον ρόλο του «δικηγόρου του διαβόλου» |
| Εντύπωση παλινδρόμησης | Συνεχείς αυξήσεις ή μειώσεις, μπορεί να οφείλονται στην τύχη παρά σε κάποια πραγματική τάση | -Πρέπει να γίνει κατανοητό ότι αν τα σφάλματα είναι τυχαία, η φαινομενική τάση πιθανότατα δεν θα συνεχιστεί |
| Απόδοση της επιτυχίας και της αποτυχίας | Πιστεύουμε ότι η επιτυχία οφείλεται στις ικανότητες μας ενώ η αποτυχία στην κακή τύχη ή στα σφάλματα κάποιου άλλου. Αυτό εμποδίζει την μάθηση καθώς δεν επιτρέπει την αναγνώριση των σφαλμάτων μας | -Μην τιμωρείτε τα λάθη. Αντίθετα, ενθαρρύνετε τους ανθρώπους να τα δέχονται και να τα μοιράζονται με τους υπόλοιπους, ώστε και οι άλλοι να μαθαίνουν από αυτά και να αποφεύγουν παρόμοια σφάλματα στο μέλλον ( αυτή την τακτική ακολουθούν οι εταιρείες στην Ιαπωνία) |
| Αισιοδοξία, ευσεβής πόθος | Οι προτιμήσεις των ανθρώπων για το μέλλον συχνά επηρεάζει τις προβλέψεις τους για αυτό | -Αναθέστε τις προβλέψεις σε τρίτα πρόσωπα, αμερόληπτα<br>-Αναθέστε σε περισσότερους από έναν ανθρώπους, ανεξάρτητες προβλέψεις |
| Υποτίμηση της αβεβαιότητας | Υπερβολική αισιοδοξία, λανθασμένες συσχετίσεις και η ανάγκη να μειώσουμε την ανησυχία οδηγούν σε υποτίμηση μελλοντικών αβεβαιοτήτων | -Εκτιμήστε την αβεβαιότητα με αντικειμενικό τρόπο. Θεωρήστε πολλά πιθανά μελλοντικά γεγονότα, ζητώντας από διαφορετικούς ανθρώπους να σκεφτούν απρόβλεπτες καταστάσεις / γεγονότα |
| Επιλεκτική αντίληψη | Αντιμετωπίζουμε ένα πρόβλημα με βάση το υπόβαθρο και τις εμπειρίες ενός ανθρώπου | -Αναθέστε σε άτομα με διαφορετικές εμπειρίες και διαφορετικό υπόβαθρο να βρουν ανεξάρτητες λύσεις |

**Πίνακας 2:** Συνήθεις μεροληψίες και τρόποι αντιμετώπισής τους

## 1.7 Συμβατική σοφία (conventional wisdom)

Ένας άλλος τύπος κριτικής μεροληψίας, που μπορεί να απειλήσει την αποτελεσματικότητα της λήψης μίας απόφασης είναι η αβάσιμη πίστη ή η συμβατική σοφία. Για παράδειγμα, πιστεύουμε ότι όσο περισσότερες πληροφορίες έχουμε τόσο πιο έγκυρες θα είναι οι αποφάσεις μας. Εν τούτοις, τα εμπειρικά δεδομένα δεν υποστηρίζουν αυτή την άποψη. Αντίθετα, οι περισσότερες πληροφορίες φαίνεται απλώς να αυξάνουν την πεποίθηση μας ότι έχουμε δίκιο χωρίς, απαραίτητα, να βελτιώνουν την εγκυρότητα των αποφάσεων μας. Σε αυτό το συμπέρασμα έφτασε ο Oskamp (1965) και διάφοροι άλλοι ερευνητές οι οποίοι προειδοποιούν ότι δεν πρέπει να σπαταλάμε ενέργεια και πόρους για να συλλέγουμε πληθώρα πληροφοριών. Στην πραγματικότητα , η επιπλέον πληροφορία είναι συνήθως περιττή και προσφέρει ελάχιστη επιπρόσθετη αξία ενώ μπορεί να είναι και επιζήμια καθώς αυξάνει την βεβαιότητα μας για το μέλλον.

Ένα άλλο παράδειγμα συμβατικής σοφίας που δεχόμαστε είναι ότι μπορούμε να διακρίνουμε τις «χρήσιμες» από τις «άσχετες» πληροφορίες. Οι μελέτες δείχνουν ότι συνήθως αυτό δεν συμβαίνει. Σε διάφορα πειράματα, άτομα που τροφοδοτούνται με «καλές» και «κακές» πληροφορίες δεν είναι σε θέση να τις ξεχωρίσουν. Επιπλέον, συχνά χρησιμοποιούνται οι άσχετες πληροφορίες, μειώνοντας έτσι την αποτελεσματικότητα της διαδικασίας λήψης αποφάσεων, ειδικά αν οι σχετικές πληροφορίες είναι ποσοτικές ενώ οι άσχετες είναι ποιοτικές (οι περισσότεροι άνθρωποι φαίνεται να δίνουν μεγαλύτερη βαρύτητα σε πληροφορίες εκφρασμένες με λέξεις παρά με νούμερα).

Στον πίνακα 3, συνοψίζονται οι σχετικές συμβατικές σοφίες, συμπεριλαμβανομένων των δύο παραδειγμάτων που παρουσιάσαμε. Επίσης, καταγράφονται τα στοιχεία που είναι διαθέσιμα από την εμπειρία μας και ύστερα από μελέτες. Όπως και με τις μεροληψίες που παρουσιάσαμε παραπάνω, οι συμβατικές σοφίες μπορούν να επηρεάσουν σε μεγάλο βαθμό τις προβλέψεις μας αλλά και, γενικότερα, τις αποφάσεις μας. Επομένως, είναι ιδιαίτερα σημαντικό να προσπαθήσουμε να αποφύγουμε τις αρνητικές συνέπειες τους ακολουθώντας διαδικασίες με στόχο την ελαχιστοποίηση της επιρροής τους με την βοήθεια και των εμπειρικών δεδομένων που καταγράφονται στον πίνακα 3.

## 1.8 Συνδυάζοντας στατιστικές και κριτικές προβλέψεις

Η μεγάλη πρόκληση στην πραγματοποίηση μίας ορθής πρόβλεψης είναι η χρησιμοποίηση των καλύτερων στοιχείων της στατιστικής πρόβλεψης και η παράλληλη εκμετάλλευση της γνώσης, της κρίσης και της εμπειρίας. Στο υπόλοιπο αυτού του κεφαλαίου θα περιγράψουμε μία προσέγγιση, που έχει χρησιμοποιηθεί αρκετές

| Συμβατική Σοφία | Εμπειρικά Ευρήματα |
|---|---|
| 1. Όσο περισσότερες πληροφορίες έχουμε τόσο πιο έγκυρες είναι οι αποφάσεις. | Το μέγεθος της πληροφορίας δεν βελτιώνει την ορθότητα των αποφάσεων. Απλώς αυξάνει την πεποίθηση ότι η απόφαση μας θα είναι σωστή. |
| 2. Μπορούμε να ξεχωρίσουμε τις χρήσιμες από τις άσχετες πληροφορίες. | Οι άσχετες πληροφορίες μπορεί να είναι η αιτία της μείωσης της ορθότητας των αποφάσεων μας. |
| 3. Όσο μεγαλύτερη πεποίθηση έχουμε ότι οι αποφάσεις μας θα είναι ορθές, τόσο πιο ορθές θα είναι τελικά. | Δεν υπάρχει καμία σχέση ανάμεσα στην αυτοπεποίθηση κάποιου και στην ορθότητα τον αποφάσεων του / της. |
| 4. Μπορούμε να αποφασίσουμε λογικά πότε ήρθε η ώρα να εγκαταλείψουμε. | Νιώθουμε ότι έχουμε επενδύσει πάρα πολλά για να εγκαταλείψουμε. |
| 5. Χρηματικές αμοιβές και τιμωρίες οδηγούν σε καλύτερη απόδοση. | Η ανθρώπινη συμπεριφορά είναι ιδιαίτερα πολύπλοκη για να υποκινείται μόνο από χρηματικούς παράγοντες. |
| 6. Μπορούμε να εκτιμήσουμε σχετικά καλά τις πιθανότητες επιτυχίας και αποτυχίας μας. | Είμαστε πολύ αισιόδοξοι και τείνουμε να υποτιμούμε ή να αγνοούμε προβλήματα και δυσκολίες. |
| 7. Η εμπειρία και η εξειδίκευση βελτιώνουν την ορθότητα των αποφάσεων. | Σε πολλές επαναλαμβανόμενες αποφάσεις ρουτίνας η εμπειρία και η εξειδίκευση δεν δίνουν μεγαλύτερη αξία σε |

| | |
|---|---|
| | αποφάσεις σχετικά με το μέλλον. |
| 8. Ξέρουμε στην πραγματικότητα τι θέλουμε και οι προτιμήσεις μας είναι σταθερές. | Μικρές διαφορές σε μία κατάσταση μπορούν να αλλάξουν τις προτιμήσεις μας. |

**Πίνακας 3:** Συμβατική σοφία εναντίον εμπειρικών ευρημάτων

φορές, η οποία εκμεταλλεύεται τα πλεονεκτήματα και των κριτικών και των στατιστικών προβλέψεων ενώ αποφεύγει τα μειονεκτήματα τους. Το συγκεκριμένο παράδειγμα αναφέρεται σε πρόβλεψη προϋπολογισμού όμως η προσέγγιση αυτή μπορεί να υιοθετηθεί και σε άλλες περιπτώσεις που απαιτούν πρόβλεψη.

### 1.8.1 Κατάληξη σε τελικές προβλέψεις κατά τη διάρκεια σύσκεψης για τον προϋπολογισμό

Στις ετήσιες συσκέψεις για τον προϋπολογισμό (συνήθως γίνονται τον Οκτώβριο ή τον Νοέμβριο) ο κύριος στόχος είναι να αποφασιστεί πόσο θα αναπτυχθούν οι πωλήσεις τον επόμενο χρόνο. Με βάση αυτή την ανάπτυξη, λαμβάνονται πολλές άλλες αποφάσεις σχετικά με την κατανομή των πόρων και την εφαρμογή μακροπρόθεσμων σχεδίων. Επομένως, καταβάλλεται σημαντική προσπάθεια ώστε η πρόβλεψη της ανάπτυξης των συνολικών πωλήσεων να είναι όσο το δυνατόν ορθότερη.

Συνήθως, σε τέτοιες συσκέψεις προϋπολογισμού υπάρχουν αντικρουόμενα συμφέροντα ανάμεσα στους συμμετέχοντες καθώς οι υπεύθυνοι του marketing επιλέγουν υψηλότερη ανάπτυξη ενώ οι υπεύθυνοι παραγωγής χαμηλότερη. Η τελική απόφαση για την ανάπτυξη των πωλήσεων είναι συχνά αποτέλεσμα μίας διαδικασίας παζαρέματος ανάμεσα στους συμμετέχοντες στην σύσκεψη και δεν έχει μεγάλη σχέση με προβλέψεις (Walker και McClelland, 1991) αλλά περισσότερο με την σχετική δύναμη του κάθε διοικητικού στελέχους που μετέχει στην σύσκεψη, την δυνατότητα του / της να πείθει τους άλλους, την προσωπικότητα, θέματα πολιτικής, την γνώμη του Γενικού Διευθυντή (CEO) και αμοιβαίες ανταλλαγές και υποχωρήσεις για την αποφυγή συγκρούσεων. Επομένως, καθίσταται σημαντική η δημιουργία μίας σταθερής και αντικειμενικής βάσης και μίας συστηματικής διαδικασίας για την ορθότερη δυνατή πρόβλεψη της

ανάπτυξης των πωλήσεων. Για να είναι αυτό δυνατό, απαιτείται η ελαχιστοποίηση των κριτικών μεροληψιών και των υποκειμενικών θεωρήσεων.

### 1.8.2 Αντικειμενική «αγκυροβόληση» (Anchoring) της αρχικής πρόβλεψης

Η «αγκυροβόληση» είναι μία κριτική μεροληψία, η οποία εκδηλώνεται όταν κάποιος ξεκινήσει με μία αρχική πρόβλεψη της ανάπτυξης, που μπορεί να μην είναι ρεαλιστική, αλλά όμως χρησιμεύει ως άγκυρα για να κρατήσει την τελική πρόβλεψη σε κοντινές τιμές. Οι συνέπειες αυτής της μεροληψίας είναι ακόμη πιο σοβαρές όταν ο πρόεδρος ή ο Γενικός διευθυντής (CEO) ξεκινήσει καθορίζοντας μία πρόβλεψη της ανάπτυξης για τον επόμενο χρόνο. Η συζήτηση εστιάζεται (αγκυροβολεί) σε αυτή την πρόβλεψη και δεν εξετάζει εναλλακτικές περιπτώσεις πολύ διαφορετικές (μακρινές) από αυτή. Για την αποφυγή αυτής της μεροληψίας, ή ακόμη για την χρησιμοποίηση της προς όφελος της εταιρείας, μοιράζεται προκαταβολικά σε όλους τους συμμετέχοντες στην σύσκεψη ένας φάκελος που περιέχει ιστορικές πληροφορίες για την οικονομία, την βιομηχανία και την εταιρεία. Αυτές οι πληροφορίες επεκτείνονται παρέχοντας έτσι αντικειμενικές προβλέψεις για την οικονομία, την βιομηχανία και την εταιρεία.

Γίνεται, επίσης, σαφές ότι αυτές οι προβλέψεις θα είναι έγκυρες αν και μόνο αν το μέλλον αποτελέσει συνέχεια του παρελθόντος. Ωστόσο, δηλώνεται στους συμμετέχοντες ότι είναι πολύ πιθανόν να συμβούν αλλαγές και ότι αυτοί είναι στην κατάλληλη θέση για να εκτιμήσουν το μέγεθος και τις συνέπειες αυτών των αλλαγών.

Ζητείται, επομένως, από τους συμμετέχοντες να χρησιμοποιήσουν την γνώση τους για την αγορά και τον ανταγωνισμό, όπως επίσης και την εμπειρία τους, για να εκτιμήσουν το μέγεθος στο οποίο θα πρέπει να αλλαχθούν οι αντικειμενικές προβλέψεις της ανάπτυξης και να σημειώσουν όλους τους παράγοντες που παίζουν ρόλο. Δηλαδή, δεν ζητείται από τους συμμετέχοντες να πραγματοποιήσουν μία πρόβλεψη από το μηδέν, αλλά να τροποποιήσουν την στατιστική πρόβλεψη, η οποία χρησιμοποιείται, επομένως, ως άγκυρα για να εμποδίσει προβλέψεις που δεν βασίζονται στις αντικειμενικές πληροφορίες για το παρελθόν της εταιρείας και στις επικρατούσες συνθήκες στην οικονομία και τη βιομηχανία. Παράλληλα, ζητείται από τους συμμετέχοντες να εξηγήσουν τους παράγοντες που είναι υπεύθυνοι για τις όποιες αλλαγές και τροποποιήσεις πρότειναν στις αντικειμενικές στατιστικές προβλέψεις. Τέλος, καταγράφουν τις προβλέψεις τους, ανώνυμα ώστε να μην επηρεάζονται από την θέση στην ιεραρχία του καθενός.

Με αυτόν τον τρόπο, όλες οι προβλέψεις έχουν το ίδιο βάρος και κινούνται όλες γύρω από τις αντικειμενικές στατιστικές προβλέψεις. Για την διευκόλυνση και αποσαφήνιση της διαδικασίας, δίνεται σε κάθε ένα από τους συμμετέχοντες μία φόρμα παρόμοια με τον πίνακα 4, για να την συμπληρώσει ανώνυμα αφού διαβάσει τις πληροφορίες που περιέχονται στον φάκελο που τους δόθηκε αρχικά. Αφού συλλεχθούν όλες οι φόρμες, τα αποτελέσματα τους συνοψίζονται και παρουσιάζονται στους συμμετέχοντες πριν ξεκινήσει η συζήτηση.

Τρεις παρατηρήσεις προκύπτουν από την εμπειρία μας από την διαδικασία με τις φόρμες. Πρώτον, οι συμμετέχοντες είναι πράγματι αγκυροβολημένοι στις στατιστικές, αντικειμενικές, προβλέψεις, τις οποίες αλλάζουν μόνο αν υπάρχουν πολύ σημαντικοί λόγοι που καθιστούν την αλλαγή απαραίτητη. Δεύτερον, ορισμένοι παράγοντες, σύμφωνα με κάποιους από τους συμμετέχοντες, αυξάνουν τις πωλήσεις, ενώ άλλοι συμμετέχοντες πιστεύουν ότι τις μειώνουν. Αυτοί οι παράγοντες μπορούν να παραβλεφθούν, εκτός και αν στην συνέχεια αποδειχθεί ότι η σημασία τους είναι μεγάλη. Τρίτον, οι περισσότεροι παράγοντες προτείνονται από έναν ή δύο από τους συμμετέχοντες και όχι από όλους ή, τουλάχιστον, από την πλειοψηφία. Επομένως, η συζήτηση μπορεί να επικεντρωθεί στον ένα ή στους λίγους παράγοντες

Επεκτείνοντας τις ιστορικές ποσοτικές πληροφορίες που είναι διαθέσιμες, η καλύτερη εκτίμηση για την ανάπτυξη στις πωλήσεις μας για το 1999 είναι 3.5%. Αυτή η στατιστική εκτίμηση θεωρεί ότι το 1999 θα είναι παρόμοιο με τα προηγούμενα χρόνια. Δηλαδή, δεν θα συμβούν σημαντικές αλλαγές ή αν συμβούν, η μία θα αναιρεί την άλλη. Αν πιστεύετε ότι θα συμβούν σημαντικές αλλαγές το 1999, καταγράψτε τις παρακάτω και εκτιμήστε την θετική ή αρνητική επιρροή τους χρησιμοποιώντας το ποσοστό ανάπτυξης 3.5% ως βάση.

| **Παράγοντες** | % Θετική επίδραση του παράγοντα στην αύξηση των πωλήσεων | % Αρνητική επίδραση του παράγοντα στην μείωση των πωλήσεων |
|---|---|---|
| Οικονομικοί | | |
| Βιομηχανικοί | | |
| Ανταγωνιστικοί | | |

| | | |
|---|---|---|
| Τεχνολογικοί<br><br>Άλλοι | | |
| **Συνολική επιρροή** | % Θετική = | % Αρνητική = |
| Η εκτίμηση σας για το ποσοστό ανάπτυξης | 3.5% + %Θετική - %Αρνητική<br>3.5% + - = | |

**Πίνακας 4:** Η φόρμα καταγραφής των παραγόντων και της επιρροής τους

που θεωρούνται από την πλειοψηφία ότι είναι σημαντικοί και να συζητηθεί το μέγεθος στο οποίο θα επηρεάσουν τις πωλήσεις του επόμενου χρόνου.

Αφού ολοκληρωθεί η σύσκεψη για τον προϋπολογισμό, οι συμμετέχοντες συμφωνούν για μία τελική πρόβλεψη. Αυτή η πρόβλεψη προκύπτει από την εκμετάλλευση του πλεονεκτήματος της αντικειμενικής στατιστικής μεθόδου αλλά και των πληροφοριών και της εμπειρίας των μελών της διοίκησης σχετικά με επερχόμενες αλλαγές αλλά και τις συνέπειες τους. Αυτή η προσέγγιση καθοδηγεί, επίσης, την συζήτηση στα πιο ουσιώδη σημεία και επομένως η σύσκεψη είναι εποικοδομητική και αποτελεσματική.

Τέλος, έχοντας συμφωνήσει στους παράγοντες οι οποίοι θα επηρεάσουν την ανάπτυξη των πωλήσεων το επόμενο έτος και έχοντας εκτιμήσει το μέγεθος των επερχόμενων αλλαγών, είναι δυνατόν να γίνει η αποτίμηση, στο τέλος του επόμενου χρόνου, της ορθότητας των προβλέψεων που έγιναν σε σύγκριση με τις στατιστικές μεθόδους. Με αυτόν τον τρόπο, όχι μόνο ελέγχεται η ορθότητα των κριτικών προβλέψεων σε σύγκριση με την αντίστοιχη των στατιστικών μεθόδων, αλλά μπορεί να εκτιμηθεί και η σχετική εγκυρότητα του καθενός από τους συμμετέχοντες και η ικανότητα τους να προβλέπουν σωστά τους παράγοντες που θα επηρεάσουν τις πωλήσεις. Έτσι, οι συμμετέχοντες θα μπορούν την επόμενη χρονιά να μελετήσουν την απόδοση τους και να βελτιωθούν ατομικά αλλά και σαν ομάδα. Επιπλέον, οι διάφοροι παράγοντες που επηρέασαν τις πωλήσεις μπορούν να εκτιμηθούν και να ληφθούν ποσοτικές πληροφορίες σχετικά με την επίδραση τους, οι οποίες θα είναι χρήσιμες στο μέλλον όταν συμβούν παρόμοιες αλλαγές.

## 1.9 Συμπέρασμα

Οι κριτικές προβλέψεις είναι πράγματι απαραίτητες καθώς αποτελούν την μοναδική εναλλακτική για την πρόβλεψη συστηματικών αλλαγών από καθιερωμένα πρότυπα και υπάρχουσες σχέσεις. Την ίδια στιγμή, θα πρέπει να είμαστε προσεκτικοί ώστε να αποφεύγουμε τις μεροληψίες και άλλους περιορισμούς που χαρακτηρίζουν την κρίση μας για να μειώνουμε τις αρνητικές τους συνέπειες στις προβλέψεις. Η πρόκληση για τις εταιρείες είναι να εκμεταλλεύονται και τις στατιστικές προβλέψεις αλλά και την μοναδική ικανότητα της ανθρώπινης κρίσης να αντιμετωπίζει συστηματικές αλλαγές σε πρότυπα / σχέσεις.

Κεφάλαιο

2

# Υποκειμενική – κριτική επέμβαση στις προβλέψεις και τα ιστορικά δεδομένα χρονοσειρών με χρήση νευρωνικών δικτύων

## 2.1 Βασικές έννοιες για τα ιστορικά δεδομένα και την κριτική επέμβαση

Τα μοντέλα χρονοσειρών χρησιμοποιούν τα ιστορικά δεδομένα για την παραγωγή μελλοντικών προβλέψεων αναγνωρίζοντας παρελθοντικά πρότυπα χωρίς να μπορούν να ενσωματώσουν την επίδραση εξωτερικών μελλοντικών γεγονότων ακόμα και αν αυτά έχουν ανακοινωθεί πριν συμβούν. Είναι σύνηθες λοιπόν οι χρήστες των στατιστικών προβλέψεων να τις προσαρμόζουν σύμφωνα με την πείρα τους όταν ανακοινώνεται ένα εξωτερικό γεγονός ή όταν διαισθάνονται κάποιες επερχόμενες αλλαγές στην αγορά που τους ενδιαφέρει. Αυτή η διαδικασία οδηγεί σε βελτίωση των προβλέψεων και ονομάζεται **"Υποκειμενική – Κριτική Επέμβαση στις Προβλέψεις Στατιστικών Μοντέλων"** είναι όμως ιδιαίτερα χρονοβόρα και επίσης πολλές φορές προκατειλημμένη ιδιαίτερα όταν οι προβλέψεις αναφέρονται σε προϊόντα της εταιρείας τους. Η αυτοματοποίηση αυτής της διαδικασίας μέσω Νευρωνικών Δικτύων οδηγεί σε πολύ καλά αποτελέσματα.

**Υποκειμενικός – Κριτικός Παράγων (Judgmental Factor)** ορίζεται ένας παράγοντας του υπό εξέταση μεγέθους (για το οποίο ζητούνται μελλοντικές προβλέψεις) του οποίου η επίδραση στα ιστορικά δεδομένα δεν μπορεί να αναγνωριστεί σαφώς και να οριστεί αντικειμενικά.

Κοινά χαρακτηριστικά των υποκειμενικών παραγόντων είναι τα ακόλουθα:

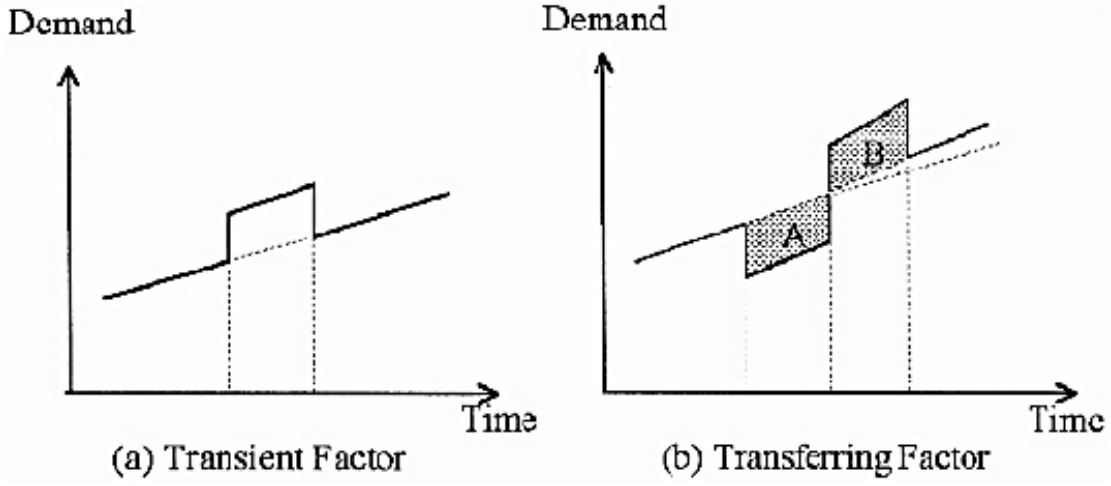
- Συνήθως συμβαίνουν απροειδοποίητα και οπωσδήποτε σε μη περιοδικά χρονικά διαστήματα. Το δείγμα είναι πολύ μικρό για στατιστική μοντελοποίηση.
- Η επίδραση τους είναι πολύ σημαντική για να αγνοηθεί.
- Η επίδραση τους είναι παροδική
- Είναι δύσκολο να προσδιοριστεί το μέγεθος της επίδρασης τους στο υπό μελέτη μέγεθος ακόμα και να προσδιοριστεί το ακριβές χρονικό διάστημα που αναμένεται να συμβούν.

Στο παράδειγμα που θα εξετάσουμε, που αφορά τις πωλήσεις των κρατικών Διυλιστηρίων της Κορέας, μπορούν να αναγνωριστούν τέσσερις τυπικού υποκειμενικοί παράγοντες που επηρεάζουν την ζήτηση Βενζίνης στην Κορεατική αγορά.

1. Κυβερνητικές διατάξεις εξοικονόμησης ενέργειας
2. Απεργίες βιομηχανιών που καταναλώνουν πολύ ενέργεια (πετρέλαιο)
3. Ειδικές αργίες – διακοπές
4. Ανακοινώσεις μελλοντικών αλλαγών στην τιμή των καυσίμων

Οι τέσσερις αυτοί παράγοντες αποτελούν την **Βάση Υποκειμενικών Παραγόντων** για το μέγεθος που εξετάζουμε.

Η επίδραση των τριών πρώτων παραγόντων είναι παροδική (transient) ενώ του τελευταίου μεταφερόμενη (transferring) πριν και μετά την αναγγελθείσα ημερομηνία. Η διαφορά των δύο επιδράσεων φαίνεται στο σχήμα που ακολουθεί στην αρχή της επόμενης σελίδας




**Σχήμα 1.** Παροδική (transient) και μεταφερόμενη (transferring) επίδραση.

Η μοντελοποίηση και τα χαρακτηριστικά των τεσσάρων υποκειμενικών παραγόντων ορίζονται ακολούθως:

***Κυβερνητικές διατάξεις εξοικονόμησης ενέργειας***

Συμβολισμός: $R_m(k, i, d)$, (R: Regulation), όπου

k : Τύπος διάταξης

i : Αυστηρότητα της διάταξης (υποχρεωτική, συμβουλευτική κ.λ.π.)

d : Διάρκεια επίδρασης (σε ημέρες)

***Απεργίες βιομηχανιών που καταναλώνουν πολύ ενέργεια (πετρέλαιο)***

Συμβολισμός: $S_m(k, i, d)$, (S: Strike), όπου

k : Τύπος απεργίας

i : Ένταση της απεργίας

d : Διάρκεια επίδρασης (σε ημέρες)

***Ειδικές αργίες – διακοπές***

Συμβολισμός: $H_m(k, d)$, (H: Holiday), όπου

k : Κατηγορία διακοπών

d : Διάρκεια (σε ημέρες)

***Ανακοινώσεις μελλοντικών αλλαγών στην τιμή των καυσίμων***

Συμβολισμός: $A_m(p, t_a, t_e)$, (A: Advanced announcement), όπου

p : Ποσοστό αλλαγής της τιμής

$t_a$ : Ημερομηνία ανακοίνωσης

$t_e$ : Ημερομηνία εφαρμογής

Η εμφάνιση ενός υποκειμενικού παράγοντα σε κάποια χρονική στιγμή ονομάζεται **Υποκειμενικό – Κριτικό Γεγονός (Judgmental Event)** και η αναγνώριση αυτών των εμφανίσεων στις ιστορικές παρατηρήσεις οδηγεί στην κατασκευή της **Βάσης Ιστορικών Υποκειμενικών Γεγονότων.**

Ένα υποκειμενικό γεγονός και η επίδραση αυτού στην τιμή κάποιων παρατηρήσεων της χρονοσειράς ονομάζεται **Υποκειμενικό – Κριτικό Σενάριο (Judgmental Case)** και η αναγνώριση αυτών των επιδράσεων στις ιστορικές παρατηρήσεις αλλά και η κατασκευή μελλοντικών πιθανολογούμενων εμφανίσεων υποκειμενικών γεγονότων μαζί με τις επιδράσεις τους οδηγεί στην κατασκευή της **Βάσης Υποκειμενικών Σεναρίων.**

## 2.2 Τεχνητά Νευρωνικά Δίκτυα

Σε αυτό το σημείο, θα παραθέσουμε μία εισαγωγή στα Νευρωνικά Δίκτυα. Αυτό κρίθηκε σκόπιμο, αφού η μοντελοποίηση της υπο-μελέτης κριτικής επέμβασης θα γίνει χρησιμοποιώντας Τεχνητά Νευρωνικά Δίκτυα (Τ.Ν.Δ.).

### 2.2.1 Γενικά

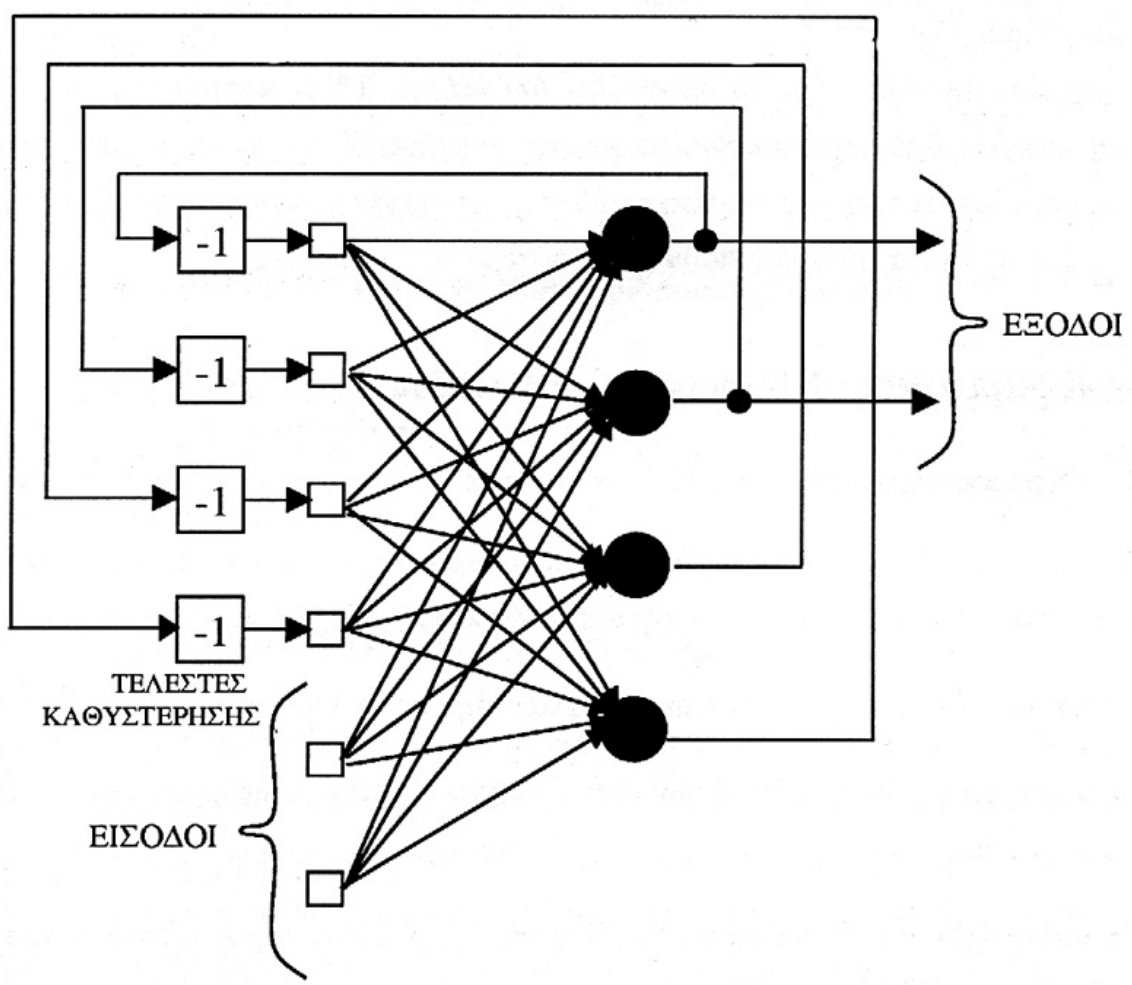
Τα Τεχνητά Νευρωνικά Δίκτυα ανήκουν στο γνωστικό πεδίο της Υπολογιστικής Νοημοσύνης και η χρήση τους έγκειται σε επίλυση προβλημάτων Τεχνητής Νοημοσύνης, χρησιμοποιώντας αριθμητικά μοντέλα.

Παλαιότερα, οι υπολογισμοί στον προγραμματισμό ήταν ακολουθιακοί αλγόριθμοι. Με την εξέλιξη, όμως της Τεχνητής Νοημοσύνης έγιναν βήματα στην προσομοίωση της διαδικασίας υπολογισμών του ανθρώπινου εγκέφαλου.

Οι υπολογισμοί αυτοί: α) είναι κατανεμημένοι και εκτελούνται παράλληλα, β) δεν περιέχουν όλη την πληροφορία για την άρτια λειτουργία του προγράμματος, η «μάθηση» είναι που παρέχει αυτή την δυνατότητα.

Τεχνητό Νευρωνικό Δίκτυο είναι μια αρχιτεκτονική δομή αποτελούμενη από έναν αριθμό τεχνητών νευρώνων. Κάθε νευρώνας χαρακτηρίζεται από εισόδους και εξόδους και υλοποιεί έναν απλό υπολογισμό. Κάθε σύνδεση μεταξύ δύο μονάδων χαρακτηρίζεται από μία τιμή βάρους. Οι τιμές των βαρών των συνδέσεων αποτελούν την αποθηκευμένη γνώση στο δίκτυο και καθορίζουν πλήρως την λειτουργικότητά του. Η έξοδος κάθε νευρώνα καθορίζεται από τον τύπο του νευρώνα, τη διασύνδεση με τους υπόλοιπους και πιθανώς κάποιες εξωτερικές εισόδους.

Στο σχήμα που ακολουθεί, φαίνεται μία δομή ενός νευρωνικού δικτύου με πρόσθια τροφοδότηση (Επαναληπτικό Δίκτυο).




**Σχήμα 1.:** Επαναληπτικό Δίκτυο με ένα κρυμμένο επίπεδο.

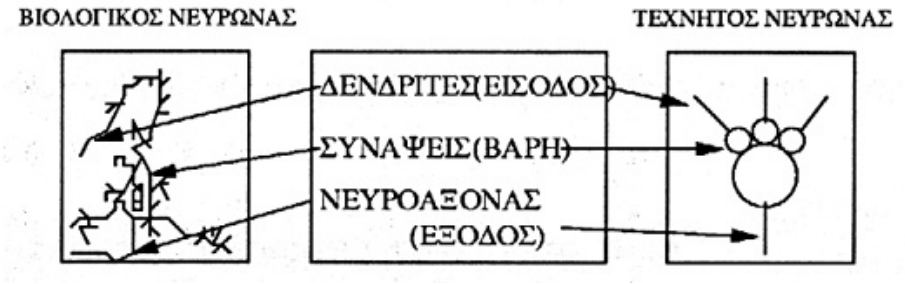
## 2.2.2 Βιολογικοί Νευρώνες – Τεχνητοί Νευρώνες

Είπαμε ότι τα Τ.Ν.Δ. προήλθαν από την προσπάθεια μίμησης των διεργασιών του ανθρώπινου μυαλού. Ας εστιάσουμε περισσότερο στις ομοιότητές αυτές:

Ο εγκέφαλος έχει ιδιότητες όπως προσαρμοστικότητα, ικανότητα αναγνώρισης από τα συμφραζόμενα, ανοχή στα λάθη, μεγάλη χωρητικότητα μνήμης, ικανότητα επεξεργασίας πληροφοριών σε πραγματικό χρόνο. Αυτές οι ιδιότητες προκύπτουν κυρίως από την τοπολογία του εγκεφάλου.

Ο εγκέφαλος αποτελείται από νευρικά κύτταρα συνδεδεμένα μεταξύ τους, οι λεγόμενοι νευρώνες. Τα κύτταρα αυτά κάνουν συνάψεις με άλλους γειτονικούς νευρώνες. Κάθε νευρώνας εκτελεί απλούς ακολουθιακούς υπολογισμούς στα σήματα που δέχεται από τους άλλους νευρώνες. Τα φιλτράρει, τα ενισχύει κατάλληλα και παράγει τελικά ένα σήμα εξόδου το οποίο μεταδίδει μέσο των συνάψεών του στους άλλους νευρώνες. Η επίδραση ενός σήματος σε έναν νευρώνα μπορεί να είναι θετική ή αρνητική. Αυτή η απλή διεργασία εκτελείται παράλληλα σε κάθε νευρώνα, έτσι έχουμε ένα πανίσχυρο ,υπολογιστικά, μοντέλο.

Σε πλήρη αντιστοιχία με το βιολογικό αυτό μοντέλο, ο τεχνητός νευρώνας αποτελείται από εισόδους που εφαρμόζονται με κάποια βάρη και από εξόδους. Το σχήμα 2. περιγράφει την αντιστοιχία:

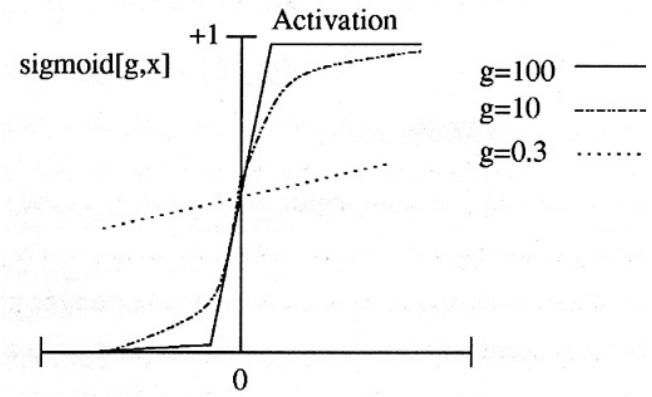



**Σχήμα 2.** Αντιστοιχία βιολογικού-τεχνητού νευρώνα.

Έστω τεχνητός νευρώνας d με συνδέσεις εισόδου χ1,…,χd, και αντίστοιχες τιμές βαρών w1,…,wd. Ο υπολογισμός που εκτελεί ένας τέτοιος νευρώνας χωρίζεται σε δύο στάδια α) υπολογισμός της ενεργοποίησης $u=\Sigma w_i x_i+\theta$, όπου θ η πόλωση του νευρώνα, β) υπολογισμός της εξόδου του νευρώνα περνώντας την ενεργοποίηση u μέσα από μια συνάρτηση ενεργοποίησης f: y=f(u).

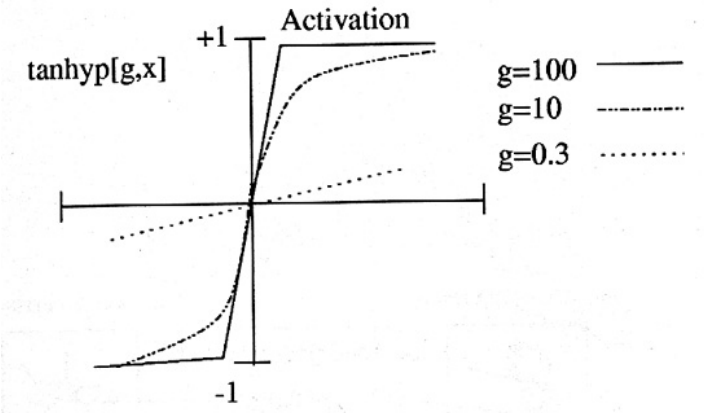
Η συνάρτηση ενεργοποίησης είναι μη γραμμική (εισαγάγει τη «μη γραμμικότητα» στο νευρώνα) και τις περισσότερες περιπτώσεις σιγμοειδής. Βασικοί τύποι σιγμοειδών συναρτήσεων είναι η λογιστική συνάρτηση και η υπερβολική εφαπτομένη.

Στα σχήματα 3. και 4. φαίνονται αντίστοιχα οι συναρτήσεις αυτές.




**Σχήμα 3.:** Λογιστική συνάρτηση f(u)=1/(1+exp(-gu))

**Σχήμα 4.:** Υπερβολική Εφαπτομένη f(u)=tanh(gu)




Η λογιστική συνάρτηση έχει πεδίο τιμών από 0 έως +1, ενώ η υπερβολική εφαπτομένη από –1 έως +1.

Μερικές από τις βασικές ικανότητες του ανθρωπίνου εγκεφάλου που οφείλονται στις πολλές μονάδες επεξεργασίας (Νευρώνες) και στην πολυπλοκότητα των μεταξύ τους διασυνδέσεων είναι οι ακόλουθες:

1. Αποθήκευση εμπειριών όπως η κατηγοριοποίηση ή συσχέτιση των δεδομένων εισόδου. Αυτό-οργάνωση των εμπειριών αυτών.

2. Απόκριση σε νέες εμπειρίες μέσο της συσχέτισής τους με τις αποθηκευμένες.

3. Εκτέλεση προβλέψεων για νέες καταστάσεις σύμφωνα με τις αποθηκευμένες εμπειρίες. Ικανότητα γενίκευσης.

4. Ανοχή στην παραμόρφωση, διαταραχή ή ατέλεια των δεδομένων εισόδου.

5. Ανοχή στις βλάβες. Ακόμα και η απώλεια μερικών νευρώνων αντιμετωπίζεται με κατάλληλη προσαρμογή αυτών που μένουν και πρόσθετη εκπαίδευση.

5. Ο εγκέφαλος φαίνεται να έχει διαθέσιμους νευρώνες, πιθανών αχρησιμοποίητους, έτοιμους προς χρήση. Άρα έχει τη δυνατότητα διαρκώς να μαθαίνει.

7. Η εξέταση του εγκεφάλου δεν παρέχει αρκετές πληροφορίες για την λειτουργία του. Υπάρχει μία αδιαφάνεια στην λειτουργία του.

Πολλές από τις ιδιότητες αυτές υπάρχουν και στα Τ.Ν.Δ. και έτσι τα καθιστούν την αιχμή του δόρατος στις εφαρμογές τεχνητής νοημοσύνης και εξομοίωσης της ανθρώπινης συμπεριφοράς όπως: Επεξεργασία εικόνας / σήματος, , Μηχανική όραση, Αναγνώριση προτύπων, Ιατρικές εφαρμογές, Αμυντικά συστήματα, εφαρμογές Οικονομίας, Έμπειρα συστήματα, Πρόβλεψη χρονοσειρών, επικοινωνία ανθρώπου-υπολογιστή.

## 2.3 Μοντέλο Πρόβλεψης

Είναι πλέον σαφές ότι η χρήση των Τ.Ν.Δ. στις στατιστικές προβλέψεις και τις κριτικές επεμβάσεις γίνεται γιατί αυτά είναι πιο κοντά στον τρόπο σκέψης του ανθρώπου. Βέλτιστο μοντέλο προβλέψεων είναι το ανθρωπομορφικότερο.

Το προτεινόμενο μοντέλο αποτελείται από τα ακόλουθα βήματα:

Α) Κατασκευή Βάσης Υποκειμενικών Παραγόντων (Judgmental Factor Base)

Β) Κατασκευή Βάσης Ιστορικών Υποκειμενικών Γεγονότων (Judgmental Event Base)

Γ) Φιλτράρισμα της χρονοσειράς για απαλοιφή των ιστορικών υποκειμενικών γεγονότων (Monthly Judgmental Instances)

Δ) Κατασκευή Βάσης Υποκειμενικών Σεναρίων (Judgmental Case Base)

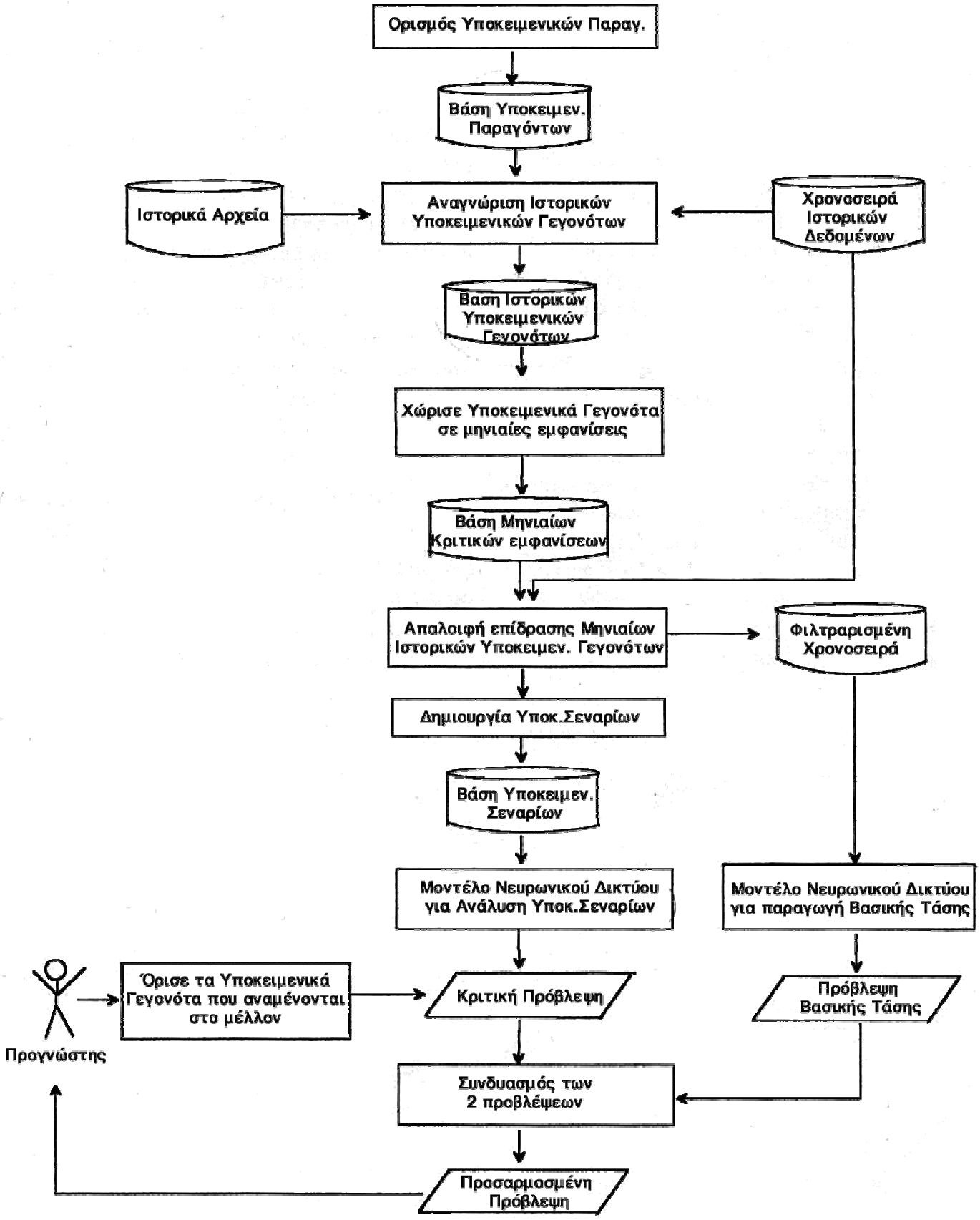
Ε) Κατασκευή Μοντέλου Νευρωνικού Δικτύου για Επεξεργασία Υποκειμενικών Σεναρίων (Judgmental Adjustment)

ΣΤ) Κατασκευή Μοντέλου Νευρωνικού Δικτύου Παραγωγής Βασικής Τάσης (Main Trend Forecasting)

Ζ) Ο Προγνώστης, ορίζει τις εισόδους στο Νευρωνικό Δίκτυο, δηλαδή τα υποκειμενικά σενάρια προς επεξεργασία (Judgmental Occurences)

Η) Παραγωγή προβλέψεων από κάθε Μοντέλο Νευρωνικών Δικτύων και συνδυασμός αυτών (Additive Adjustment of Main Trend Forecast)

Σχηματικά το μοντέλο παρουσιάζεται στην επόμενη σελίδα με τις επεξηγήσεις που παραθέσαμε σε κάθε βήμα.




**Σχήμα 2.** Μοντέλο Πρόβλεψης κατά Jae Kyu Lee και Chang Seon Yum.

Αναλυτική περιγραφή των βασικών βημάτων του αλγορίθμου:

**Α) Κατασκευή Βάσης Υποκειμενικών Παραγόντων (Judgmental Factor Base)**

Η κατασκευή της Βάσης Υποκειμενικών παραγόντων γίνεται ανάλογα την εφαρμογή. Για παράδειγμα, στη περίπτωση της ζήτησης πετρελαίου έχουμε τέσσερις παράγοντες.

Η Βάση περιέχει τους παράγοντες με τις επιμέρους παραμέτρους του κάθε ενός.

**Β) Κατασκευή Βάσης Ιστορικών Υποκειμενικών Γεγονότων (Judgmental Event Base)**

Η Βάση Ιστορικών Υποκειμενικών Γεγονότων γίνεται σε δεύτερη φάση, αφού έχουν οριστεί οι Υποκειμενικοί Παράγοντες.

Τότε, με είσοδο μία Βάση Ιστορικών Γεγονότων, κατασκευάζουμε την Βάση Ιστορικών Υποκειμενικών Γεγονότων τις οποίας οι καταχωρίσεις είναι τα Ιστορικά Υποκειμενικά Γεγονότα που συνέβησαν σε μία χρονική στιγμή (π.χ. 1 μήνας) με επίδρασή στο υπό-μελέτη μέγεθος (π.χ. Οκτ.1998 | $H_{98}(1,4)$, α(22.7,12) ).

**Γ) Φιλτράρισμα της χρονοσειράς για απαλοιφή των ιστορικών υποκειμενικών γεγονότων (Monthly Judgmental Instances)**

Από τα δεδομένα της χρονοσειράς αφαιρούμε την επίδραση των Υποκειμενικών Γεγονότων για την εκάστοτε περίοδο όπου αυτά συμβαίνουν. Το αποτέλεσμα είναι η φιλτραρισμένη χρονοσειρά που περιέχει μόνο την βασική τάση.

Υπάρχουν 2 μέθοδοι φιλτραρίσματος: α) με χρήση εξωτερικής φόρμουλας υπολογισμού της επίδρασης των Υποκειμενικών Γεγονότων π.χ. η επίδραση των «διατάξεων εξοικονόμησης ενέργειας» στο μέγεθος «ζήτηση πετρελαίου» μπορεί να υπολογιστεί από την παρακάτω σχέση, αν είναι διαθέσιμα τα απαιτούμενα στοιχεία:

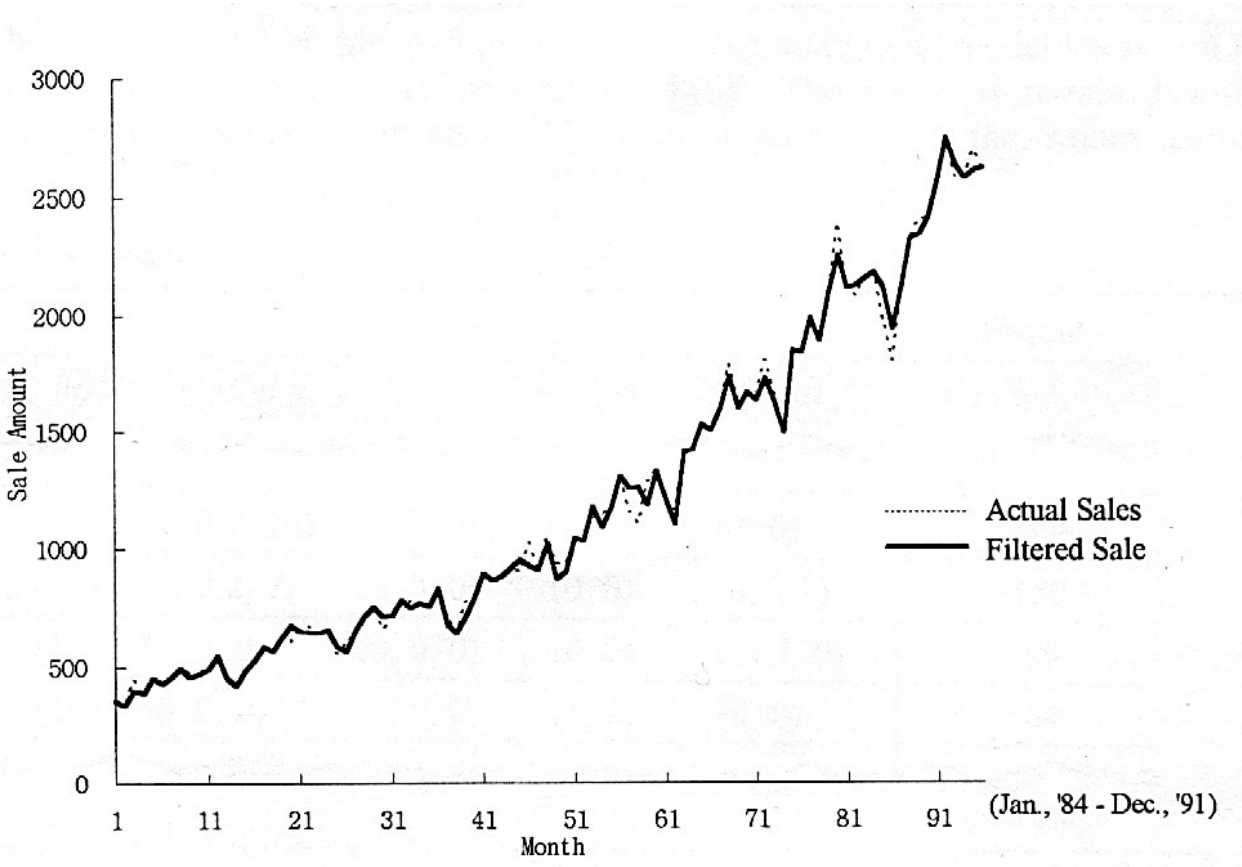
**Επίδραση** = (**αριθμός αυτοκίνητων** που αναφέρεται η διάταξη**)*(μέση ημερήσια κατανάλωση** ανά αυτοκίνητο)*(**διάρκεια δράσης** της διάταξης)

Από την χρονοσειρά, αφαιρούμε την υπολογισμένη επίδραση, κατασκευάζοντας τελικά την καμπύλη της Βασικής Τάσης, καθώς και μία βάση με τα Ιστορικά Υποκειμενικά Γεγονότα και την επίδρασή τους σε συγκεκριμένη χρονική περίοδο.

β) με την εκτίμηση που κάνει ο προγνώστης. Έχοντας στην διάθεσή του τα ιστορικά στοιχεία, ο προγνώστης, ορίζει για κάθε Υποκειμενικό γεγονός που συνέβη, πόση ήταν η επίδρασή του.

Όπως και παραπάνω, αφαιρείται η εκάστοτε επίδραση από την χρονοσειρά και κατασκευάζουμε την βασική τάση και την Βάση με τα Ιστορικά Υποκειμενικά Γεγονότα και την επίδρασή τους.

Παραθέτουμε ένα παράδειγμα χρονοσειράς, η οποία έχει φιλτραριστεί από τα Υποκειμενικά Γεγονότα στο παρακάτω σχήμα 3. Παρατηρούμε ότι η χρονοσειρά έχει «εξομαλυνθεί» αφού έχουν αφαιρεθεί τα Υπ.Γεγ. Τα Γεγονότα αυτά, μαζί με την επίδρασή τους, θα δημιουργήσουν την Βάση Δεδομένων που εξετάζουμε στο Δ.




**Σχήμα 3**. Χρονοσειρά Φιλτραρισμένη από τα Υποκειμενικά Γεγονότα.

**Δ) Κατασκευή Βάσης Υποκειμενικών Σεναρίων (Judgmental Case Base)**

Με την διεργασία Γ. παράγεται, όπως είδαμε, και μία βάση με τα Υποκειμενικά Γεγονότα που συνέβησαν μία χρονική περίοδο, μαζί με την επίδρασή τους στην χρονοσειρά. Αυτή η Βάση Δεδομένων , ονομάζεται **Βάση Υποκειμενικών Σεναρίων** και η μορφή των εγγραφών της είναι: π.χ. Σενάριο 34: Οκτ.1998 | $H_{98}(1,4)$, α(22.7,12) | -150.

Στο παρακάτω σχήμα 4. φαίνεται η δομή μίας τέτοιας βάσης δεδομένων. Η Μηνιαία επίδραση (Monthly Impact) είναι η επίδραση που έχει ένας συνδυασμός Υποκειμενικών Γεγονότων αναλυμένων στους επιμέρους παράγοντές τους, σε συγκεκριμένη χρονική περίοδο (m).

| Judgmental Cases | Inputs | | | | | Output |
|---|---|---|---|---|---|---|
| | $m$, | $r_m(k, i, d_m)$, | $s_m(k, i, d_m)$, | $h_m(k, d_m)$, | $a_m(p_m, d_m)$ | Monthly Impact |
| Case 20 | (57) | (1, 0.2, 16) | (0, 0, 0) | (2, 1) | (0, 0) | -106 |
| Case 21 | (58) | (1, 0.2, 7) | (0, 0, 0) | (0, 0) | (4.7, 9) | -150 |
| Case 22 | (59) | (0, 0, 0) | (0, 0, 0) | (0, 0) | (-4.7, 9) | 99 |
| Case 23 | (60) | (0, 0, 0) | (0, 0, 0) | (1, 2) | (0, 0) | 64 |

**Σχήμα 4.:** Τμήμα Βάσης Δεδομένων Υποκειμενικών Σεναρίων

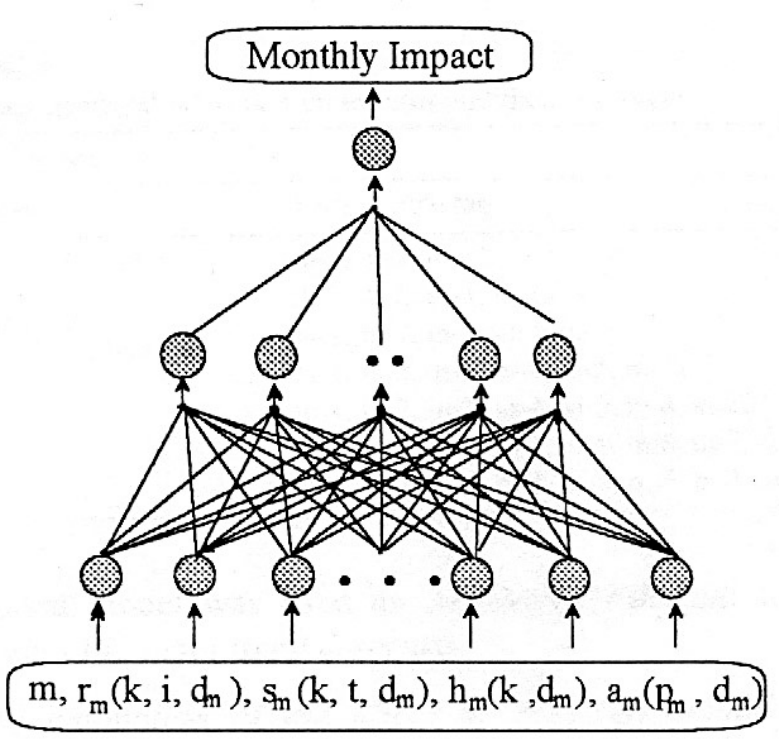
**Ε) Κατασκευή Μοντέλου Νευρωνικού Δικτύου για Επεξεργασία Υποκειμενικών Σεναρίων (Judgmental Adjustment)**

Η Βάση Δεδομένων με τα Υποκειμενικά Σενάρια είναι η είσοδος σε ένα Νευρωνικό Δίκτυο, κατάλληλα επιλεγμένο, το οποίο αφού **εκπαιδευτεί κατάλληλα**, θα μπορεί να **προβλέπει** την μελλοντική επίδραση που θα έχει ένας ή συνδυασμός υποκειμενικών παραγόντων που θα συμβούν μία χρονική περίοδο, στην χρονοσειρά.

**Το Μοντέλο Νευρωνικού Δικτύου θα εκπαιδευτεί σαν «προγνώστης επίδρασης Υποκειμενικών Σεναρίων»** χρησιμοποιώντας τη Βάση Δεδομένων. Η εκπαίδευσή του θα το εξειδικεύσει στην υπό-μελέτη χρονοσειρά με τους Υποκειμενικούς παράγοντες μόνο, που υπάρχουν στην Β.Δ.

Στο κεφάλαιο **3.** ασχολούμαστε αναλυτικά με την κατασκευή ενός τέτοιου μοντέλου Νευρωνικού Δικτύου και την λειτουργία του.

Το σχήμα 5. δείχνει ένα τέτοιο Νευρωνικό Δίκτυο με εισόδους m=μήνας και τα Υποκειμενικά Γεγονότα αναλυμένα στις παραμέτρους των. Το "Monthly Impact" είναι η επίδραση των Υποκειμενικών Γεγονότων την εκάστοτε χρονική περίοδο (μήνας).




**Σχήμα 5**. Νευρωνικό Δίκτυο επεξεργασίας Υποκειμενικών Σεναρίων

**ΣΤ) Κατασκευή Μοντέλου Νευρωνικού Δικτύου Παραγωγής Βασικής Τάσης (Main Trend Forecasting)**

Η φιλτραρισμένη χρονοσειρά από το βήμα Γ. θα χρησιμοποιηθεί σαν είσοδος σε ένα δεύτερο μοντέλο Νευρωνικού Δικτύου το οποίο θα παράγει την Βασική Τάση (χρονοσειρά χωρίς την επίδραση των Υποκειμενικών Γεγονότων).

Το Μοντέλο αυτό θα πρέπει να επιλεγεί κατάλληλα και να **εκπαιδευτεί** ειδικευμένα σαν **«προγνώστης Βασικής Τάσης της Χρονοσειράς»** και ακολούθως θα **προλέγει** τις μελλοντικές τιμές της με βάση τις παρελθοντικές.

Στο σχήμα 6. φαίνονται οι είσοδοι (inputs) που χρειάζεται ένα τέτοιο μοντέλο νευρωνικού, προκειμένου να παράγει την πρόβλεψη (output). Παρατηρούμε ότι οι είσοδοι είναι προσεκτικά επιλεγμένοι ώστε να περιλαμβάνουν τις γειτονικές τιμές της χρονοσειράς (m-1, m-2, …) καθώς όμως και τις

απομακρυσμένες. Έτσι εξασφαλίζουμε ότι η πρόβλεψη Βασικής Τάσης (m) θα είναι το δυνατόν ακριβέστερη.

Η Βάση Δεδομένων αυτή, αποτελεί την Βάση Δεδομένων των Σετ Εκπαίδευσης με τα οποία θα «μάθει» το Νευρωνικό να προβλέπει.

| Patterns | Inputs | Output |
|---|---|---|
| | Filtered Data Set at (*m-12, m-6, m-5, m-4, m-3, m-2, m-1*) | Filtered Data at (*m*) |
| Pattern 55 | (885, 872, 902, 1039, 1031, 1176, 1088) | 1,174 |
| Pattern 56 | (917, 902, 1039, 1031, 1176, 1088, 1174) | 1,306 |
| Pattern 57 | (953, 1039, 1031, 1176, 1088, 1174, 1306) | 1,259 |
| Pattern 58 | (928, 1031, 1176, 1088, 1174, 1306, 1259) | 1,257 |
| Pattern 59 | (908, 1176, 1088, 1174, 1306, 1259, 1257) | 1,182 |
| Pattern 60 | (1018, 1088, 1174, 1306, 1259, 1257, 1182) | 1,326 |
| Pattern 61 | (872, 1174, 1306, 1259, 1257, 1182, 1326) | 1,204 |
| Pattern 62 | (902, 1306, 1259, 1257, 1182, 1326, 1204) | 1,097 |
| Pattern 63 | (1039, 1259, 1257, 1182, 1326, 1204, 1097) | 1,408 |
| Pattern 64 | (1301, 1257, 1182, 1326, 1204, 1097, 1408) | 1,419 |
| Pattern 65 | (1176, 1182, 1326, 1204, 1097, 1408, 1419) | 1,529 |
| Pattern 66 | (1088, 1326, 1204, 1097, 1408, 1419, 1529) | 1,499 |

**Σχήμα 6.:** Είσοδοι και έξοδοι Μοντέλου Νευρωνικού Δικτύου για την πρόβλεψη της Βασικής Τάσης. Τμήμα της Βάσης Δεδομένων Σετ Εκπαίδευσης

**Ζ) Ο Προγνώστης, ορίζει τις εισόδους στο Νευρωνικό Δίκτυο, δηλαδή τα υποκειμενικά σενάρια προς επεξεργασία (Judgmental Occurrences)**

Στο Μοντέλο πρόβλεψης που παραθέτουμε, απαραίτητη είναι και η ύπαρξη ενός εξωτερικού προγνώστη, ο οποίος ορίζει τα Υποκειμενικά Γεγονότα που πρόκειται να συμβούν στο μέλλον.

Ο εξωτερικός προγνώστης δεν είναι απαραίτητο να είναι πρόσωπο, μπορεί να είναι και ένας συνδυασμός αλγορίθμων όπου ο ένας π.χ. υπολογίζει τις αργίες που πρόκειται να συμβούν με βάση το ημερολόγιο, ο άλλος μαζεύει τις ανακοινώσεις του υπουργείου συγκοινωνιών για τα αυτοκίνητα τον τύπο, τις κατατάσσει, τις αξιολογεί και υπολογίζει τις μελλοντικές εξαγγελίες της κυβέρνησης. κ.α.

**Η) Παραγωγή προβλέψεων από κάθε Μοντέλο Νευρωνικών Δικτύων και συνδυασμός αυτών (Additive Adjustment of Main Trend Forecast)**

Από τα Ε. (με κατάλληλη είσοδο για τα Υποκειμενικά Γεγονότα που θα συμβούν Ζ.) και ΣΤ. (με είσοδο την παρελθοντική Βασική Τάση) έχουμε δύο δεδομένα για το μέλλον: α) την επίδραση που θα έχουν οι μελλοντικοί Υποκειμενικοί παράγοντες και β) την τιμή της Βασικής Τάσης στο μέλλον. Με την άθροιση αυτών, έχουμε την ολική πρόβλεψη για την υπό-μελέτη χρονοσειρά.

Από το παραπάνω μοντέλο πρόβλεψης κατά Jae Lee και Chang Seon Yum εμείς θα ασχοληθούμε μόνο με την ανάλυση των υποκειμενικών σεναρίων και την παραγωγή κριτικής πρόβλεψης και όχι με την πρόβλεψη της βασικής τάσης της χρονοσειράς και τον συνδυασμό των δύο προβλέψεων. Θα προσαρμόσουμε δηλαδή το αρχικό μοντέλο έτσι ώστε να περιλαμβάνει μόνο τα βήματα: Α, Β, Δ, Ε, Ζ και Η(έχοντας υπολογίσει μόνο την επίδραση που θα έχουν οι μελλοντικοί υποκειμενικοί παράγοντες).

Κεφάλαιο

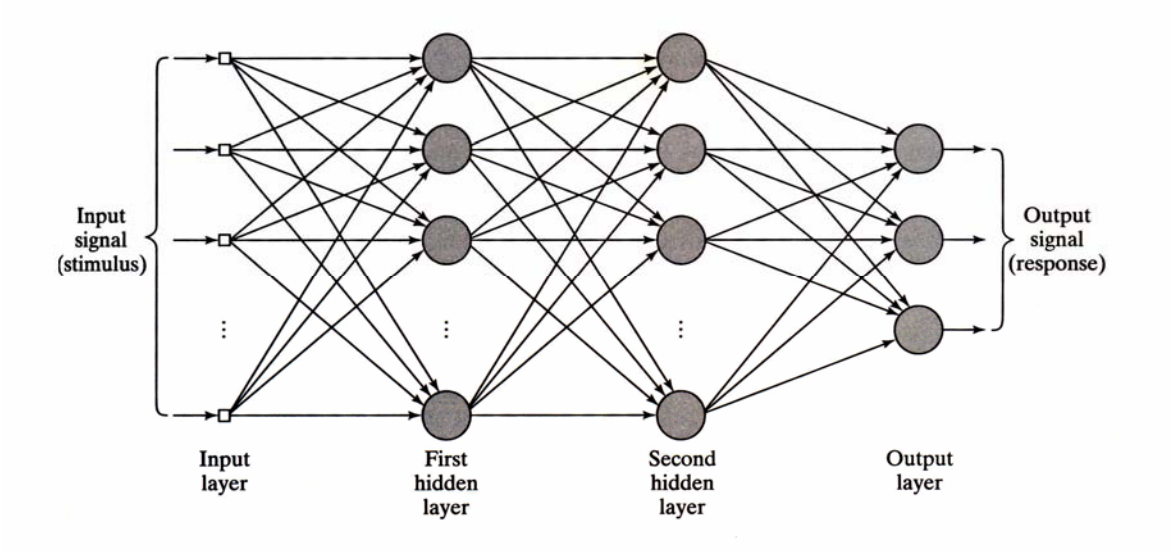
3

# Μη επαναληπτικά Πολυστρωματικά Νευρωνικά Δίκτυα

## 3.1 Εισαγωγή

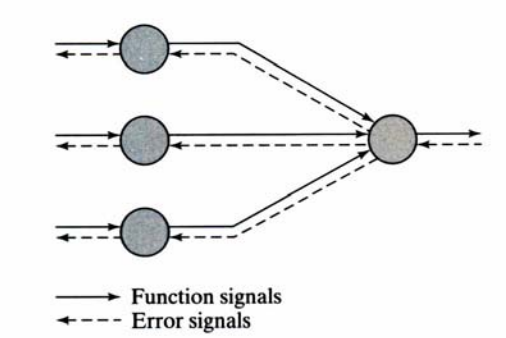
Στο κεφάλαιο αυτό θα ασχοληθούμε με τα μη επαναληπτικά νευρωνικά δίκτυα πολλών επιπέδων. Θα παραθέσουμε όλα τα θεωρητικά στοιχεία που αφορούν τα τελευταία και έχουν χρησιμοποιηθεί για την κατασκευή του Πληροφοριακού συστήματος. Συνοπτικά, θα ασχοληθούμε με τον αλγόριθμο Back Propagation ο οποίος χρησιμοποιείται για την εκπαίδευση του δικτύου καθώς και με ευριστικές μεθόδους που εφαρμόστηκαν σε αυτόν με σκοπό την καλύτερη λειτουργία του. Θα αναφερθούμε στον αλγόριθμο Optimal Brain Surgeon ο οποίος χρησιμοποιείται για το 'κλάδεμα' πολύπλοκων δικτύων και επιτυγχάνει την περικοπή άχρηστων συνάψεων και θα αναφέρουμε και πολλά άλλα θεωρητικά στοιχεία τα οποία λήφθηκαν υπόψη για την κατασκευή του πληροφοριακού συστήματος, όπως για παράδειγμα η μέθοδος Cross Validation σαν μέθοδος πρόωρου τερματισμού και επιλογής καλύτερου μοντέλου ή η μέθοδος Leave One Out, η οποία είναι μια παραλλαγή της πρώτης.

Στο σχήμα 1 στην επόμενη σελίδα φαίνεται το αρχιτεκτονικό διάγραμμα ενός μη επαναληπτικού πολυστρωματικού νευρωνικού δικτύου.




**Σχήμα 1.** Αρχιτεκτονικό διάγραμμα ενός μη επαναληπτικού πολυστρωματικού νευρωνικού δικτύου

Το συγκεκριμένο δίκτυο αποτελείται από n εισόδους, δύο κρυμμένα επίπεδα και 3 εξόδους. Στο σχήμα 2 φαίνεται η διάδοση των δύο σημάτων κατά τη διαδικασία της εκπαίδευσης. Το πρώτο σήμα είναι το σήμα εισόδου το οποίο διαδίδεται από την αρχή ως τους νευρώνες εξόδου και το δεύτερο τα σήμα σφάλματος, το οποίο διαδίδεται ανάστροφα.




**Σχήμα 2.** Τα δύο βασικά σήματα διάδοσης ενός μη επαναληπτικού πολυστρωματικού νευρωνικού δικτύου

Κάθε νευρώνας του δικτύου είναι σχεδιασμένος να κάνει δύο λειτουργίες.

1). Τον υπολογισμό του σήματος που θα εμφανιστεί στην έξοδό του ως συνάρτηση του σήματος που φτάνει στην είσοδό του.

2). Τον προσεγγιστικό υπολογισμού του διανύσματος κλίσης της επιφάνειας σφάλματος σε σχέση με τις συνάψεις που συνδέονται στην είσοδο του νευρώνα.

Παρακάτω παρατίθεται η σημειογραφία που έχει χρησιμοποιηθεί στα επόμενα κεφάλαια.

- Οι δείκτες i, j και k αναφέρονται σε νευρώνες του δικτύου.
- Τη χρονική στιγμή n το παράδειγμα με αριθμό σειράς n παρουσιάζεται στο δίκτυο
- Το σύμβολο $\mathrm{E}(n)$ αναφέρεται στο στιγμιαίο άθροισμα των τετραγώνων των λαθών των νευρώνων του επιπέδου εξόδου στην επανάληψη n. Ο μέσος όρος όλων των $\mathrm{E}(n)$ (για όλο δηλαδή το σετ εκπαίδευσης) συμβολίζεται με $\mathrm{E}_{av}$ και ονομάζεται μέση ενέργεια λάθους.

$\mathrm{E}(n) = \frac{1}{2} \cdot \sum_{j \in C} e_j^2(n)$ όπου το σύνολο C περιλαμβάνει όλους τους νευρώνες του επιπέδου εξόδου.

$\mathrm{E}_{av} = \frac{1}{N} \cdot \sum_{n=1}^{N} E(n)$ όπου το N είναι ο αριθμός των παραδειγμάτων που χρησιμοποιούνται για την εκπαίδευση του δικτύου.

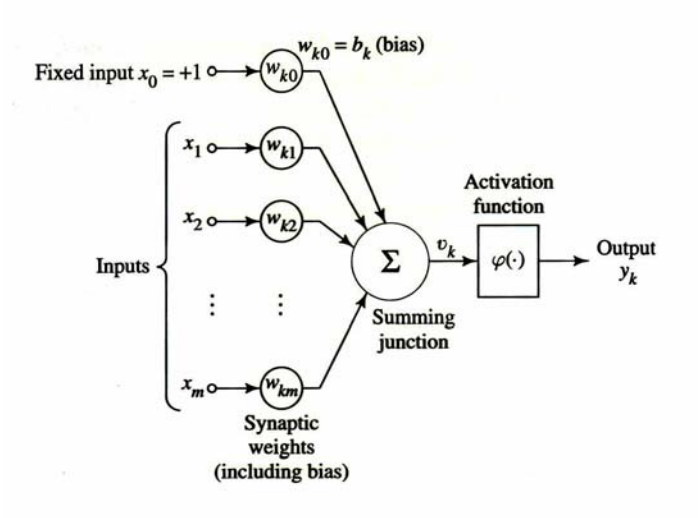
- Το σύμβολο $e_j(n)$ αναφέρεται στο σήμα λάθους για τον νευρώνα j στην επανάληψη n.
- Το σύμβολο $d_j(n)$ αναφέρεται στην επιθυμητή έξοδο του νευρώνα j στην επανάληψη n και χρησιμοποιείται για τον υπολογισμό του $e_j(n)$.
- Το σύμβολο $y_j(n)$ αναφέρεται στο σήμα που εμφανίζεται στην έξοδο του νευρώνα j στην επανάληψη n. Ισχύει: $e_j(n) = d_j(n) - y_j(n)$
- Το σύμβολο $w_{ji}(n)$ αναφέρεται στο βάρος της σύναψης που συνδέει την έξοδο του νευρώνα i με την είσοδο του νευρώνα j στην επανάληψη n. Η διόρθωση που εφαρμόζεται στο βάρος αυτής της σύναψης στην επανάληψη n συμβολίζεται με $\Delta w_{ji}(n)$.
- Το σταθμισμένο άθροισμα όλων των εισόδων του νευρώνα j στην επανάληψη n μαζί με την πόλωση του νευρώνα αποτελεί το σήμα εισόδου του νευρώνα και συμβολίζεται με $u_j(n)$
- Η συνάρτηση ενεργοποίησης που εκφράζει τη σχέση εισόδου εξόδου του νευρώνα j συμβολίζεται με $\phi_j(\bullet)$. Ισχύει: $y_j(n) = \phi_j(u_j(n))$.

● Η πόλωση του νευρώνα j συμβολίζεται με $b_j$. Αυτή προσομοιάζεται με μία σύναψη με βάρος $w_{j0} = b_j$ με σταθερή είσοδο ίση με +1. (Σχήμα 3)

● Το στοιχείο i του διανύσματος εισόδου συμβολίζεται με $x_i(n)$

● Ο ρυθμός εκπαίδευσης συμβολίζεται με η.

● Το σύμβολο $m_l$ συμβολίζει τον αριθμό των νευρώνων στο επίπεδο l. l=0,1, …, L όπου L είναι το 'βάθος' του δικτύου. Έτσι το $m_0$ συμβολίζει το μέγεθος του επιπέδου εισόδου, το $m_1$ το μέγεθος του πρώτου κρυμμένου επιπέδου και το $m_L$ το μέγεθος του επιπέδου εξόδου. Μερικές φορές χρησιμοποιείται και ο συμβολισμός $M = m_L$.




**Σχήμα 3.** Μοντέλο νευρώνα σε ένα μη επαναληπτικό πολυστρωματικό νευρωνικό δίκτυο

## 3.2 Ο αλγόριθμος Back Propagation

### 3.2.1 Γενικά

Ο αλγόριθμος Back Propagation είναι ίσως σήμερα από τους πιο δημοφιλής αλγορίθμους εκπαίδευσης, εξαιτίας κυρίως της εύκολης υλοποίησής του και βέβαια της παραλληλίας που παρουσιάζει στην εκτέλεση των βημάτων του. Βασίζεται κυρίως στη μέθοδο Steepest Descent η οποία δίνει λύση στο εξής πρόβλημα: Δεδομένου ότι έχουμε ένα αρχικό διάνυσμα $w(0)$, ποια είναι η ακολουθία διανυσμάτων $w(1)$, $w(2)$, ..., τέτοια ώστε σε κάθε επανάληψη του αλγορίθμου να προκαλεί μείωση σε μια συνεχώς διαφορίσιμη συνάρτηση κόστους $E(w)$, δηλαδή να ισχύει $E(w(n+1)) < E(w(n))$. Σύμφωνα με τη μέθοδο Steepest Descent η διόρθωση του διανύσματος που πρέπει να γίνει σε κάθε επανάληψη είναι $\Delta w(n) = w(n+1) - w(n) = -n \cdot g(n)$ όπου $g(n) = \nabla E(w)$.Σε αυτή ακριβώς την αρχή βασίζεται και ο Back Propagation. Εδώ η συνάρτηση κόστους είναι το τελικό σφάλμα $E_{av}$ το οποίο εξαρτάται από τα βάρη των συνάψεων του δικτύου τα οποία αποτελούν το διάνυσμα $w$. Σε επόμενες ενότητες θα φανεί αυτή ακριβώς η σχέση όπου και θα δώσουμε αναλυτικά τα βήματα του αλγορίθμου.

Κάθε επανάληψη του αλγορίθμου εκτελείται σε δύο φάσεις. Στην πρώτη γίνεται η διάδοση του σήματος από το επίπεδο εισόδου μέχρι το επίπεδο εξόδου όπου και υπολογίζεται το σφάλμα από την επιθυμητή έξοδο. Στη δεύτερη φάση υπολογίζεται και εφαρμόζεται η διόρθωση που πρέπει να γίνει στα βάρη των συνάψεων προκειμένου να μειωθεί το σφάλμα της εξόδου.

### 3.2.2 Σύνοψη του αλγορίθμου

Στην επόμενη σελίδα φαίνονται συνοπτικά τα βήματα του Back Propagation. Περαιτέρω πληροφορίες για το πως καταλήξαμε στα παρακάτω αποτελέσματα έχουν παραλειφθεί μιας και δεν είναι το αντικείμενο εξέτασης της συγκεκριμένης εργασίας.

**1. Αρχικοποίηση**

**α).** Αρχικοποίηση των συνάψεων και της πόλωσης του κάθε νευρώνα από μία κανονική κατανομή με μέση τιμή μηδέν και διασπορά τέτοια ώστε η τυπική απόκλιση των εισόδων του κάθε νευρώνα να βρίσκεται ενδιάμεσα του κορεσμένου και του γραμμικού τμήματος της συνάρτησης ενεργοποίησης. Η παραπάνω συνθήκη ικανοποιείται αν θέσουμε τη ζητούμενη διασπορά ίση με $\sigma_w = m^{-1/2}$, όπου m είναι ο αριθμός των συνάψεων εισόδου του κάθε νευρώνα.

**β).** Χρήση **σειριακής μεθόδου** παρουσίασης των παραδειγμάτων εκπαίδευσης. Ορίζουμε εποχή την παρουσίαση όλου του σετ εκπαίδευσης στο δίκτυο. Οι διόρθωση των συνάψεων του δικτύου θα γίνεται σε κάθε επανάληψη μετά από την παρουσίαση κάθε δείγματος. Η εκπαίδευση μπορεί να διαρκέσει αρκετές εποχές.

**γ).** Χρήση του **υπερβολικού ημίτονου** ως συνάρτηση ενεργοποίησης: $\phi(u) = a \cdot \tanh(b \cdot u)$ με α=1.7159 και b=2/3.

**δ). Γραμμικός μετασχηματισμός του των διανυσμάτων εξόδου** έτσι ώστε η μέγιστη τιμή να παίρνει την τιμή $\max Value = d_j = a - \varepsilon$ και η ελάχιστη την $\min Value = -d_j = -a + \varepsilon$ με ε=0.7159.

**ε). Κανονικοποίηση της εισόδου.** Αφαίρεση τις μέσης τιμής από τα διανύσματα εκπαίδευσης και όμοιος γραμμικός μετασχηματισμός με τα διανύσματα εξόδου.

**2). Παρουσίαση του δικτύου με μία εποχή δειγμάτων εκπαίδευσης**

**α).** Τυχαία αρχικοποίηση της σειράς παρουσίασης των δειγμάτων του σετ εκπαίδευσης.

Για κάθε παράδειγμα:

**2.1). Εμπρόσθιος υπολογισμός**

**α).** $w_{j0}^{(l)}(n) = b_j^{(l)}(n),\ y_0^{(l-1)}(n) = +1,\ y_j^{(0)}(n) = x_j(n)$

**β).** $u_j^{(l)}(n) = \sum_{i=0}^{m} w_{ji}^{(l)}(n) \cdot y_i^{(l-1)}$

**γ).** $y_j^{(l)}(n) = \phi_j(u_j^{(l)}(n))$

**δ).** $e_j(n) = d_j(n) - y_j^{(L)}(n)$

**2.2). Προς τα πίσω υπολογισμός**

**α).** $\delta_j^{(l)}(n) = \begin{cases} e_j^{(l)}(n) \cdot \phi_j'(u_j^{(L)}(n)), \ j \in output\ layer \\ \phi_j'(u_j^{(l)}(n)) \cdot \sum_k \delta_k^{(l+1)}(n) \cdot w_{kj}^{(l+1)}(n), \ j \notin output\ layer \end{cases}$

**β).** $w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + a \cdot \Delta w_{ji}^{(l)}(n-1) + \eta \cdot \delta_j^{(l)}(n) \cdot y_i^{(l-1)}(n),\ 0 < |a| < 1$

**γ).** $\Delta w_{ji}^{(l)}(n) = w_{ji}^{(l)}(n+1) - w_{ji}^{(l)}(n)$

**3). Επανάληψη**

**α).** Επανάληψη του βήματος 2 παρουσιάζοντας νέες εποχές με παραδείγματα στο δίκτυο μέχρι κάποιο κριτήριο τερματισμού (ενότητα 3.4 – Cross Validation).

**4). Τέλος**

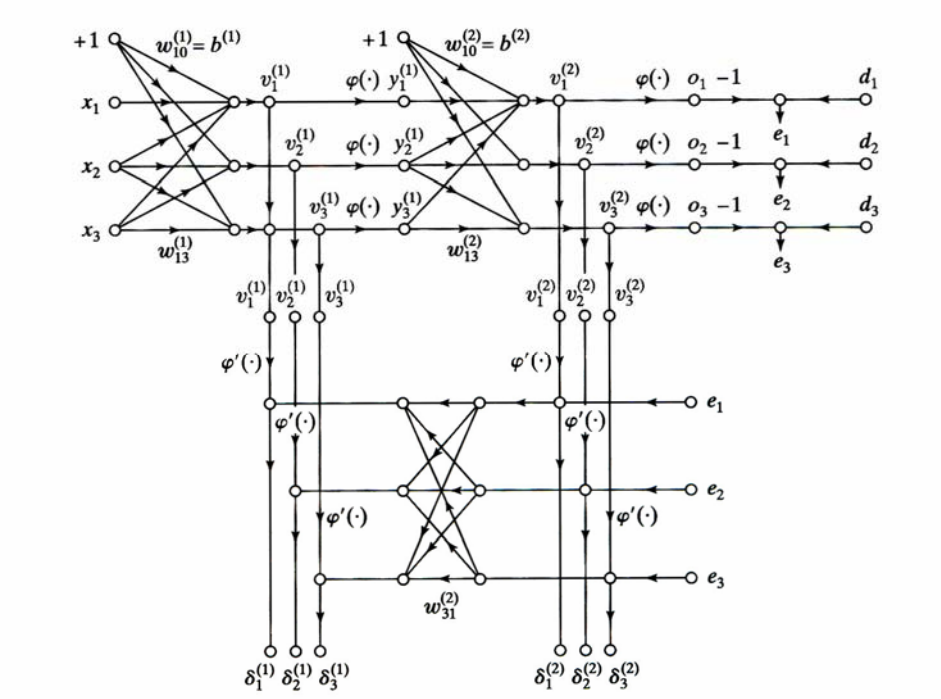
### 3.2.3 Παρατηρήσεις

Θα πρέπει να σημειωθεί ότι ο παραπάνω αλγόριθμος είναι ο Back Propagation με αρκετές προσθήκες οι οποίες βελτιώνουν τη λειτουργία του. Αυτές είναι κυρίως η αρχικοποίηση των βαρών των συνάψεων με τον τρόπο που περιγράφτηκε, ο γραμμικός μετασχηματισμός της εισόδου και της εξόδου, η χρησιμοποίηση του υπερβολικού ημίτονου ως συνάρτησης ενεργοποίησης και η διαδοχική διόρθωση των βαρών μετά την παρουσίαση κάθε παραδείγματος.

Επίσης είναι πολύ σημαντικό να σημειώσουμε την προσθήκη που έγινε στον τύπο που περιγράφει τη διόρθωση των βαρών των συνάψεων σε σχέση με τον αντίστοιχο τύπο της μεθόδου Steepest Descent. Παρατηρούμε ότι στον τύπο : $w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + a \cdot \Delta w_{ji}^{(l)}(n-1) + \eta \cdot \delta_j^{(l)}(n) \cdot y_i^{(l-1)}(n),\ 0 < |a| < 1$ υπάρχει ένας επιπλέον όρος $+ a \cdot \Delta w_{ji}^{(l)}(n-1)$. Ο όρος αυτός εκφράζει κάποια μορφή μνήμης των διαδοχικών διορθώσεων που γίνονται σε κάθε σύναψη. Η σημαντικότητα του όρου είναι πολύ μεγάλη γιατί βοηθάει τον αλγόριθμο να συγκλίνει γρήγορα προς κάποιο τοπικό ελάχιστο. Αν για παράδειγμα μετά την τυχαία αρχικοποίηση των συνάψεων του δικτύου, βρεθούμε σε κάποιο σημείο της επιφάνειας σφάλματος με

πολύ μικρή κλίση, τότε ο αλγόριθμος θα συγκλίνει πολύ αργά στο τοπικό ελάχιστο, αφού η ταχύτητα σύγκλισης εξαρτάται από τον παράγοντα $\eta \cdot \delta_j^{(l)}(n) \cdot y_i^{(l-1)}(n)$ που εκφράζει την κλίση της επιφάνειας σφάλματος.. Σε αυτή την περίπτωση τώρα όμως ο όρος $\Delta w_{ji}^{(l)}(n-1)$ θα έχει συνέχεια σταθερό πρόσημο με αποτέλεσμα να συνεισφέρει όλο και περισσότερο στην ενίσχυση της διόρθωσης λειτουργώντας κατά κάποιο τρόπο ολοκληρωτικά. Αντίθετα αν η επιφάνεια σφάλματος είναι απότομη με πολλές εναλλαγές ο όρος $\Delta w_{ji}^{(l)}(n-1)$ θα έχει εναλλασσόμενο πρόσημο και η συνεισφορά του θα είναι πολύ μικρότερη. Συνοψίζοντας ο πρόσθετος αυτός όρος εμποδίζει το δίκτυο να παγιδευτεί σε κάποιο σημείο της επιφάνειας με μικρή κλίση επιταχύνοντας τη σύγκλιση του αλγορίθμου κάθε φορά προς το τοπικό ελάχιστο. Ο νέος κανόνας διόρθωσης των βαρών που προκύπτει με την προσθήκη του όρου αυτού ονομάζεται generalized delta rule.

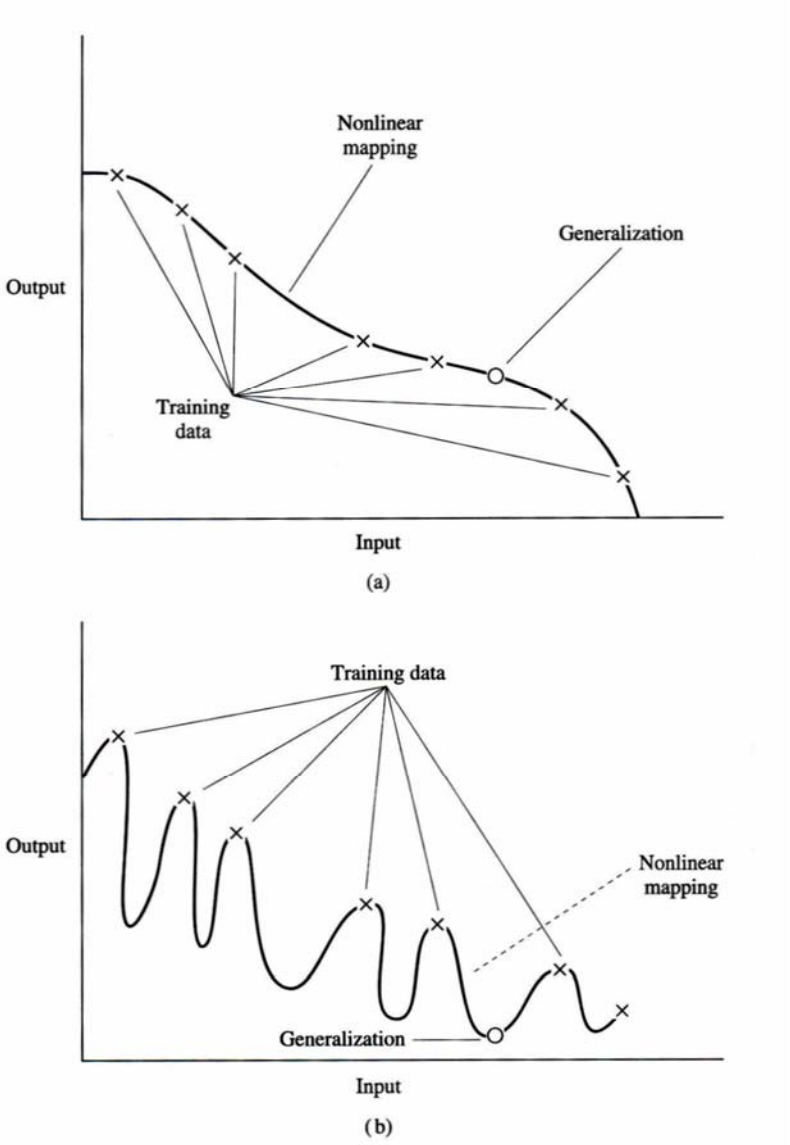
Οι τιμές που έχουμε επιλέξει για τις παραμέτρους η και α είναι 0.1 και 0.3 αντίστοιχα. Αυτές θεωρούνται αρκετά μικρές με αποτέλεσμα να εμποδίζουν τη γρήγορη σύγκλιση, αλλά να επιτυγχάνουν ομαλότερη σύγκλιση προς το τοπικό ελάχιστο. Στην περίπτωσή μας αυτό δε μας ενοχλεί ιδιαίτερα μιας και δεν πρόκειται για κάποια εφαρμογή που απαιτεί επεξεργασία σε πραγματικό χρόνο. Αντίθετα μας ικανοποιεί το γεγονός ότι η σύγκλιση γίνεται ομαλά και σταθερά.




**Σχήμα 4.** Διάγραμμα ροής σήματος του αλγορίθμου Back Propagation. Στο πάνω μέρος φαίνεται η εμπρόσθια διάδοση και στο κάτω η προς τα πίσω διάδοση.

## 3.3 Η έννοια της γενίκευσης

Γενίκευση ονομάζουμε την ιδιότητα των νευρωνικών δικτύων να μπορούν να δίνουν αποτελέσματα πολύ κοντά στα πραγματικά, για δεδομένα εισόδου που δεν έχουν ξαναδεί. Δηλαδή αν υποθέσουμε ότι εκπαιδεύουμε ένα νευρωνικό δίκτυο με ένα σετ δεδομένων εκπαίδευσης και μετά το βάλουμε να δώσει αποτελέσματα για δεδομένα εισόδου τα οποία δεν ανήκουν στο σετ εκπαίδευσης, τότε η ορθότητα των αποτελεσμάτων αποτελεί και κριτήριο για το πόσο καλά μπορεί το δίκτυο να γενικεύει. Ουσιαστικά η εκπαίδευση του δικτύου με τον Back Propagation προκαλεί τον καθορισμό των βαρών των συνάψεών του και επομένως την μη γραμμική αντιστοίχιση εισόδου – εξόδου. Άρα το δίκτυο μαθαίνει μια μη γραμμική συνάρτηση με την οποία προσπαθεί να προσομοιάσει όσο το δυνατόν πιστότερα την αντιστοίχιση αυτή. Στο σχήμα 5 παρακάτω φαίνονται δύο τέτοιες μη γραμμικές αντιστοιχίσεις που έχει μάθει να κάνει ένα ANN. Η πρώτη αποτελεί παράδειγμα καλής γενίκευσης ενώ η δεύτερη κακής.



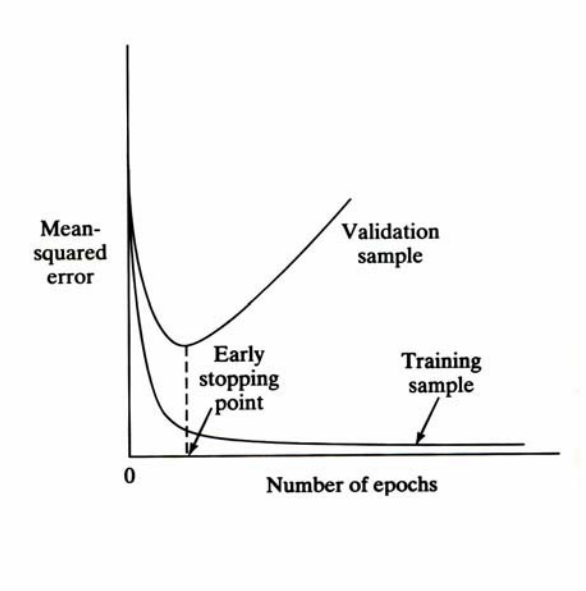

**Σχήμα 5.** (a) Αποτέλεσμα καλής γενίκευσης (b) Αποτέλεσμα κακής γενίκευσης

Στην πρώτη περίπτωση το δίκτυο έχει εκπαιδευτεί με τέτοιο τρόπο έτσι ώστε η μη γραμμική συνάρτηση που έχει μάθει να αποτελεί μια επιτυχημένη μη γραμμική παρεμβολή των δεδομένων εξόδου. Στη δεύτερη όμως περίπτωση βλέπουμε ότι η αντίστοιχη μη γραμμική παρεμβολή δεν είναι επιτυχημένη. Έτσι σε δεδομένα που το δίκτυο δεν έχει ξαναδεί, υπάρχει η πιθανότητα μεγάλου σφάλματος. Το αποτέλεσμα αυτό οφείλεται κυρίως στην υπερεκπαίδευση. Όταν το δίκτυο αφήνεται πολλές εποχές να εκπαιδεύεται τότε στην προσπάθειά του να μειώσει όλο και περισσότερο το λάθος $E_{av}$, αναγκάζεται να

προσαρμόσει έτσι τις συνάψεις του προκαλώντας μια έντονη μη γραμμικότητα η οποία επαληθεύει μεν το σετ εκπαίδευσης αλλά όχι αναγκαστικά και τη μη γραμμική σχέση εισόδου – εξόδου που εμείς θέλουμε να μάθει. Επίσης μπορεί να οφείλεται και σε τυχόν θόρυβο που έχουν τα δεδομένα εξόδου. Εμείς δε θα θέλαμε σε καμία περίπτωση να μάθει το δίκτυό μας το θόρυβο αυτό. Σε αυτό το πολύ σημαντικό πρόβλημα, δηλαδή το πότε θα πρέπει να σταματάει η εκπαίδευση του δικτύου χωρίς να προκαλείται υπερεκπαίδευση αλλά μια ικανοποιητική παρεμβολή στα δεδομένα εξόδου δίνει λύση η μέθοδος Cross Validation η οποία εφαρμόστηκε με επιτυχία στο πληροφοριακό σύστημα και εξετάζεται στην επόμενη ενότητα.

## 3.4 Η μέθοδος Cross Validation

Η μέθοδος αυτή αποτελεί το κριτήριο τερματισμού του αλγορίθμου Back Propagation και δίνει λύση στο πρόβλημα της υπερεκπαίδευσης που εξετάσαμε στην προηγούμενη ενότητα. Η φιλοσοφία της μεθόδου είναι εξαιρετικά απλή. Απλά το σύνολο δεδομένων εκπαίδευσης χωρίζεται σε δύο υποσύνολα, ένα σύνολο εκτίμησης και ένα σύνολο ελέγχου. Το πρώτο σύνολο χρησιμοποιείται για να εκπαιδεύσει το δίκτυο και το δεύτερο για την επικύρωσή του, τον έλεγχο δηλαδή της ικανότητας γενίκευσης του δικτύου σε δεδομένα που δεν έχει ξαναδεί. Στο σχήμα 6 φαίνεται αυτή ακριβώς η διαδικασία. Το ANN εκπαιδεύεται με το σετ επικύρωσης και ταυτόχρονα ελέγχεται με το σετ ελέγχου ανά μια περίοδο εποχών. Τη στιγμή που το σφάλμα απόκρισης στο σετ ελέγχου αρχίζει να αυξάνεται τότε σταματάει πρόωρα και η εκπαίδευση. Τη στιγμή αυτή είναι και που το δίκτυο έχει μάθει μια μη γραμμική συνάρτηση η οποία μπορεί και παρεμβάλει ικανοποιητικά στα δεδομένα εξόδου χωρίς να έχει υπερεκπαιδευτεί ή να έχει μάθει τυχόν θόρυβο. Από εκεί και μετά βλέπουμε πως στην προσπάθειά του το δίκτυο να μειώσει το σφάλμα $E_{av}$ μπορεί να προσομοιάσει μια έντονα μη γραμμική συνάρτηση η οποία επαληθεύει μεν το σετ εκπαίδευσης, δε μπορεί όμως να γενικεύσει σε άγνωστα δεδομένα.




**Σχήμα 6.** Παρουσίαση της μεθόδου Cross Validation και του σημείου πρόωρου τερματισμού.

Σημασία έχει επίσης το σημείο στο οποίο πρέπει να χωρίσουμε το αρχικό σύνολο εκπαίδευσης σε δύο υποσύνολα. Αυτό εξαρτάται από το συνολικό αριθμό των συνάψεων του δικτύου και το συνολικό αριθμό των δειγμάτων του αρχικού συνόλου. Έτσι αν N ο συνολικός αριθμός δειγμάτων στο αρχικό σύνολο εκπαίδευση και W ο συνολικός αριθμός συνάψεων στο ANN τότε:

- Αν N > 30W (Asymptotic mode) τότε έχουμε πάρα πολλά δεδομένα εκπαίδευσης. Σε αυτή την περίπτωση δεν υπάρχει λόγος πρόωρου τερματισμού. Εδώ το κριτήριο τερματισμού για τον Back Propagation είναι η μεταβολή του σφάλματος $E_{av}$ να γίνει μικρότερη του 0.01% από εποχή σε εποχή.

- Αν N < 30W (Nonasymptotic mode) τότε έχουμε $r_{opt} = 1 - \frac{\sqrt{2 \cdot w - 1} - 1}{2 \cdot (w - 1)}$. Αν για παράδειγμα W=100, τότε $r_{opt} = 0.07$, δηλαδή 93% του αρχικού συνόλου θα γίνει το σετ εκτίμησης και 7% το σετ ελέγχου.

- Αν N < 30W και N αρκετά μικρό (έχουμε δηλαδή πολύ λίγα αρχικά δεδομένα) τότε χρησιμοποιούμε μια παραλλαγή της μεθόδου Cross Validation, τη μέθοδο Leave One Out. Σύμφωνα με αυτή N-1 παραδείγματα χρησιμοποιούνται για την εκπαίδευση του δικτύου και 1 για τον έλεγχό του. Αυτό επαναλαμβάνεται N φορές επιλέγοντας κυκλικά κάθε φορά ένα διαφορετικό παράδειγμα ελέγχου. Το σφάλμα εδώ $E_{av}$ υπολογίζεται πάνω στα N διαφορετικά παραδείγματα ελέγχου και η εποχή ορίζεται ως το τέλος μιας επανάληψης ελέγχου και με τα N παραδείγματα.

## 3.5 Ο αλγόριθμος Optimal Brain Surgeon

### 3.5.1 Γενικά

Το πρόβλημα που προσπαθούμε να αντιμετωπίσουμε με τον συγκεκριμένο αλγόριθμο είναι η μείωση της πολυπλοκότητας ενός νευρωνικού δικτύου περικόπτοντας 'άχρηστες' συνάψεις χωρίς όμως να προκαλέσουμε μεγάλη αύξηση στο σφάλμα εξόδου $E_{av}$. Αυτός εφαρμόζεται κυρίως σε μεγάλα δίκτυα όπου και έχει θεαματικά αποτελέσματα. Τέτοιο παράδειγμα αποτελεί το NETalk ANN το οποίο έχει ένα κρυμμένο επίπεδο και 18.000 συνάψεις. Η μείωση του δικτύου που προκύπτει με τον OBS χάρη στους Sejnowski και Rosenberg (1987) είναι μόλις στις 1560 συνάψεις. Απαραίτητος περιορισμός για να εφαρμοστεί ο συγκεκριμένος αλγόριθμος είναι το νευρωνικό δίκτυο να έχει εκπαιδευτεί πρώτα σε κάποιο τοπικό ελάχιστο. Στην επόμενη ενότητα παραθέτουμε συνοπτικά τα βήματα του αλγορίθμου. Στο Κεφάλαιο 5 βρίσκεται και ο πηγαίος κώδικας του OBS μαζί με τον υπόλοιπο κώδικα του πληροφοριακού συστήματος.

### 3.5.2 Ο Optimal Brain Surgeon συνοπτικά

**1). Εκπαίδευση** του δικτύου σε ένα τοπικό ελάχιστο της επιφάνειας λάθους $E_{av}$.

**2). Υπολογισμός του διανύσματος** $\xi(n) = \frac{1}{\sqrt{N}} \cdot \frac{\partial F(w, x(n))}{\partial w}$ όπου N ο συνολικός αριθμός των παραδειγμάτων εκπαίδευσης, F η συνάρτηση απεικόνισης εισόδου στην έξοδο και w, το w-by-1 διάνυσμα με w να συμβολίζει το συνολικό αριθμό συνάψεων. Η F για ένα ANN τριών κρυμμένων επιπέδων και μιας εξόδου είναι:

$$F(w, x(n)) = \phi(\sum_{k=0}^{m_3} w_{1k} \cdot \phi(\sum_{l=0}^{m_2} w_{kl} \cdot \phi(\sum_{j=0}^{m_1} w_{lj} \cdot \phi(\sum_{i=0}^{m_0} w_{ji} \cdot x_i))))$$

- Για ένα νευρώνα του πρώτου κρυμμένου επιπέδου έχουμε:

$$\frac{\partial F(w,x(n))}{\partial w_{ji}^{(1)}} = \phi^{'}(u_1^{(4)}) \cdot \phi^{'}(u_j^{(1)}) \cdot x_i \cdot \sum_{k=0}^{m_3}(w_{1k} \cdot \phi'(u_k^{(3)}) \cdot \sum_{l=0}^{m_2} w_{kl} \cdot \phi'(u_l^{(2)}) \cdot w_{lj})$$

- Για ένα νευρώνα κάποιου άλλου κρυμμένου επιπέδου έχουμε: (πχ. για το δεύτερο κρυμμένο επίπεδο)

$$\frac{\partial F(w,x(n))}{\partial w_{lj}^{(2)}} = \phi^{'}(u_1^{(4)}) \cdot \phi^{'}(u_l^{(2)}) \cdot \phi(u_j^{(1)}) \cdot \sum_{k=0}^{m_3}(w_{1k} \cdot \phi'(u_k^{(3)}) \cdot w_{kl})$$

- Για το νευρώνα του επιπέδου εξόδου έχουμε:

$$\frac{\partial F(w,x(n))}{\partial w_{1k}^{(4)}} = \phi^{'}(u_1^{(4)}) \cdot \phi^{'}(u_k^{(3)})$$

Με παρόμοιο τρόπο υπολογίζονται και οι μερικές παράγωγοι για ένα ΑΝΝ Ν κρυμμένων επιπέδων.

**3). Υπολογισμός του αντίστροφου πίνακα** $H^{-1}$ από τον τύπο:

$$H^{-1}(n) = H^{-1}(n-1) - \frac{H^{-1}(n-1) \cdot \xi(n) \cdot \xi^T(n) \cdot H^{-1}(n-1)}{1 + \xi^T(n) \cdot H^{-1}(n-1) \cdot \xi(n)}$$

με $H^{-1}(0) = \delta^{-1} \cdot I$ όπου δ είναι ένας πολύ μικρός θετικός ακέραιος.

4**). Εύρεση του i στο οποίο αντιστοιχεί στο μικρότερη ποσότητα:**

$S_i = \frac{w_i^2}{2 \cdot [H^{-1}]_{i,i}}$, όπου $[H^{-1}]_{i,i}$ είναι το στοιχείο στη θέση i,i του πίνακα Η.

Αν $S_i << E_{av}$ τότε διέγραψε τη σύναψη i και επανέλαβε το βήμα 4 αλλιώς πήγαινε στο βήμα 5.

**5). Ενημέρωσε όλα τα βάρη** των συνάψεων του δικτύου σύμφωνα με τις ποσότητες:

$\Delta w = -\frac{w_i}{[H^{-1}]_{i,i}} \cdot H^{-1} \cdot 1_i$, όπου $1_i$ είναι το μοναδιαίο διάνυσμα του οποίου τα στοιχεία είναι όλα 0 εκτός από το στοιχείο i. Πήγαινε στο βήμα 2.

**6). Σταμάτησε** εάν κανένα $S_i << E_{av}$ δεν μπορεί να βρεθεί.

**7). Επανεκπαίδευσε** το δίκτυο.

Κεφάλαιο

4

# Το Πληροφοριακό Σύστημα

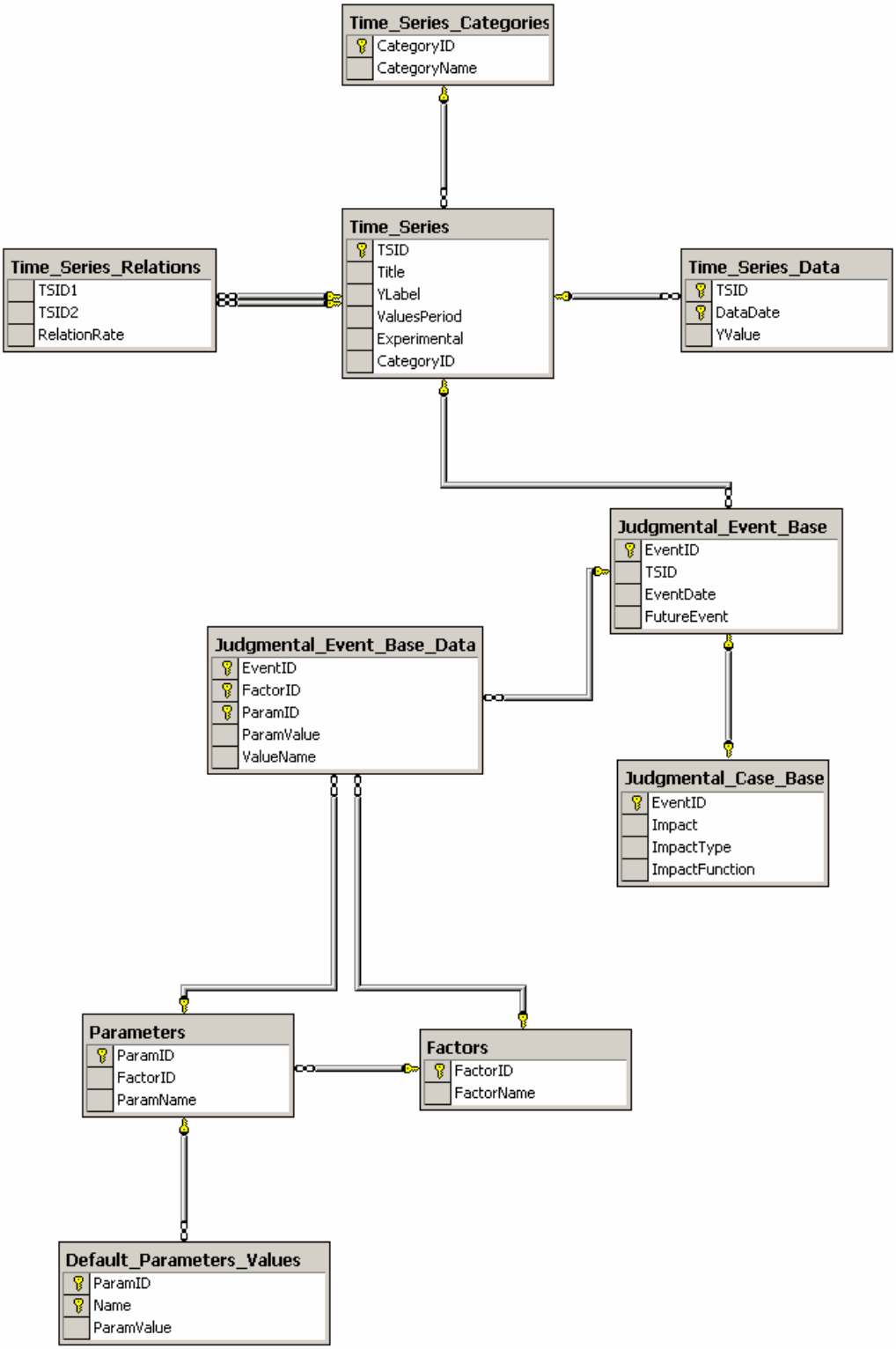
## 4.1 Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιάσουμε την εφαρμογή που κατασκευάστηκε στα πλαίσια της συγκεκριμένης πτυχιακής εργασίας εξηγώντας αναλυτικά όλες τις δυνατότητες που παρέχει και τις λειτουργίες που επιτελεί. Για το λόγο αυτό έχουμε εισάγει και πολλά screenshots τα οποία συνοδεύουν το κείμενο. Ουσιαστικά το τέταρτο κεφάλαιο αποτελεί το manual της εφαρμογής η οποία υλοποιεί εκτός άλλων, όλη τη θεωρία των δύο προηγούμενων κεφαλαίων.

Η γλώσσα που χρησιμοποιήθηκε για την κατασκευή της εφαρμογής είναι η C++ και ο compiler ο C++ Builder 5 της Borland. Η βάση δεδομένων που συνοδεύει το πληροφοριακό σύστημα φτιάχτηκε με τον SQL Server 2000 της Microsoft.

## 4.2 Η Βάση Δεδομένων

### 4.2.1 Το σχήμα της βάσης δεδομένων.


Time_Series_Categories
CategoryID
CategoryName
Time_Series
TSID
Title
YLabel
ValuesPeriod
Experimental
CategoryID
Time_Series_Relations
TSID1
TSID2
RelationRate
Time_Series_Data
TSID
DataDate
YValue
Judgmental_Event_Base
EventID
TSID
EventDate
FutureEvent
Judgmental_Event_Base_Data
EventID
FactorID
ParamID
ParamValue
ValueName
Judgmental_Case_Base
EventID
Impact
ImpactType
ImpactFunction
Parameters
ParamID
FactorID
ParamName
Factors
FactorID
FactorName
Default_Parameters_Values
ParamID
Name
ParamValue

Σε απόλυτη συμφωνία με το κεφάλαιο 2 και τους ορισμούς που δόθηκαν, οι πίνακες της Βάσης Δεδομένων υλοποιούνε τα παρακάτω:

- Ο πίνακας Time_Series περιέχει τα στοιχεία των χρονοσειρών που είναι αποθηκευμένες στη ΒΔ. Περιέχει στοιχεία όπως τον τίτλο της χρονοσειράς, την περιοδικότητα των δεδομένων της, αν είναι πειραματική η όχι, και την κατηγορία στην οποία ανήκει.
- Ο πίνακας Time_Series_Data περιέχει τα δεδομένα όλων των χρονοσειρών της ΒΔ. Δηλαδή ημερομηνίες και αντίστοιχες τιμές.
- Ο πίνακας Time_Series_Categories περιέχει διάφορες κατηγορίες στις οποίες μπορούμε να κατατάξουμε τις χρονοσειρές για την ευκολότερη διαχείρισή τους μέσα στη ΒΔ.
- Ο πίνακας Time_Series_Relations περιέχει τις συσχετίσεις των χρονοσειρών που ορίζει ο χρήστης του πληροφοριακού συστήματος. Έτσι μια χρονοσειρά μπορεί να συσχετίζεται με μια άλλη ως προς το βαθμό ομοιότητας και συμπεριφοράς απέναντι σε όμοια υποκειμενικά γεγονότα, με κάποιο βαθμό.
- Ο πίνακας Judgmental_Event_Base είναι ο πίνακας υποκειμενικών γεγονότων. Περιέχει τα ιστορικά γεγονότα που εμφανίζονται σε κάθε χρονοσειρά αλλά και τα μελλοντικά τα οποία ορίζει ο χρήστης προκειμένου να κάνει κάποια πρόβλεψη. Ο πίνακας αυτός περιέχει μόνο την ημερομηνία εμφάνισης των γεγονότων και το κλειδί EventID που είναι χαρακτηριστικό για κάθε γεγονός.
- Ο πίνακας Judgmental_Case_Base είναι ο πίνακας υποκειμενικών σεναρίων. Ουσιαστικά αντιστοιχεί κάποιο υποκειμενικό γεγονός με την επίδρασή του στη χρονοσειρά. Ο συνδυασμός αυτών των δύο σύμφωνα και με το κεφάλαιο 2 ονομάζεται υποκειμενικό σενάριο. Ο πίνακας αυτός θα μπορούσε και να συγχωνευτεί με τον προηγούμενο (Judgmental_Event_Base) χωρίς τεχνικά να περιοριστούν οι λειτουργικές δυνατότητες της ΒΔ.
- Ο πίνακας Judgmental_Event_Base_Data περιέχει τους παράγοντες με τις παραμέτρους τους για κάθε υποκειμενικό γεγονός που εμφανίζεται σε κάθε χρονοσειρά στη ΒΔ. Είναι ίσως λειτουργικά ο σημαντικότερος πίνακας γιατί πάνω στα δεδομένα του στηρίζεται όλη η φιλοσοφία λειτουργίας της εφαρμογής. Κάθε πλειάδα αυτού του πίνακα, όπως φαίνεται και από το σχήμα της ΒΔ, αποτελεί μια παράμετρο ενός παράγοντα ενός υποκειμενικού γεγονότος μιας χρονοσειράς. Η πλειάδα αυτή περιέχει και την τιμή της παραμέτρου. Το πληροφοριακό σύστημα με δεδομένα που παίρνει από τους τρεις τελευταίους πίνακες εφαρμόζει τρεις βασικούς αλγορίθμους πρόβλεψης που θα δούμε παρακάτω.
- Ο πίνακας Factors περιέχει τους ορισμούς των παραγόντων που έχει δώσει ο χρήστης μέσα από την εφαρμογή.

- Ο πίνακας Parameters περιέχει τους ορισμούς των παραμέτρων για κάθε παράγοντα που έχει δώσει ο χρήστης μέσα από την εφαρμογή.
- Τέλος, ο πίνακας Default_Parameters_Values περιέχει κάποιες δυνατές τιμές παραμέτρων που αν θέλει μπορεί να ορίσει ο χρήστης. Τις τιμές αυτές μπορεί να τις αναθέτει στις αντίστοιχες παραμέτρους που εμφανίζονται σε κάποιο υποκειμενικό γεγονός σε κάποια χρονοσειρά.

Παρακάτω παραθέτουμε και τον SQL κώδικα που δημιουργεί την παραπάνω βάση δεδομένων.

### 4.2.2 SQL script

```
CREATE DATABASE JudgmentalDB
GO

use JudgmentalDB
GO

CREATE TABLE Default_Parameters_Values
	(
		ParamID		int		NOT NULL ,
		Name		varchar(250)	NOT NULL ,
		ParamValue	float		NOT NULL
	)
GO

CREATE TABLE Factors
	(
		FactorID	int		IDENTITY (1, 1)	NOT NULL ,
		FactorName	varchar(250)	NOT NULL
	)
GO

CREATE TABLE Judgmental_Case_Base
	(
		EventID		int		NOT NULL ,
		Impact		float		NOT NULL ,
		ImpactType	int		NOT NULL ,
		ImpactFunction	varchar(250)	NOT NULL
	)
GO
```

```
CREATE TABLE Judgmental_Event_Base
	(
		EventID		int		IDENTITY (1, 1)	NOT NULL ,
		TSID		int		NOT NULL ,
		EventDate	datetime	NOT NULL ,
		FutureEvent	bit		NOT NULL
	)
GO

CREATE TABLE Judgmental_Event_Base_Data
	(
		EventID		int		NOT NULL ,
		FactorID	int		NOT NULL ,
		ParamID		int		NOT NULL ,
		ParamValue	float		NOT NULL ,
		ValueName	varchar(250)	NULL
	)
GO

CREATE TABLE Parameters
	(
		ParamID		int		IDENTITY (1, 1)	NOT NULL ,
		FactorID	int		NOT NULL ,
		ParamName	varchar(250)	NOT NULL
	)
GO

CREATE TABLE Time_Series
	(
		TSID		int		IDENTITY (1, 1)	NOT NULL ,
		Title		varchar(250)	NOT NULL ,
		YLabel		varchar(250)	NULL ,
		ValuesPeriod	int		NULL ,
		Experimental	bit		NULL ,
		CategoryID	int		NULL
	)
GO

CREATE TABLE Time_Series_Categories
	(
		CategoryID	int		IDENTITY (1, 1)	NOT NULL ,
		CategoryName	varchar(50)	NOT NULL
	)
GO
```

```
CREATE TABLE Time_Series_Data
	(
		TSID		int		NOT NULL ,
		DataDate	datetime	NOT NULL ,
		YValue		float		NOT NULL
	)
GO

CREATE TABLE Time_Series_Relations
	(
		TSID1		int		NOT NULL ,
		TSID2		int		NOT NULL ,
		RelationRate	int		NOT NULL
	)
GO

ALTER TABLE Default_Parameters_Values
	WITH NOCHECK
	ADD CONSTRAINT PK_Default_Parameters_Values PRIMARY KEY CLUSTERED
	(
		ParamID,
		Name
	)
GO

ALTER TABLE Factors
	WITH NOCHECK
	ADD CONSTRAINT PK_Factors PRIMARY KEY CLUSTERED
	(
		FactorID
	)
GO

ALTER TABLE Judgmental_Case_Base
	WITH NOCHECK
	ADD CONSTRAINT PK_Judgmental_Case_Base PRIMARY KEY CLUSTERED
	(
		EventID
	)
GO
```

```
ALTER TABLE Judgmental_Event_Base
	WITH NOCHECK
	ADD CONSTRAINT PK_Judgmental_Event_Base PRIMARY KEY CLUSTERED
	(
		EventID
	)
GO

ALTER TABLE Judgmental_Event_Base_Data
	WITH NOCHECK
	ADD CONSTRAINT PK_Judgmental_Event_Base_Data PRIMARY KEY CLUSTERED
	(
		EventID,
		FactorID,
		ParamID
	)
GO

ALTER TABLE Parameters
	WITH NOCHECK
	ADD CONSTRAINT PK_Parameters PRIMARY KEY CLUSTERED
	(
		ParamID
	)
GO

ALTER TABLE Time_Series
	WITH NOCHECK
	ADD CONSTRAINT PK_Time_Series PRIMARY KEY CLUSTERED
	(
		TSID
	)
GO

ALTER TABLE Time_Series_Categories
	WITH NOCHECK
	ADD CONSTRAINT PK_Time_Series_Categories PRIMARY KEY CLUSTERED
	(
		CategoryID
	)
GO
```

```
ALTER TABLE Time_Series_Data
	WITH NOCHECK
	ADD CONSTRAINT PK_Time_Series_Data PRIMARY KEY CLUSTERED
	(
		TSID,
		DataDate
	)
GO

ALTER TABLE Time_Series
	WITH NOCHECK
	ADD CONSTRAINT IX_Time_Series UNIQUE NONCLUSTERED
	(
		Title
	)
GO

ALTER TABLE Default_Parameters_Values
	ADD CONSTRAINT FK_Default_Parameters_Values_Parameters FOREIGN KEY
	(
		ParamID
	)
	REFERENCES Parameters
	(
		ParamID
	)
	ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE Judgmental_Case_Base
	ADD CONSTRAINT FK_Judgmental_Case_Base_Judgmental_Event_Base FOREIGN KEY
	(
		EventID
	)
	REFERENCES Judgmental_Event_Base
	(
		EventID
	)
	ON DELETE CASCADE  ON UPDATE CASCADE
GO
```

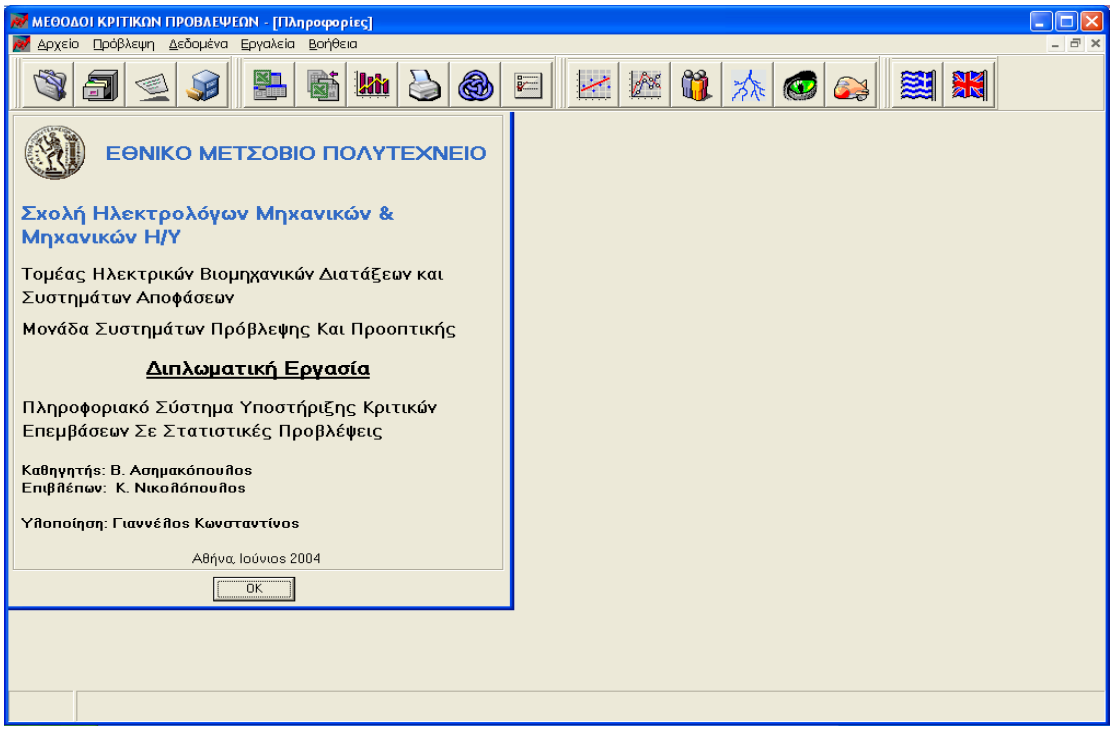
```
ALTER TABLE Judgmental_Event_Base
	ADD CONSTRAINT FK_Judgmental_Event_Base_Time_Series FOREIGN KEY
	(
		TSID
	)
	REFERENCES Time_Series
	(
		TSID
	)
	ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE Judgmental_Event_Base_Data
	ADD CONSTRAINT FK_Judgmental_Event_Base_Data_Factors FOREIGN KEY
	(
		FactorID
	)
	REFERENCES Factors
	(
		FactorID
	),
	CONSTRAINT FK_Judgmental_Event_Base_Data_Judgmental_Event_Base FOREIGN KEY
	(
		EventID
	)
	REFERENCES Judgmental_Event_Base
	(
		EventID
	)
	ON DELETE CASCADE  ON UPDATE CASCADE,
	CONSTRAINT FK_Judgmental_Event_Base_Data_Parameters FOREIGN KEY
	(
		ParamID
	)
	REFERENCES Parameters
	(
		ParamID
	)
	ON DELETE CASCADE  ON UPDATE CASCADE
GO
```

```
ALTER TABLE Parameters
	ADD CONSTRAINT FK_Parameters_Factors FOREIGN KEY
	(
		FactorID
	)
	REFERENCES Factors
	(
		FactorID
	)
	ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE Time_Series
	ADD CONSTRAINT FK_Time_Series_Time_Series_Categories FOREIGN KEY
	(
		CategoryID
	)
	REFERENCES Time_Series_Categories
	(
		CategoryID
	)
	ON UPDATE CASCADE
GO

ALTER TABLE Time_Series_Data
	ADD CONSTRAINT FK_Time_Series_Data_Time_Series FOREIGN KEY
	(
		TSID
	)
	REFERENCES Time_Series
	(
		TSID
	)
	ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE Time_Series_Relations
	ADD CONSTRAINT FK_Time_Series_Relations_Time_Series FOREIGN KEY
	(
		TSID1
	)
	REFERENCES Time_Series
	(
		TSID
	),
```

```
        CONSTRAINT FK_Time_Series_Relations_Time_Series1 FOREIGN KEY
        (
                TSID2
        )
        REFERENCES Time_Series
        (
                TSID
        )
GO
```

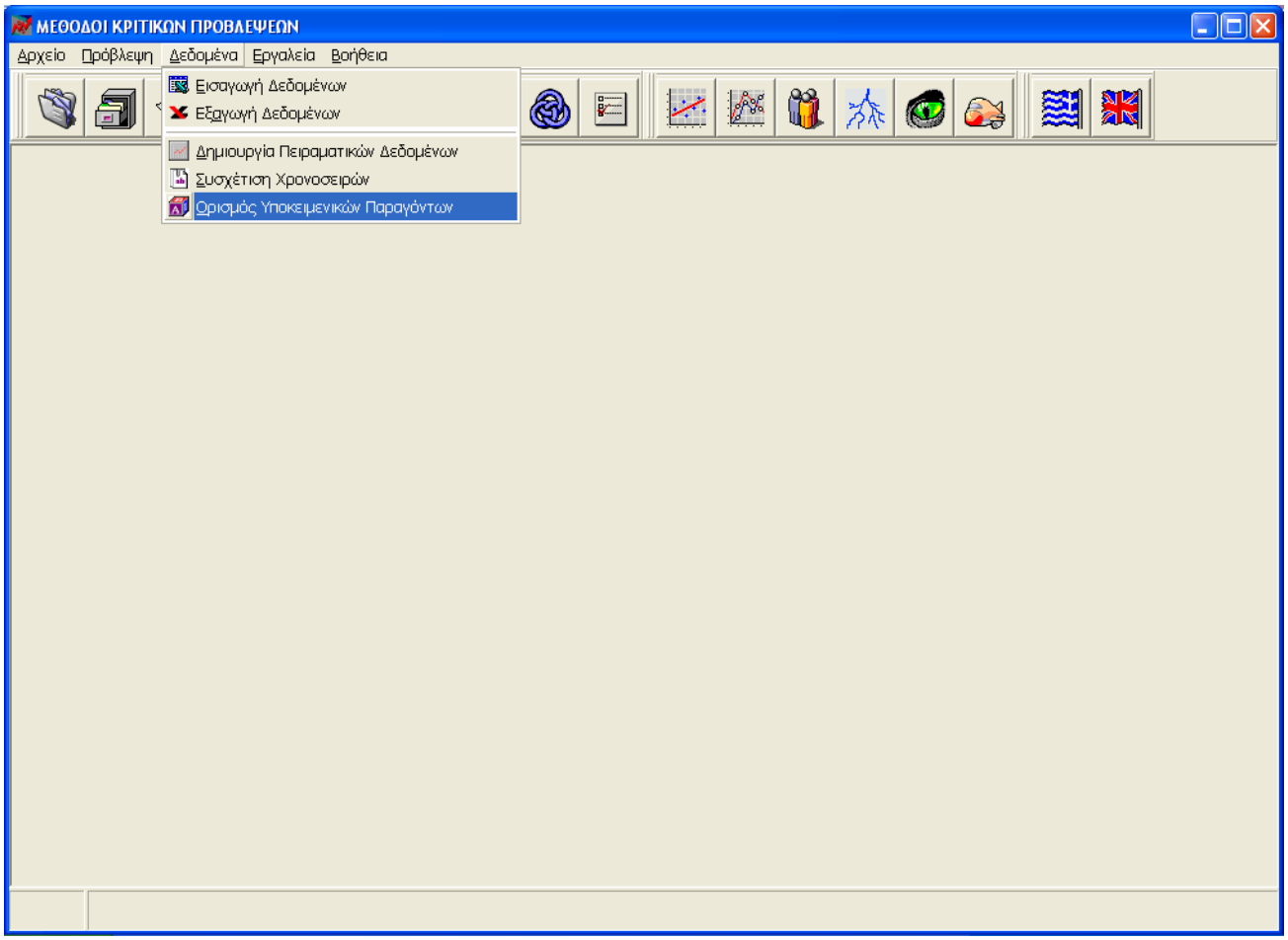
## 4.3 Το πληροφοριακό σύστημα

Η συγκεκριμένη ενότητα αποτελεί το manual της εφαρμογής, η κατασκευή της οποίας είχε και το μεγαλύτερο βάρος σε αυτήν την εργασία. Παρακάτω θα δούμε αρκετά screenshots με τη βοήθεια των οποίων θα εξηγήσουμε αναλυτικά τις δυνατότητες και τις λειτουργίες που μπορεί ο χρήστης να επιτελέσει. Στην επόμενη εικόνα βλέπουμε το πρώτο στιγμιότυπο τρεξίματος που εμφανίζεται στην οθόνη και αποτελείται από μία φόρμα καλωσορίσματος.

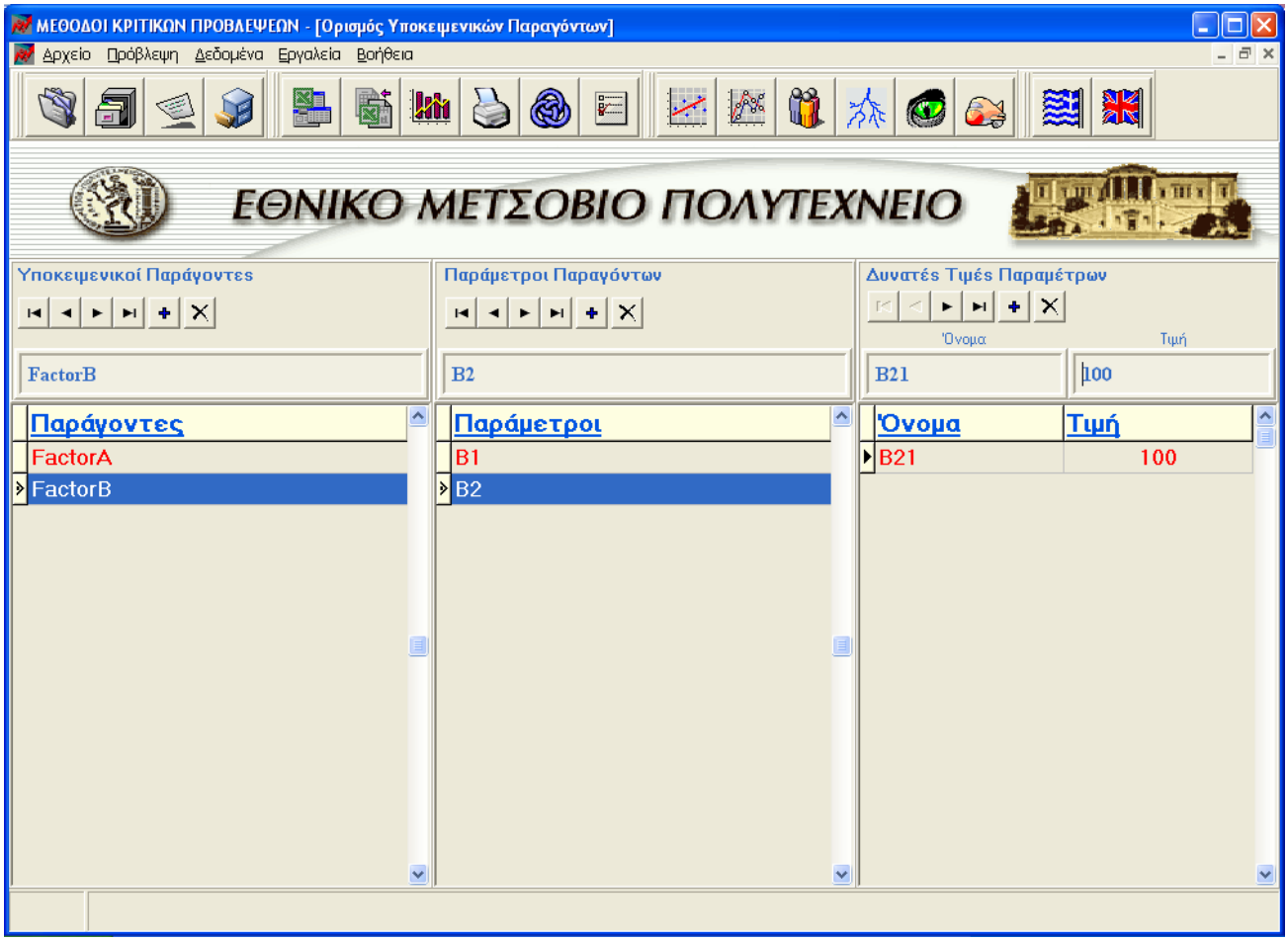


### 4.3.1 Ορισμός Υποκειμενικών Παραγόντων και των Παραμέτρων τους

Μια από τις δυνατότητες που δίνει η εφαρμογή στο χρήστη είναι ο ορισμός των Υποκειμενικών Παραγόντων και των Παραμέτρων τους. Αυτό γίνεται μέσα από τη φόρμα που εμφανίζεται είτε πατώντας το κουμπί  είτε μέσα από το βασικό menu όπως φαίνεται παρακάτω.



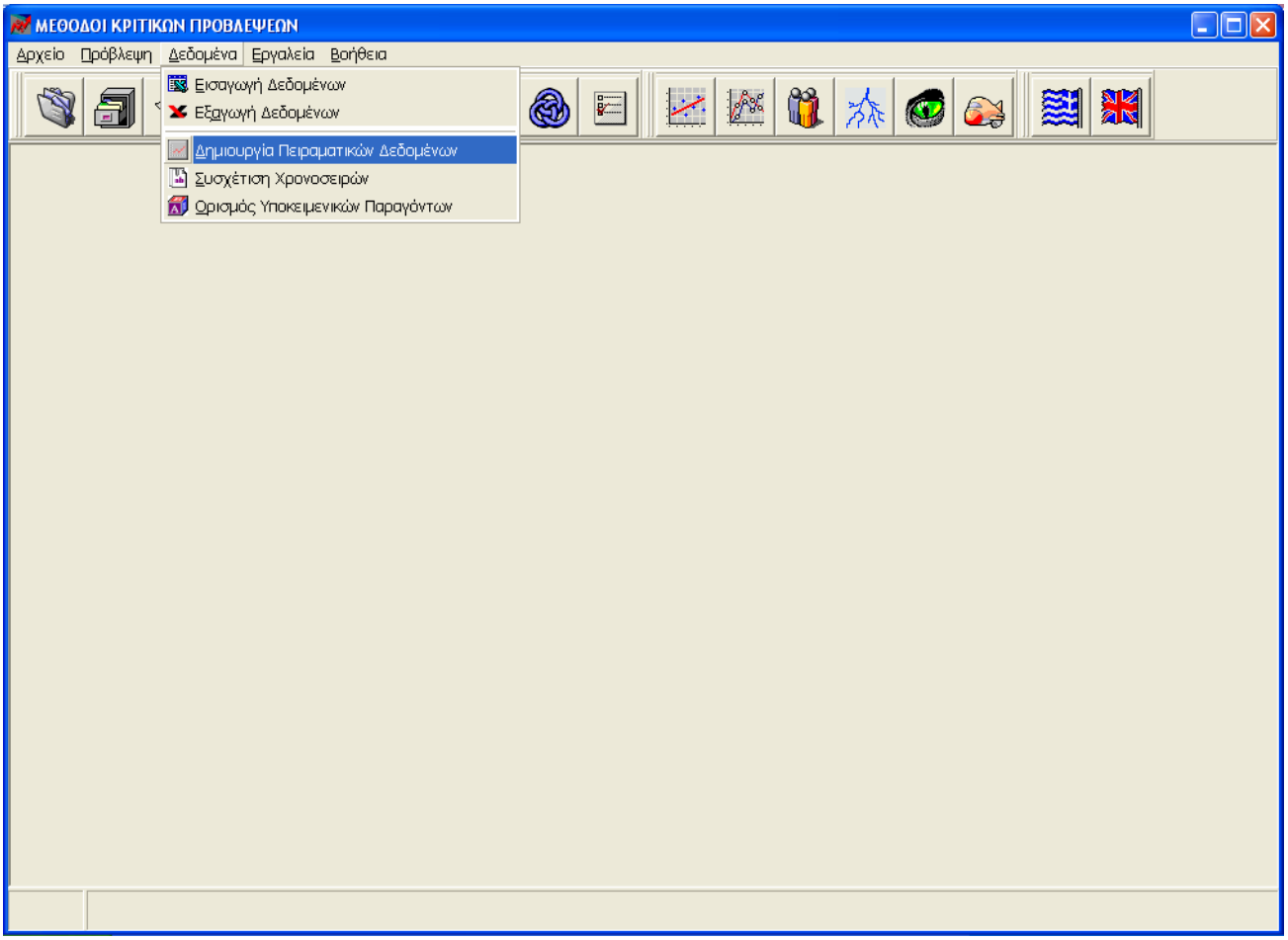

Στην επόμενη φόρμα η οποία φαίνεται στην επόμενη σελίδα μπορούμε να εισάγουμε στη βάση Υποκειμενικών Παραγόντων ένα Παράγοντα, μία Παράμετρο ή μία δυνατή τιμή μιας Παραμέτρου απλά συμπληρώνοντας τα κατάλληλα στοιχεία στα αντίστοιχα κουτιά και χρησιμοποιώντας τα κουμπιά πλοήγησης.



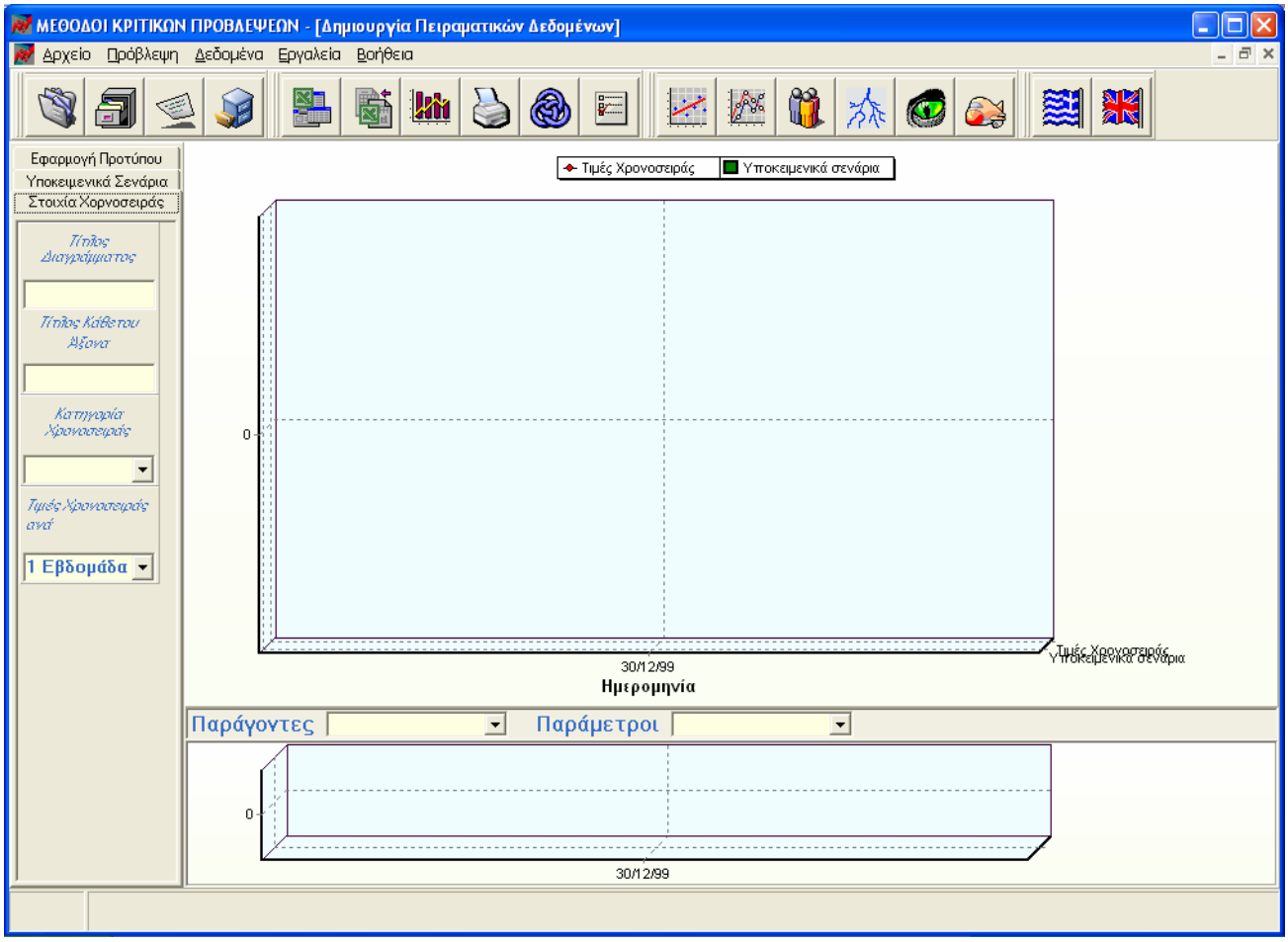

Εδώ βλέπουμε ότι έχουμε εισάγει δύο Παράγοντες τους FactorA και FactorB. Για τον FactorB βλέπουμε ότι έχουμε εισάγει δύο παραμέτρους, τις B1 και B2. Για την παράμετρο B2 έχουμε εισάγει μία δυνατή τιμή, την B21. Για τον Παράγοντα FactorA έχουμε εισάγει μία μόνο παράμετρο την A1. Τα κουμπιά πλοήγησης μας δίνουν τη δυνατότητα και να διαγράψουμε τα επιλεγμένα στοιχεία. Η πολλαπλή επιλογή στοιχείων γίνεται κρατώντας πατημένο το ctrl και επιλέγοντας με το mouse.

### 4.3.2 Δημιουργία Πειραματικών Δεδομένων

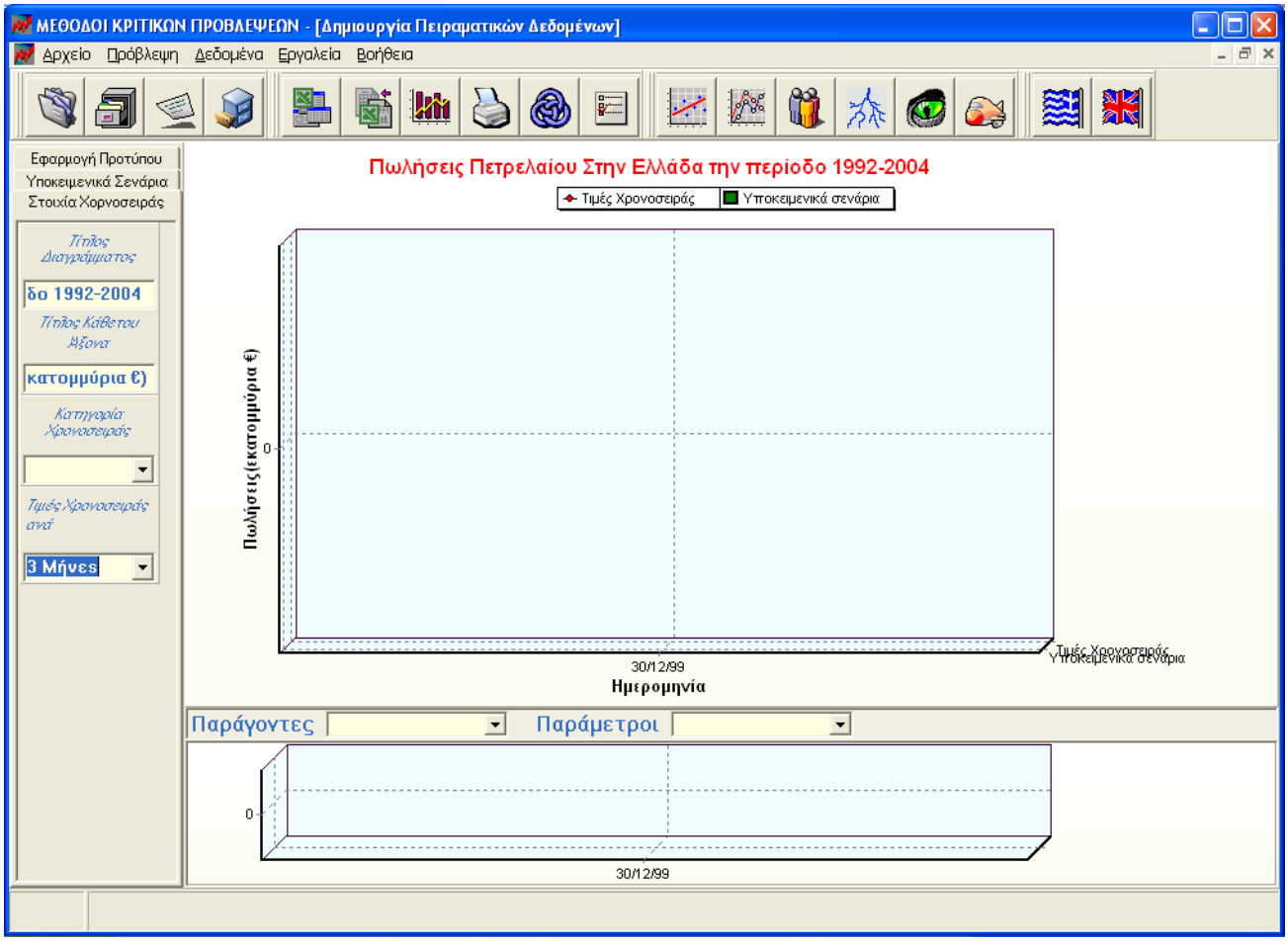
Μία από τις κυριότερες λειτουργίες που μπορεί ο χρήστης να επιτελέσει είναι η δημιουργία πειραματικών δεδομένων. Βασικός στόχος είναι η προσομοίωση πραγματικών χρονοσειρών και υποκειμενικών σεναρίων με σκοπό την επαλήθευση και σύγκριση αποτελεσμάτων των διαφόρων μεθόδων πρόβλεψης. Η δημιουργία μιας πειραματικής χρονοσειράς μπορεί να γίνει είτε από το κυρίως menu είτε πατώντας το κουμπί με το εικονίδιο: 




Η επόμενη φόρμα που εμφανίζεται είναι αυτή των πειραματικών χρονοσειρών και φαίνεται στην επόμενη σελίδα.

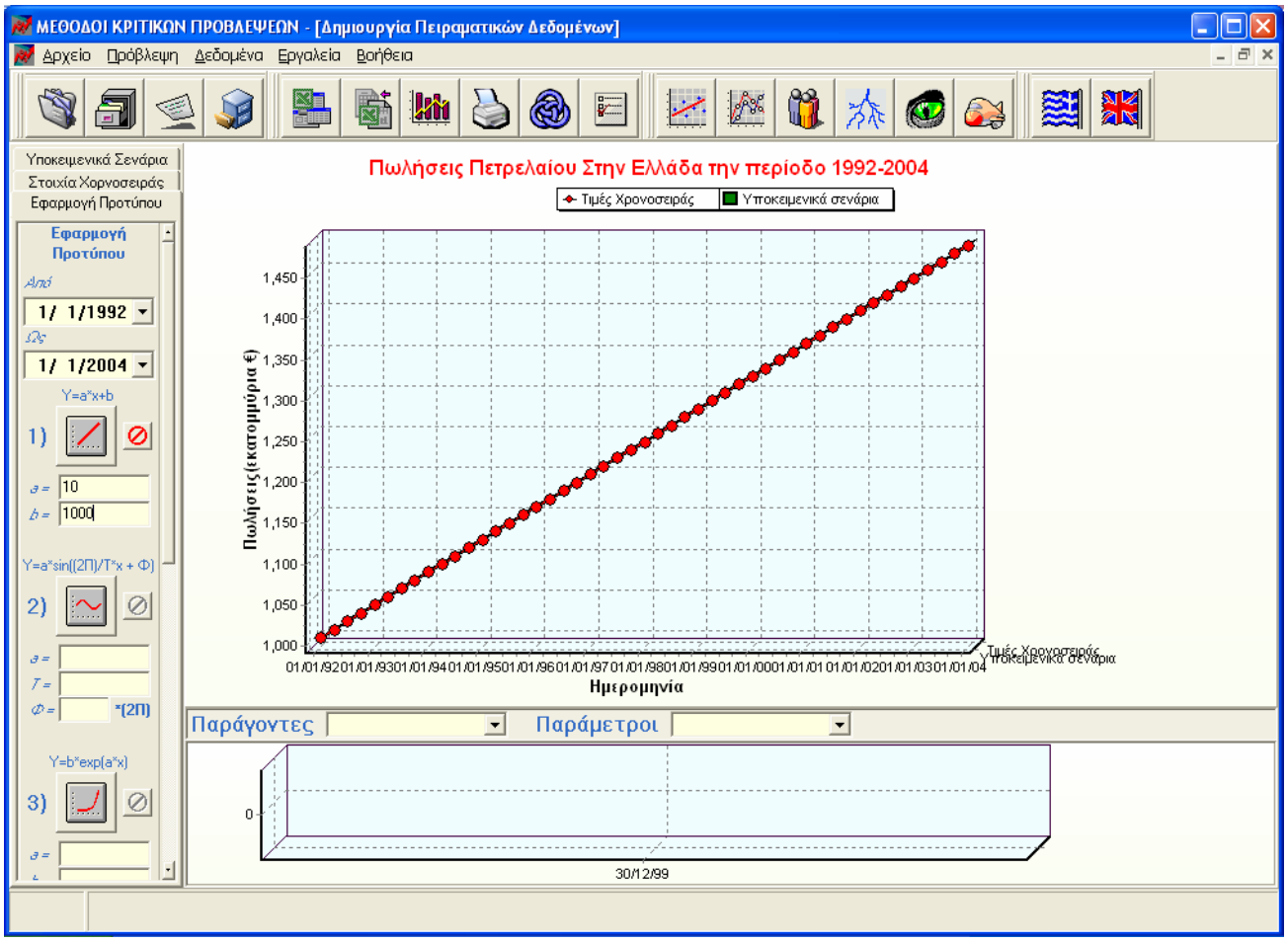



Στα αριστερά της φόρμας βλέπουμε τα τρία βασικά tab της φόρμας με τη βοήθεια των οποίων επιτελούνται όλες οι λειτουργίες. Το αρχικό tab που εμφανίζεται είναι αυτό με τίτλο 'Στοιχεία Χρονοσειράς'. Εδώ μπορούμε να πληκτρολογήσουμε τα βασικά στοιχεία μιας χρονοσειράς όπως τον τίτλο της, τον τίτλο του μεγέθους που απεικονίζεται στον κάθετο άξονα, την περιοδικότητα των δεδομένων και να επιλέξουμε την κατηγορία στην οποία ανήκει. Για παράδειγμα έστω φτιάχνουμε μια χρονοσειρά με τίτλο 'Πωλήσεις Πετρελαίου στην Ελλάδα την περίοδο 1992-2004' και τριμηνιαία δεδομένα. Τότε η εικόνα της εφαρμογής πρέπει να είναι κάπως έτσι:

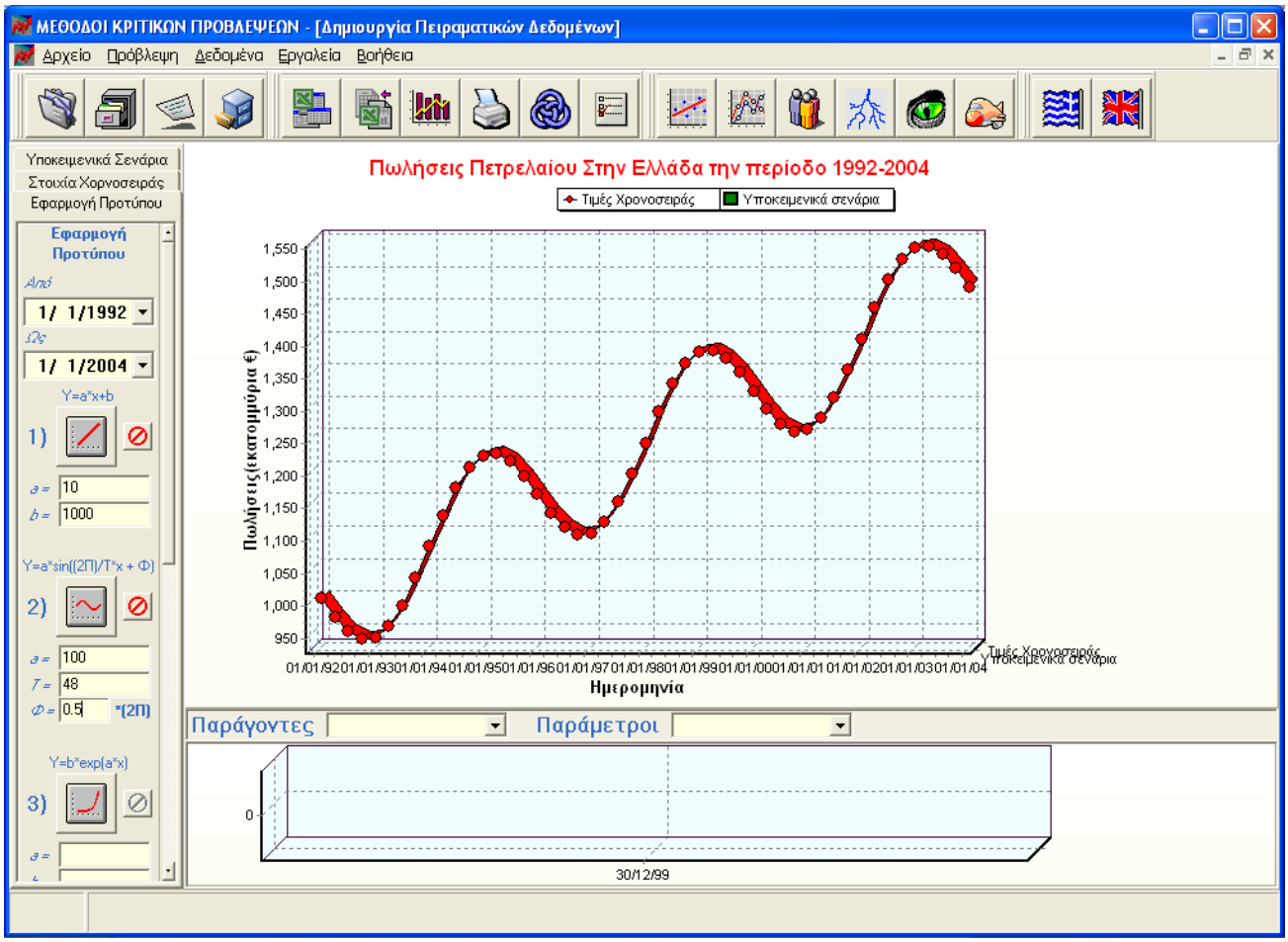



Πατώντας τώρα στο tab εφαρμογή προτύπου εμφανίζεται η εικόνα της επόμενης σελίδας. Στα αριστερά τώρα βλέπουμε τις νέες επιλογές που μας δίνονται. Εδώ μπορούμε να δώσουμε σχεδόν όποια μορφή θέλουμε στην πειραματική χρονοσειρά εφαρμόζοντας κάποια βασικά πρότυπα. Αρχικά διαλέγουμε για ποιο χρονικό διάστημα θέλουμε να εφαρμοστεί κάποιο πρότυπο. Στην περίπτωσή μας διαλέγουμε 1-1-1992 ως 1-1-2004. Τα βασικά πρότυπα που μπορούμε να εφαρμόσουμε είναι τα:

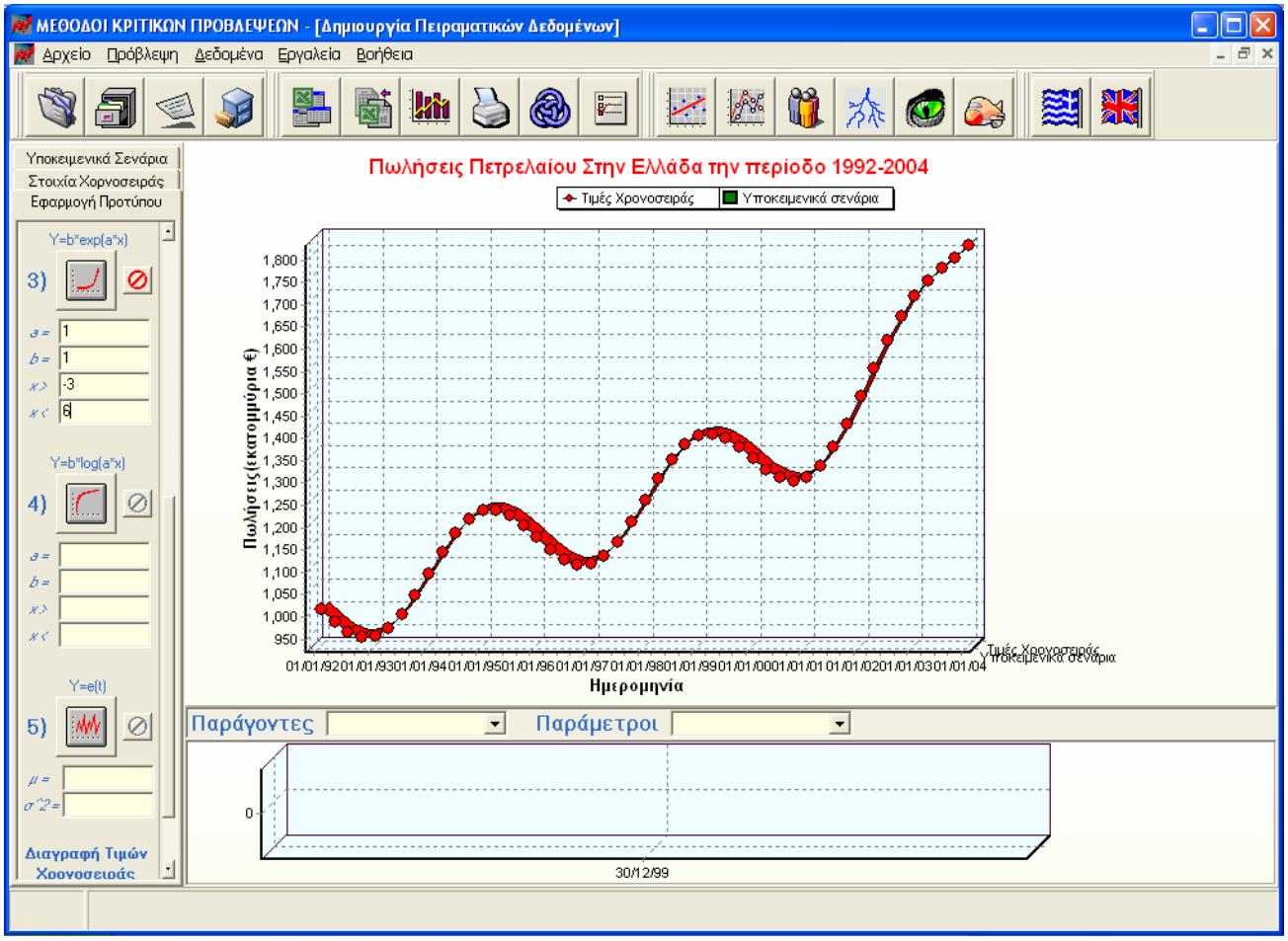
**1). Απλό γραμμικό πρότυπο.** Αποτέλεσμα αυτού είναι να προστεθεί στα δεδομένα της χρονοσειράς για την επιλεγμένη περίοδο η γραμμική συνάρτηση $y = a \cdot x + b$ με το χρήστη να ορίζει τα α και b. Στο παράδειγμά μας έστω επιλέγουμε α=10 και b=1000. Η εικόνα που θα πρέπει τώρα να βλέπουμε στην οθόνη μας φαίνεται παρακάτω.



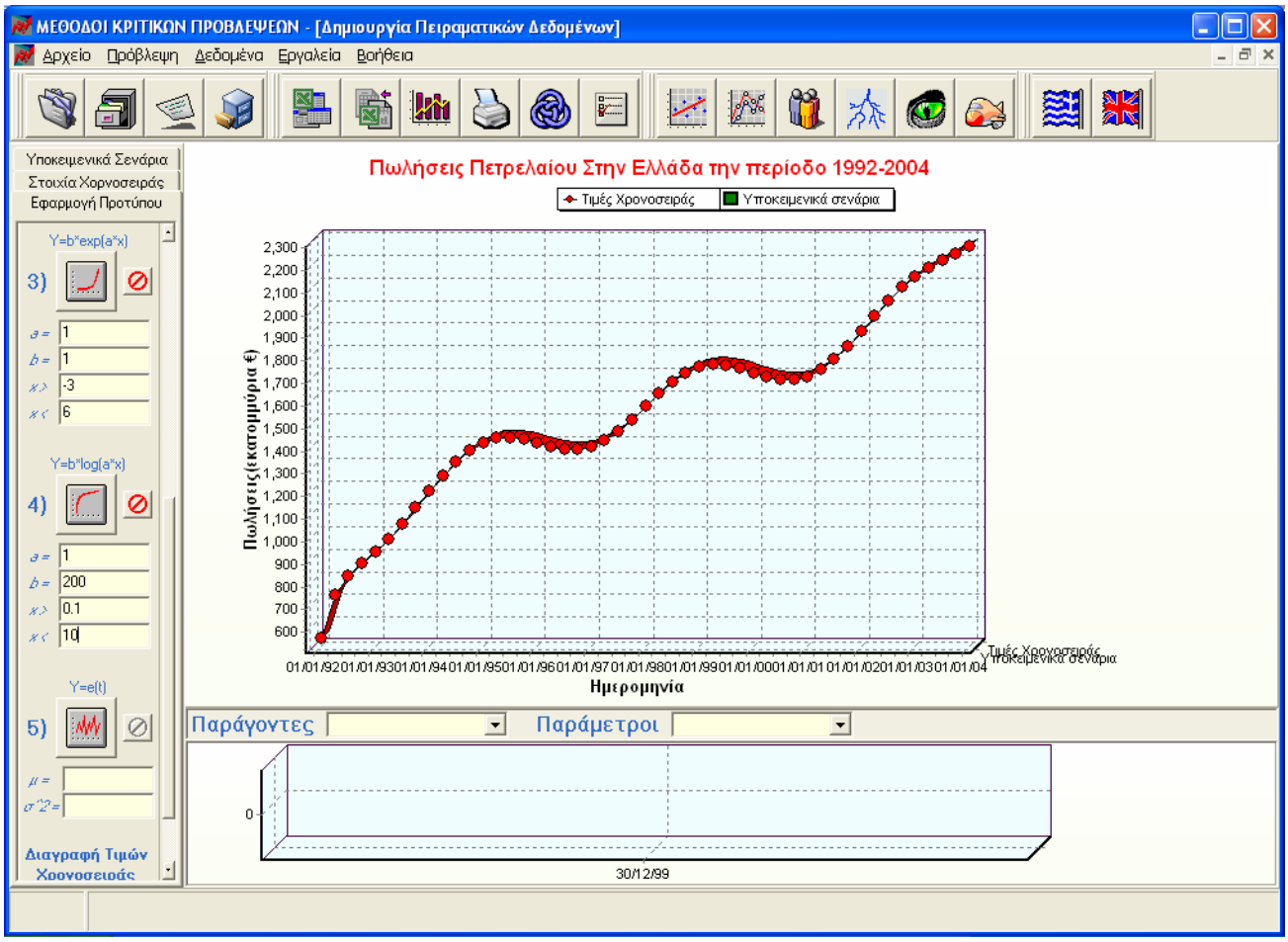
**2). Ημιτονοειδές πρότυπο.** Εδώ ο χρήστης μπορεί να εφαρμόσει κάποιο ημιτονοειδές πρότυπο ορίζοντας την περίοδο ταλάντωσης, το πλάτος και την αρχική φάση. Για παράδειγμα έστω εφαρμόζουμε ένα τέτοιο πρότυπο επιλέγοντας για πλάτος α=100 περίοδο Τ=48 μήνες και αρχική φάση Φ=0.5*2Π=Π. Η εικόνα που θα πρέπει τώρα να βλέπουμε στην οθόνη φαίνεται παρακάτω.



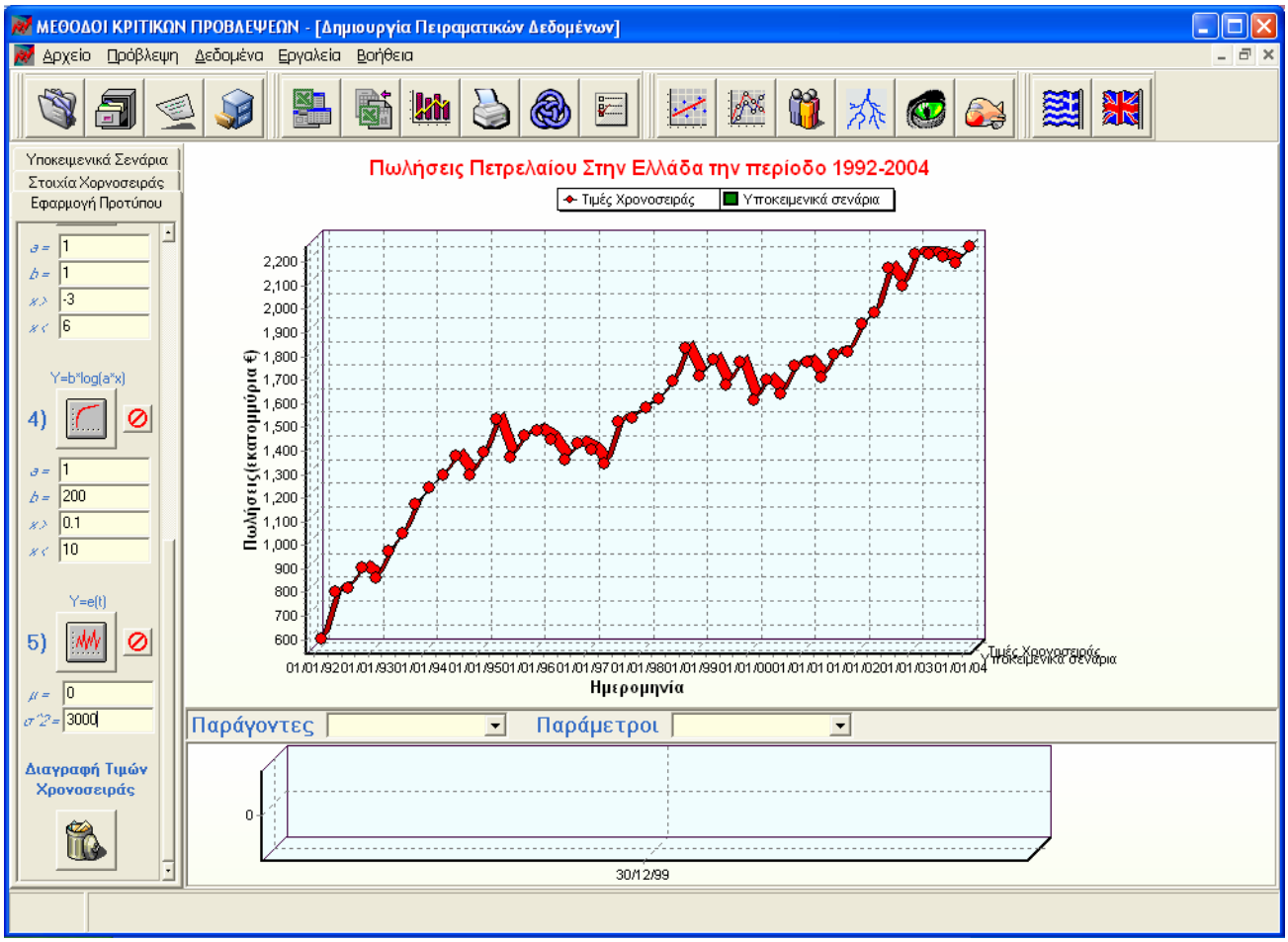
**3). Εκθετικό πρότυπο.** Η εφαρμογή του εκθετικού προτύπου γίνεται ορίζοντας τις παραμέτρους α και b της συνάρτησης $y = b \cdot e^{a \cdot x}$. Επίσης ορίζεται και το διάστημα στον άξονα x για το οποίο θέλουμε να εφαρμόσουμε τη συγκεκριμένη συνάρτηση. Επιλέγοντας για παράδειγμα α=1, b=1, και $x \in (-3,6)$ παίρνουμε το αποτέλεσμα που φαίνεται παρακάτω.



**4). Λογαριθμικό πρότυπο.** Όμοια με το εκθετικό πρότυπο ο χρήστης πρέπει να συμπληρώσει τις παραμέτρους α και b της συνάρτησης $y = b \cdot \log(a \cdot x)$ καθώς και το διάστημα του άξονα x για το οποίο επιθυμεί αυτή να εφαρμοστεί. Έστω επιλέγουμε α=1, b=200 και $x \in (-0.1,10)$. Το αποτέλεσμα φαίνεται παρακάτω.

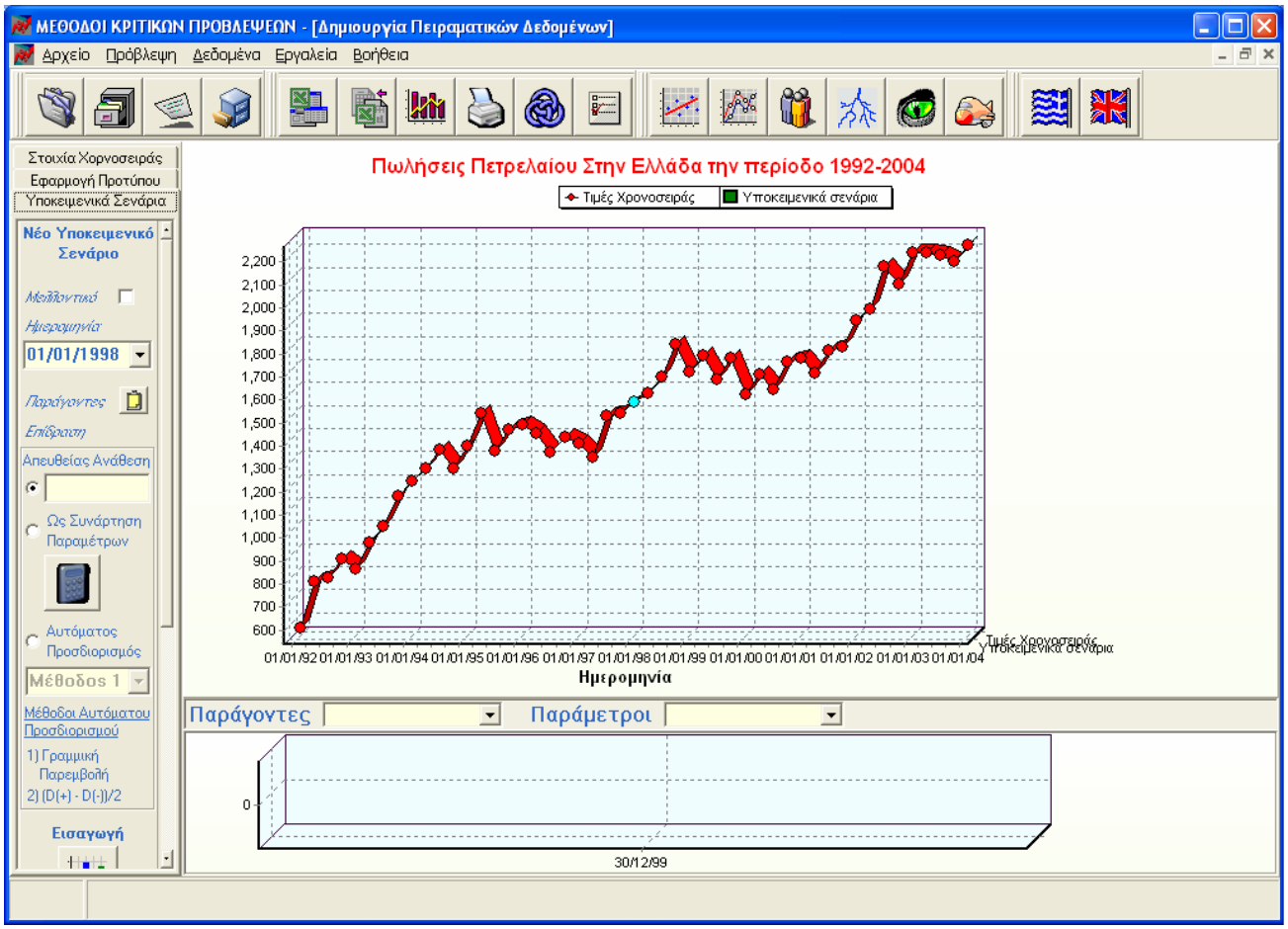


**5). Πρότυπο θορύβου.** Το συγκεκριμένο πρότυπο προσομοιάζει το θόρυβο που μία πραγματική χρονοσειρά μπορεί να παρουσιάζει. Ο θόρυβος είναι μια τυχαία μεταβλητή που ακολουθεί κανονική κατανομή και ο χρήστης πρέπει να δηλώσει τη μέση τιμή και τη διασπορά της. Στο συγκεκριμένο παράδειγμα έστω εφαρμόζουμε θόρυβο στη χρονοσειρά με μέση τιμή 0 και διασπορά 50. Το αποτέλεσμα φαίνεται παρακάτω.



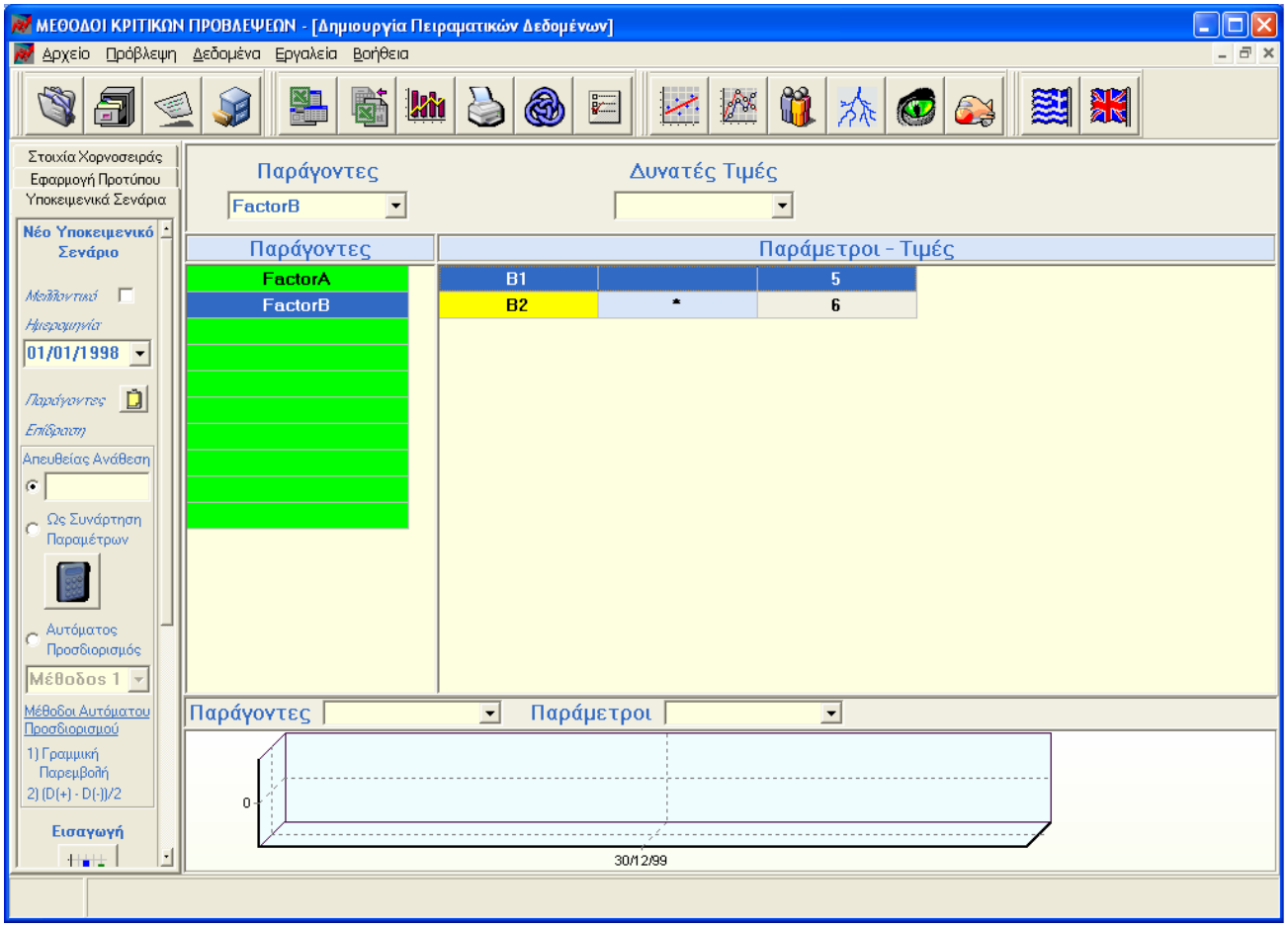

Το κουμπί με τίτλο 'Διαγραφή τιμών χρονοσειράς' διαγράφει όλες τις τιμές της χρονοσειράς. Τα κουμπιά με το εικονίδιο ⊘ ακυρώνουν διαδοχικές εφαρμογές του προτύπου δίπλα από το οποίο βρίσκονται. Επίσης βοηθητικά hint εμφανίζονται για κάθε κουμπί που υπάρχει στην εφαρμογή όταν ο χρήστης αφήσει για μικρό χρονικό διάστημα το mouse από πάνω του.

Το τρίτο tab με τίτλο 'Υποκειμενικά Σενάρια' επιτρέπει στο χρήστη τη δήλωση ιστορικών και μελλοντικών υποκειμενικών γεγονότων. Πατώντας στο συγκεκριμένο tab παίρνουμε την παρακάτω εικόνα. Εδώ μπορεί ο χρήστης να δηλώσει και να επεξεργαστεί ιστορικά αλλά και μελλοντικά υποκειμενικά γεγονότα.



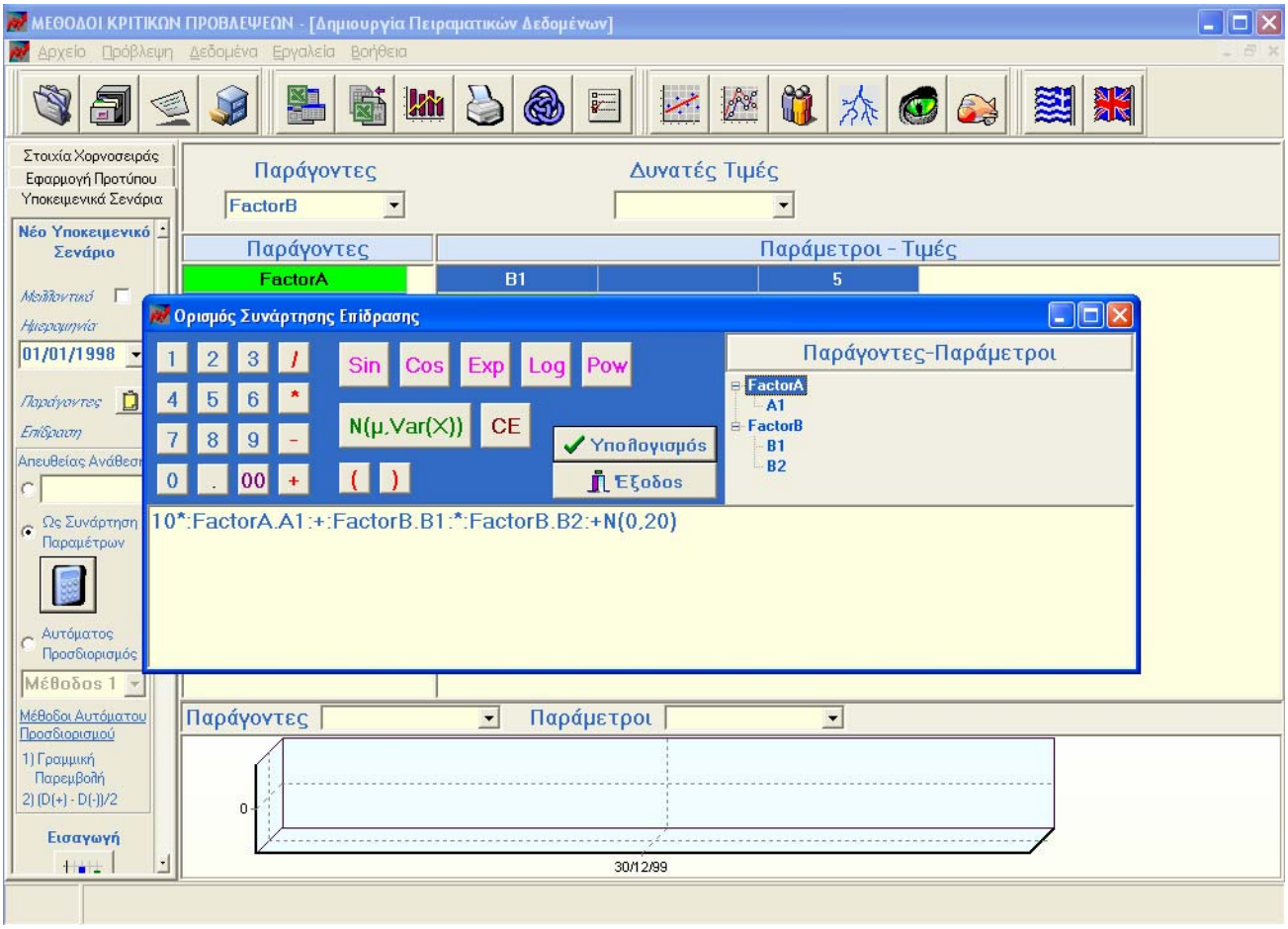

**Δήλωση νέου υποκειμενικού γεγονότος.** Αρχικά θα πρέπει ο χρήστης να επιλέξει αν πρόκειται για κάποιο ιστορικό ή μελλοντικό υποκειμενικό γεγονός σημειώνοντας το αντίστοιχο πεδίο. Στη συνέχεια θα πρέπει να επιλέξει μια ημερομηνία εισαγωγής του γεγονότος. Αυτό για τα ιστορικά γεγονότα γίνεται είτε επιλέγοντας μια ημερομηνία από το drop down box είτε πατώντας πάνω σε κάποιο σημείο της χρονοσειράς οπότε και αυτό χρωματίζεται. Για τα μελλοντικά γεγονότα η ημερομηνία εισαγωγής επιλέγεται μόνο από το drop down box. Στη συνέχεια θα πρέπει ο χρήστης να δηλώσει τους παράγοντες οι οποίοι αποτελούν το

προς εισαγωγή γεγονός. Η δήλωση των παραγόντων γίνεται πατώντας πάνω στο κουμπί  οπότε και εμφανίζεται η παρακάτω φόρμα.

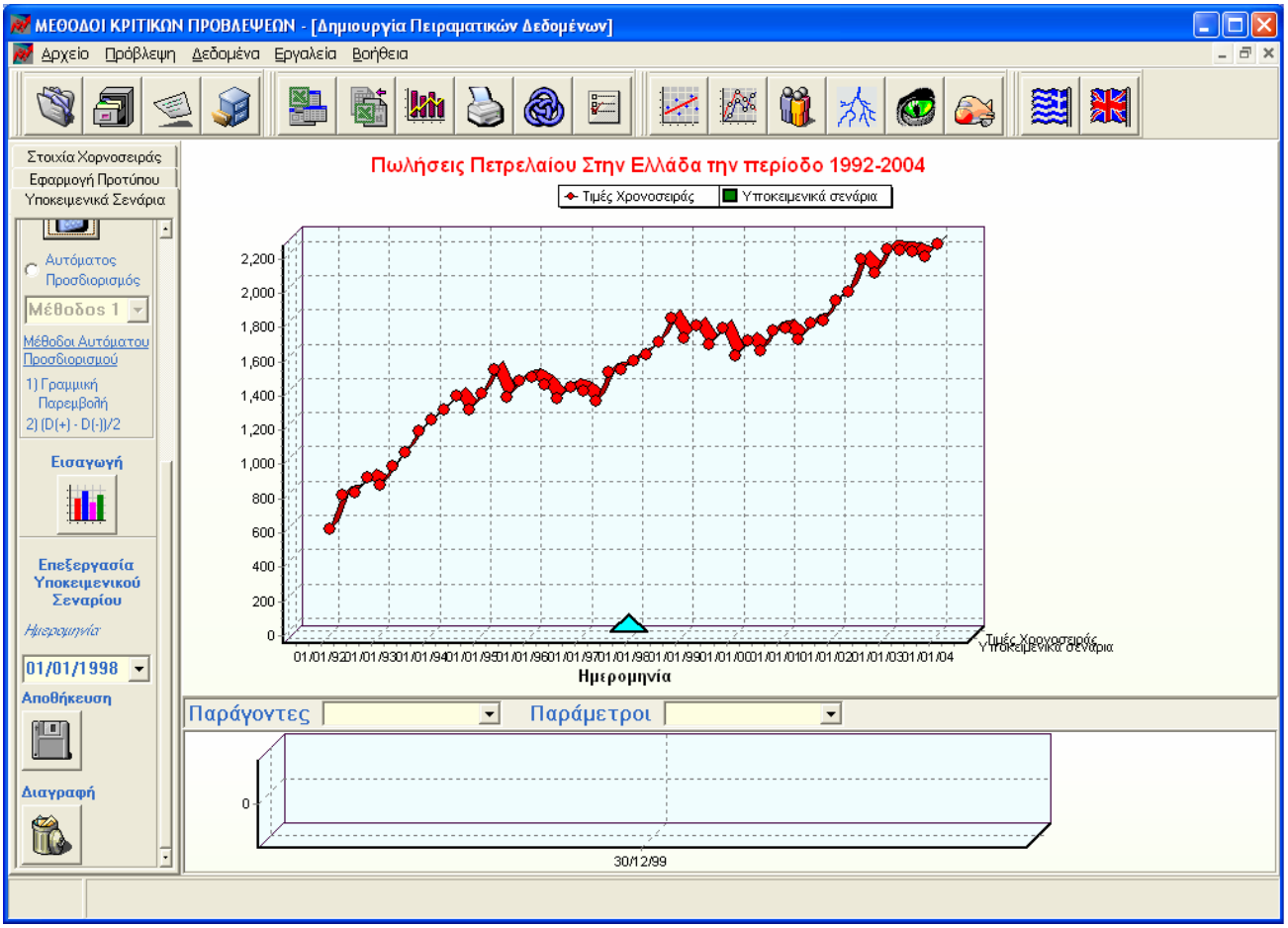



Εδώ από το drop down box με τίτλο 'Παράγοντες' επιλέγουμε με ποιους παράγοντες θέλουμε να συνθέσουμε το γεγονός προς εισαγωγή. Από το δεύτερο drop down box με τίτλο Δυνατές τιμές μπορούμε να επιλέξουμε κάποια δηλωμένη δυνατή τιμή για κάποια παράμετρο. Για όσες παραμέτρους υπάρχουν δυνατές τιμές εμφανίζεται ένα * στο μεσαίο πεδίο όπως φαίνεται παραπάνω. Ένα παράγοντα μπορούμε να τον διαγράψουμε αφού πρώτα τον επιλέξουμε και πατώντας απλά το delete. Αφού δηλώσουμε τους παράγοντες του υποκειμενικού γεγονότος πρέπει στη συνέχεια αν πρόκειται για ιστορικό γεγονός να δηλώσουμε την επίδρασή του στη χρονοσειρά. Αυτό μπορεί να γίνει με τρεις τρόπους όπως φαίνεται παραπάνω. Είτε αναθέτοντας απευθείας μια τιμή, είτε δηλώνοντας ότι η επίδραση υπολογίζεται από μια συνάρτηση των παραμέτρων είτε αφήνοντας την εφαρμογή να την προσδιορίσει αυτόματα με δύο μεθόδους.

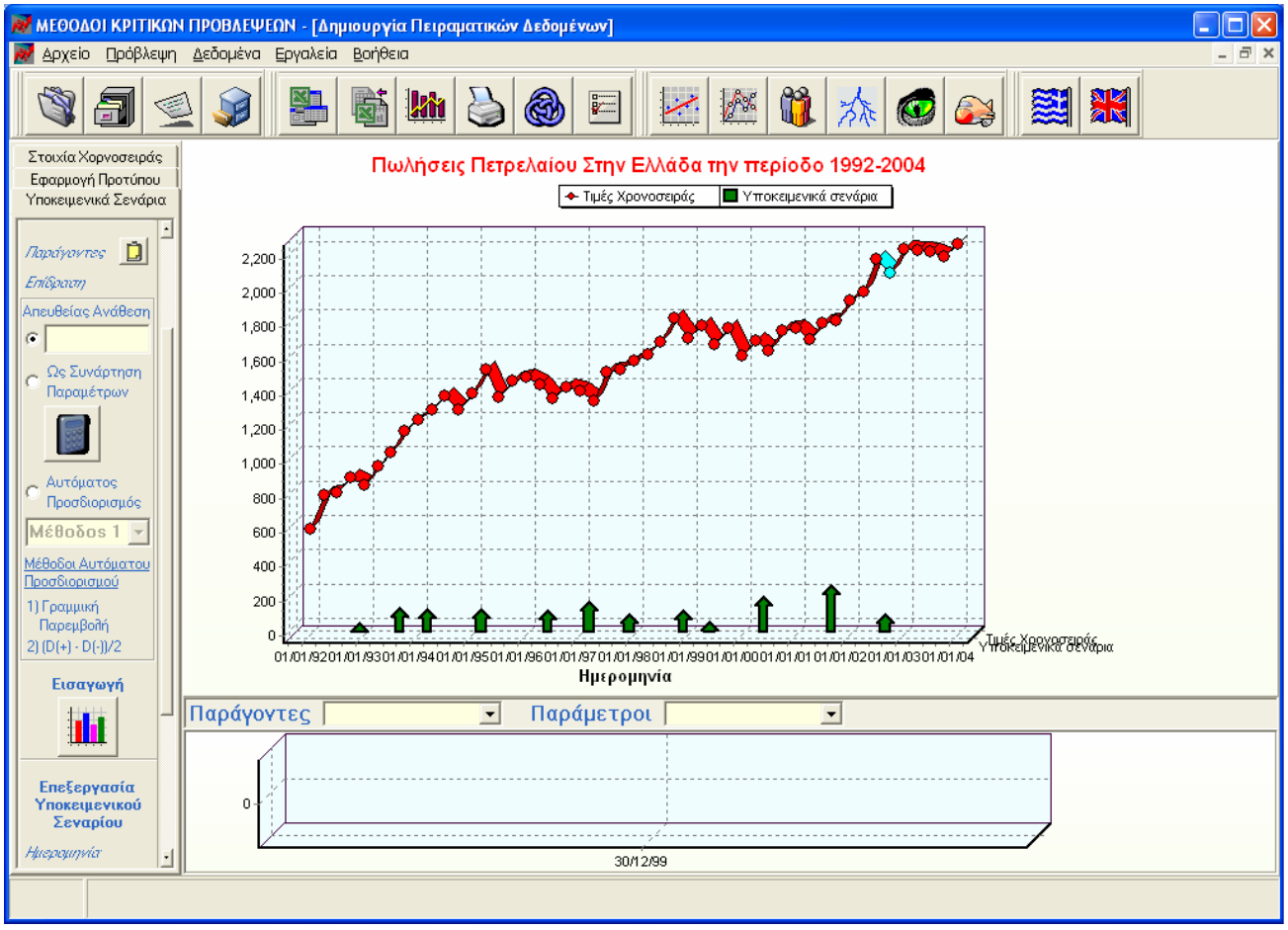
Η πρώτη είναι μια απλή γραμμική παρεμβολή των τιμών της χρονοσειράς προηγούμενων ετών. Η επίδραση προκύπτει ως η διαφορά της αναμενόμενης τιμής σύμφωνα με την γραμμική παρεμβολή από την πραγματική. Η δεύτερη ένας πολύ απλός υπολογισμός. $impact = \frac{D(+) - D(-)}{2}$. Όταν θέλουμε να δηλώσουμε την επίδραση ενός γεγονότος ως συνάρτηση των παραμέτρων πατάμε το κουμπί  αφού πρώτα επιλέξουμε την αντίστοιχη μέθοδο από το κατάλληλο radio button. Τότε εμφανίζεται η φόρμα που φαίνεται παρακάτω η οποία μας δίνει τη δυνατότητα να φτιάξουμε μια όσο πολύπλοκη συνάρτηση θέλουμε. Στο παράδειγμά μας έχει δηλωθεί μια σχετικά απλή συνάρτηση των παραμέτρων. Οι παράμετροι εισάγονται στη φόρμα υπολογισμού κάνοντας απλά διπλό click στο όνομά τους.



Πατώντας το κουμπί υπολογισμός βλέπουμε το αποτέλεσμα στο κουτί της απευθείας ανάθεσης. Τώρα είμαστε έτοιμοι να εισάγουμε το νέο γεγονός αυτό γίνεται πατώντας το κουμπί  . Το αποτέλεσμα φαίνεται παρακάτω. Με τον ίδιο τρόπο εισάγουμε και άλλα ιστορικά και μελλοντικά γεγονότα.




**Επεξεργασία υποκειμενικών γεγονότων.** Η επεξεργασία ενός γεγονότος γίνεται απλά επιλέγοντας το γεγονός, κάνοντας click πάνω στο αντίστοιχο βέλος ή από το drop down box. Αν έχουμε επιλέξει ένα γεγονός αυτό χρωματίζεται διαφορετικά. Μπορούμε να αλλάξουμε ότι θέλουμε για το γεγονός αυτό με όμοιο τρόπο όπως αυτόν που περιγράφτηκε προηγουμένως κατά τη διαδικασία εισαγωγής. Στη συνέχεια όμως θα πρέπει να αποθηκεύσουμε τις αλλαγές πατώντας το κουμπί με τίτλο 'Αποθήκευση'. Αν δεν το κάνουμε αυτό οι αλλαγές θα χαθούν. Με το κουμπί διαγραφή διαγράφουμε από τη χρονοσειρά το επιλεγμένο γεγονός. Μετά από την εισαγωγή αρκετών γεγονότων παίρνουμε την παρακάτω εικόνα:

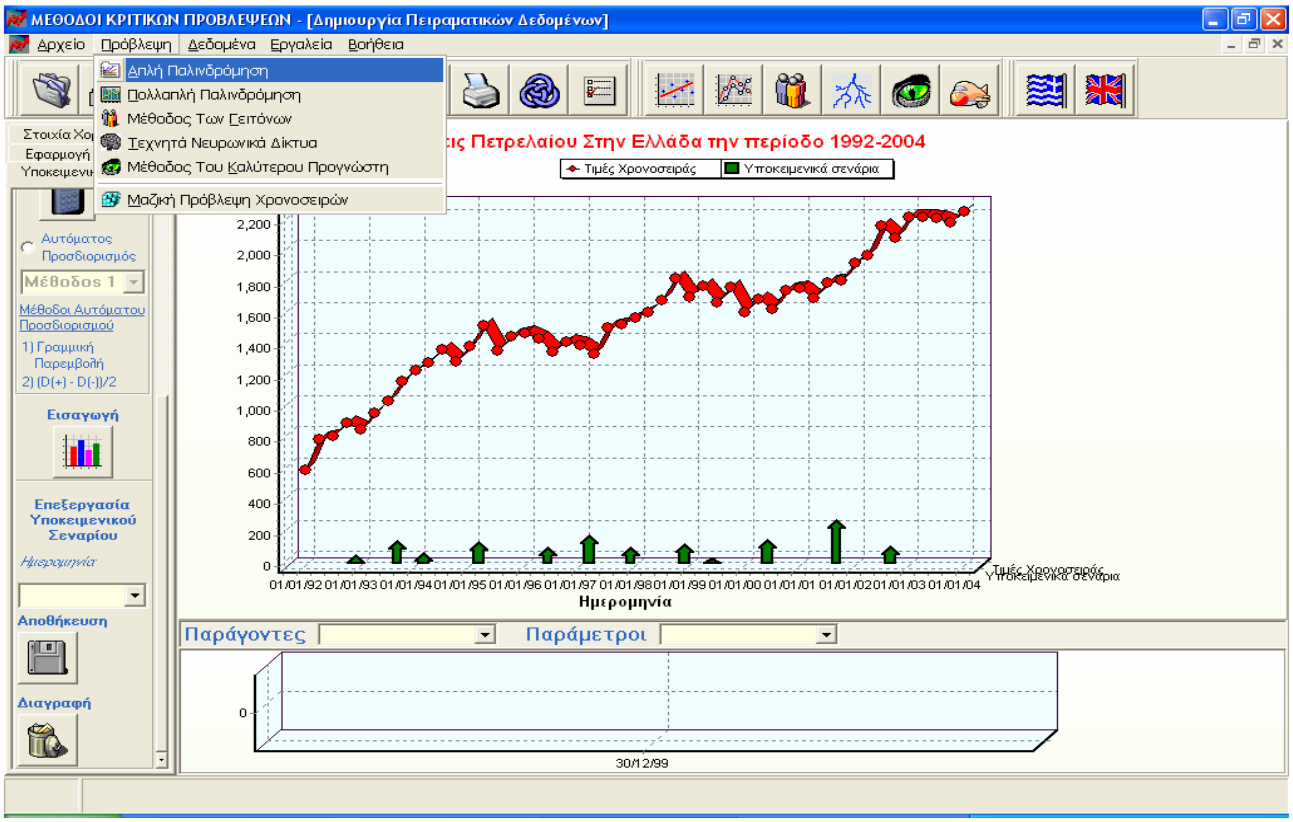



Εδώ έχουμε εισάγει και δύο μελλοντικά γεγονότα για τα οποία και θέλουμε να προβλέψουμε την επίδραση που θα έχουνε στη χρονοσειρά.

### 4.3.3 Πρόβλεψη επίδρασης μελλοντικών Υποκειμενικών Γεγονότων

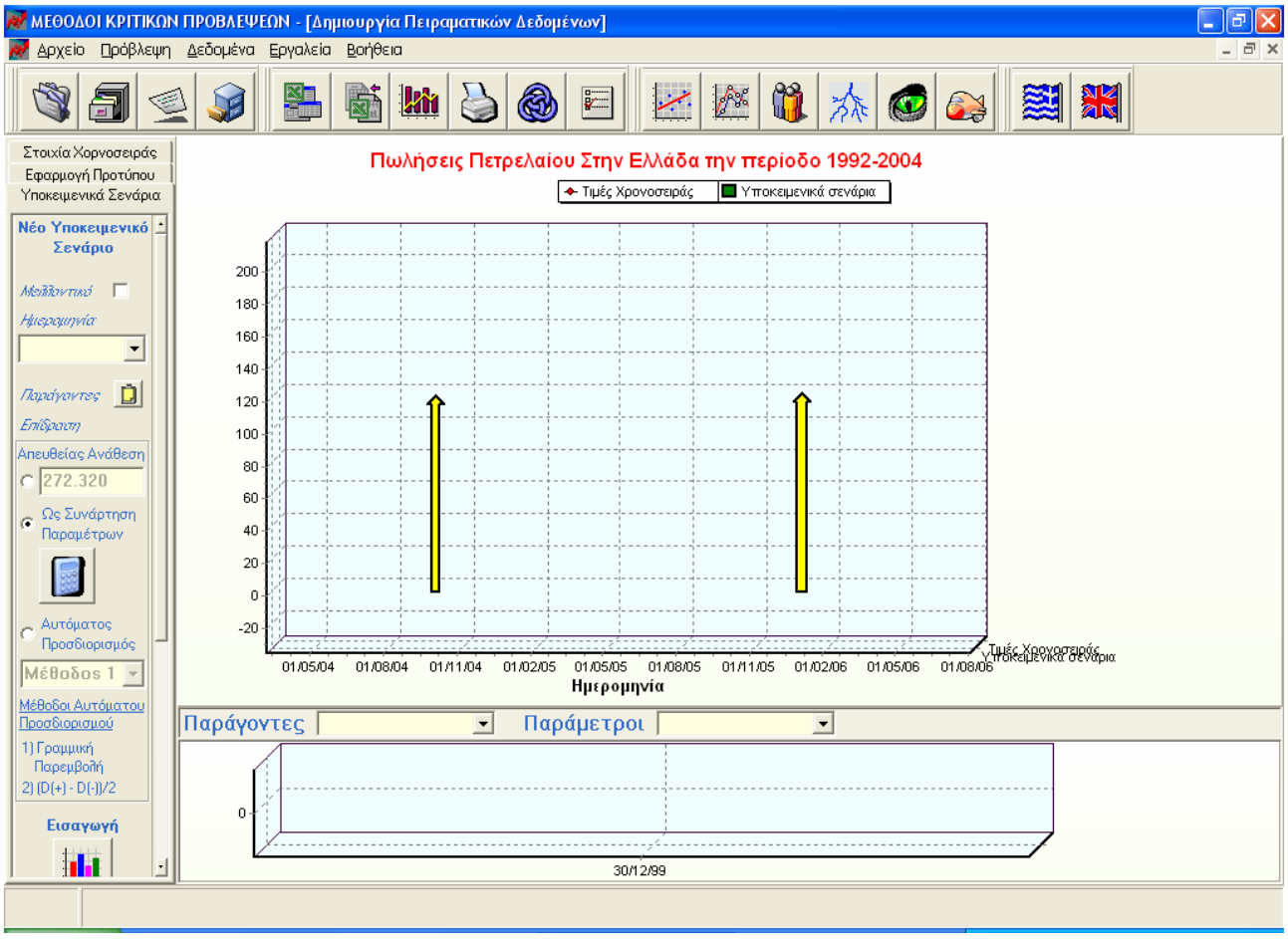
Υπάρχουν τρεις μέθοδοι πρόβλεψης τις οποίες μπορεί να εφαρμόσει ο χρήστης. Αυτές είναι η Απλή Παλινδρόμηση, Η Πολλαπλή Παλινδρόμηση και η πρόβλεψη με τη χρήση Τεχνικών Νευρωνικών Δικτύων. Για να εφαρμοστεί κάθε μία από αυτές πρέπει να πληρούνται κάποιες απαραίτητες προϋποθέσεις. Αρχικά, θα πρέπει ο χρήστης να έχει επιλέξει τη φόρμα με την χρονοσειρά που τον ενδιαφέρει. Επειδή πρόκειται για μια εφαρμογή πολλαπλών φορμών θα πρέπει η συγκεκριμένη χρονοσειρά να βρίσκεται πάνω από κάθε άλλη φόρμα. Έπειτα θα πρέπει να έχει δηλωθεί στη συγκεκριμένη χρονοσειρά τουλάχιστον ένα μελλοντικό υποκειμενικό γεγονός. Ακόμα θα πρέπει να υπάρχουν και κατάλληλα σε είδος και αριθμό ιστορικά υποκειμενικά γεγονότα ανάλογα με το ποια μέθοδο θέλουμε να εφαρμόσουμε.

**Απλή Παλινδρόμηση.** Η μέθοδος αυτή εφαρμόζεται πατώντας το κουμπί  ή επιλέγοντας από το βασικό menu την κατάλληλη επιλογή όπως φαίνεται παρακάτω:



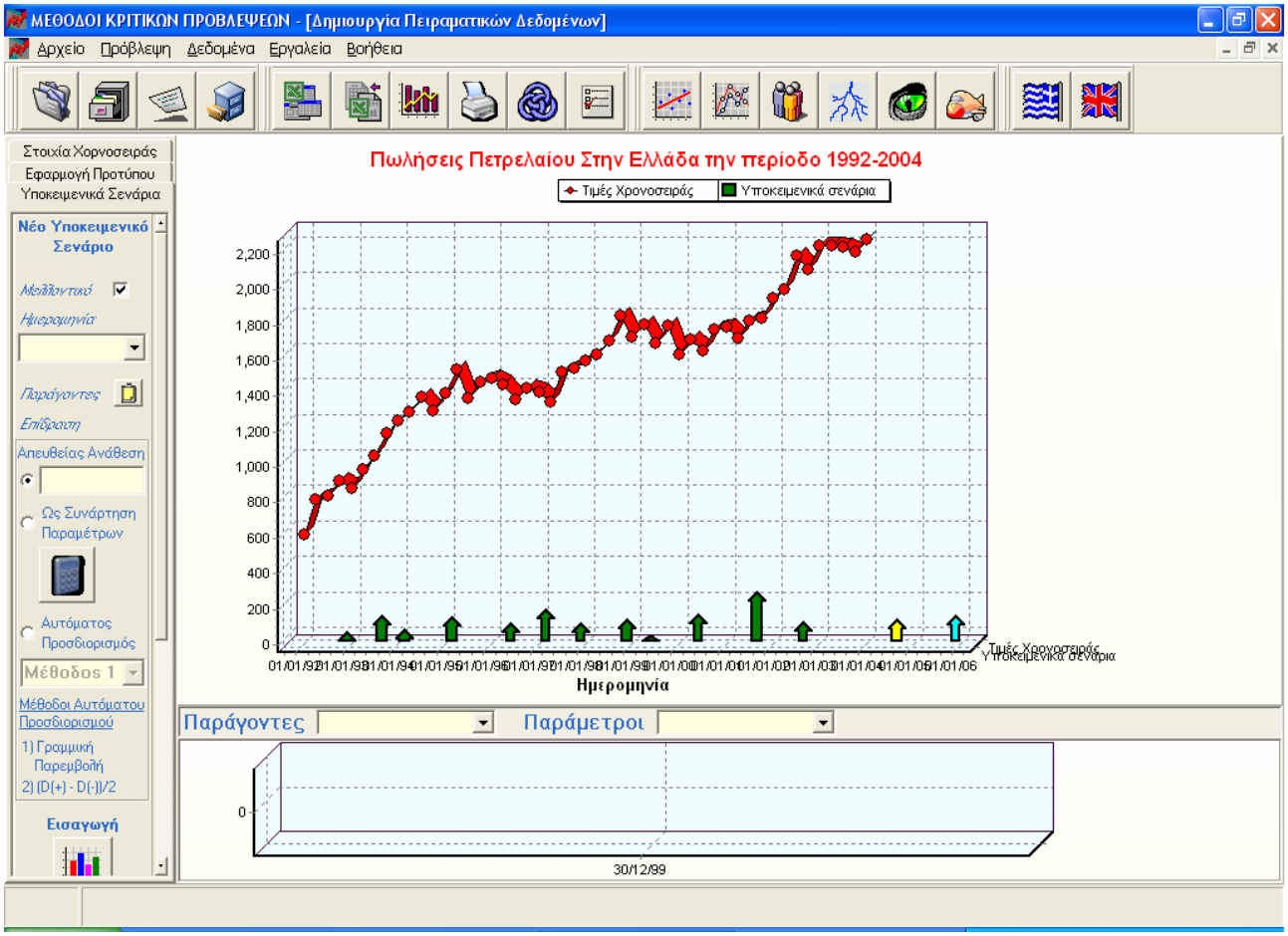
Είναι η απλούστερη μέθοδος και προσπαθεί να προβλέψει την επίδραση κάθε παραμέτρου ξεχωριστά στη χρονοσειρά. Για την πρόβλεψη της επίδρασης ενός μελλοντικού υποκειμενικού γεγονότος κάνει υπέρθεση των αποτελεσμάτων. Προκειμένου λοιπόν να εφαρμοστεί θα πρέπει να έχουμε κατάλληλα δεδομένα στη χρονοσειρά έτσι ώστε να μπορεί να εφαρμοστεί η απλή παλινδρόμηση για κάθε παράμετρο. Θα πρέπει δηλαδή για τουλάχιστον μία παράμετρο να υπάρχουν αρχικά, τουλάχιστον δύο υποκειμενικά γεγονότα τα οποία να περιέχουν μόνο τη συγκεκριμένη παράμετρο με μη μηδενική τιμή. Έτσι η επίδραση των γεγονότων αυτών θα εξαρτάται μόνο από την παράμετρο αυτή και θα μπορεί να εκφραστεί από την ευθεία $y = a \cdot x + \beta$. Μετά τον υπολογισμό των α και β αφαιρείται η επίδραση της παραμέτρου αυτής από όλα τα υποκειμενικά γεγονότα και συνεχίζεται η ίδια διαδικασία μέχρι να υπολογιστούν οι ευθείες $y = a \cdot x + \beta$ για όλες τις παραμέτρους. Η πρόβλεψη όπως είπαμε γίνεται με υπέρθεση των αποτελεσμάτων που δίνει η κάθε εξίσωση ξεχωριστά για κάθε παράμετρο.

Στη συγκεκριμένη περίπτωση το αποτέλεσμα φαίνεται παρακάτω:



Στο προηγούμενο screenshot έχουμε κάνει zoom στα μελλοντικά υποκειμενικά σενάρια. Η λειτουργία αυτή γίνεται με το mouse απλά επιλέγοντας την επιθυμητή περιοχή. Για το πρώτο μελλοντικό υποκειμενικό γεγονός η πρόβλεψη είναι 121.325. Οι παράμετροι που έχουν δηλωθεί για το γεγονός αυτό παίρνουν τιμές A1=8, B1=4 και B2=6. Η αναμενόμενη τιμή σύμφωνα με τη συνάρτηση επίδρασης που έχουμε ορίσει είναι 10*:FactorA.A1:+:FactorB.B1:*:FactorB.B2:+N(0,20)=104+N(0,20). Η τιμή πρόβλεψης παρατηρούμε ότι δεν είναι αρκετά ικανοποιητική αφού δεν βρίσκεται μέσα στα όρια του θορύβου. Για το δεύτερο υποκειμενικό γεγονός προβλέψαμε ότι θα έχει επίδραση ίση με 122,834. Η αναμενόμενη τιμή σύμφωνα με τη συνάρτηση επίδρασης είναι:
10*:FactorA.A1:+:FactorB.B1:*:FactorB.B2:+N(0,20) = 134+N(0,20) έχοντας ορίσει τιμές για τις παραμέτρους: A1=12, B1=7, B2=2. Και σε αυτήν την περίπτωση δεν μας ικανοποιεί ιδιαίτερα το αποτέλεσμα αφού βρίσκεται έξω από τα όρια του θορύβου.

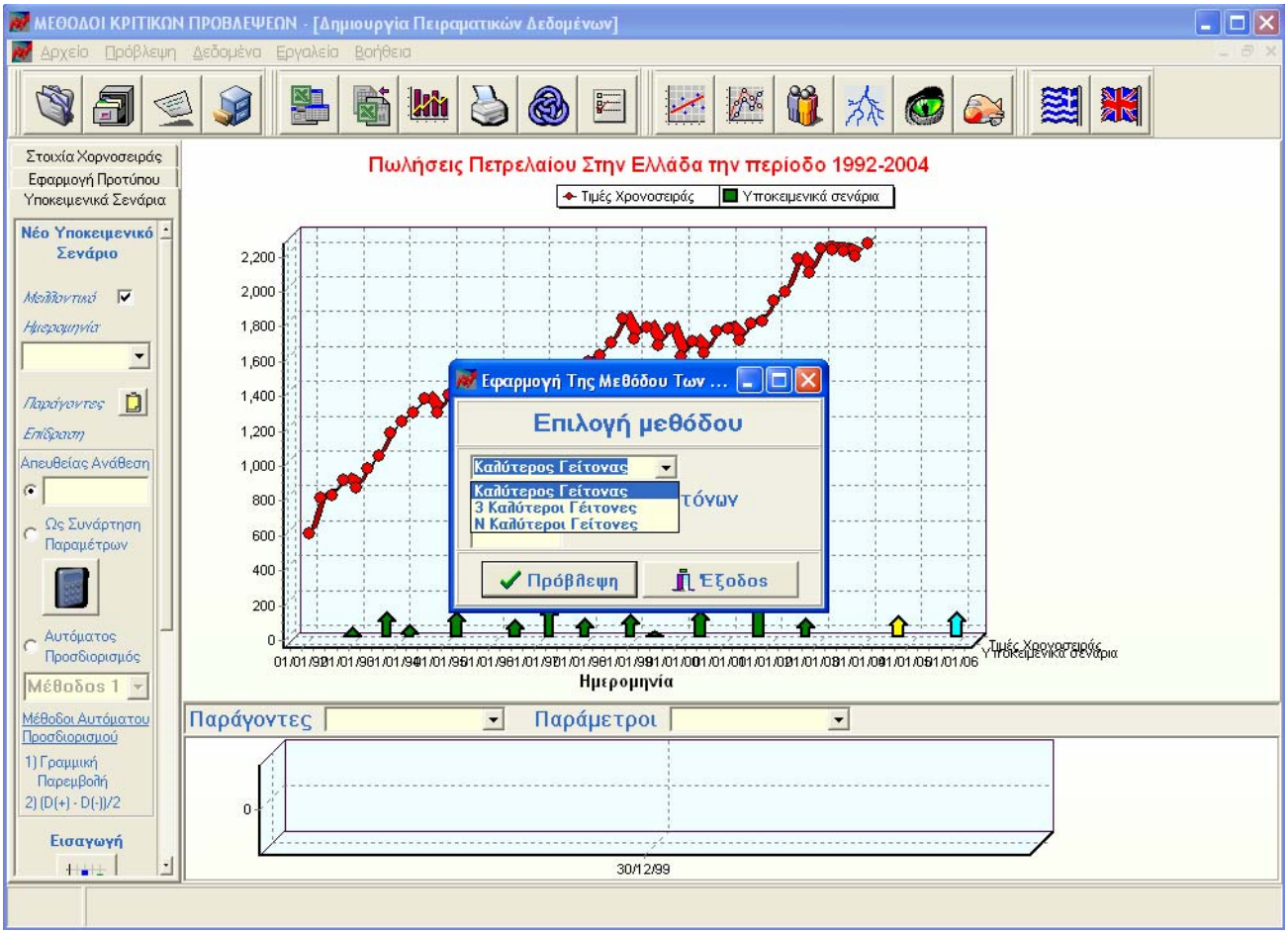
**Πολλαπλή Παλινδρόμηση.** Αυτή είναι η δεύτερη μέθοδος πρόβλεψης και βασίζεται στην επίλυση του συστήματος $\hat{\beta} = (X^{'} \cdot X)^{-1} \cdot (X^{'} \cdot Y)$ όπου X είναι ο Nx(M+1) πίνακας όπου N ο αριθμός των ιστορικών υποκειμενικών γεγονότων και M ο συνολικός αριθμός των διαφορετικών παραμέτρων που εμφανίζονται στη χρονοσειρά. Ο Y είναι ο Nx1 πίνακας και κάθε στοιχείο του είναι η επίδραση του αντίστοιχου υποκειμενικού γεγονότος στη χρονοσειρά. Σύμφωνα με τη μέθοδο της πολλαπλής παλινδρόμησης η πρόβλεψη της επίδρασης ενός μελλοντικού γεγονότος γίνεται σύμφωνα με την εξίσωση $\hat{Y} = X \cdot \hat{\beta}$. Η μέθοδος εφαρμόζεται όταν ο χρήστης πατήσει το κουμπί  ή την αντίστοιχη επιλογή από το βασικό menu. Το αποτέλεσμα για το προηγούμενο παράδειγμα φαίνεται στην επόμενη σελίδα. Τώρα για το πρώτο μελλοντικό υποκειμενικό γεγονός έχουμε προβλέψει την τιμή 118.807 και η αναμενόμενη είναι 104+N(0,20) ενώ για το δεύτερο υποκειμενικό γεγονός έχουμε προβλέψει την τιμή 141.347 ενώ η αναμενόμενη τιμή είναι 134+N(0,20). Η δεύτερη πρόβλεψη είναι αρκετά καλή ενώ η πρώτη είναι μεν καλύτερη από αυτή της μεθόδου της απλής παλινδρόμησης δεν είναι όμως και αρκετά ικανοποιητική.




**Μέθοδος των Γειτόνων**. Η τρίτη μέθοδος πρόβλεψης βασίζεται στην εξαγωγή προβλέψεων με βάση την ομοιομορφία που παρουσιάζουν τα μελλοντικά με τα ιστορικά δεδομένα όσον αφορά τις τιμές των παραμέτρων τους. Για κάθε ένα μελλοντικό γεγονός τα ιστορικά δεδομένα ταξινομούνται κατά φθίνουσα σειρά, από αυτό που παρουσιάζει τη μεγαλύτερη ομοιότητα με το μελλοντικό γεγονός σε αυτό που παρουσιάζει τη μικρότερη. Το πρώτο στη λίστα είναι ο καλύτερος γείτονας και το τελευταίο ο χειρότερος. Η πρόβλεψη γίνεται με τρεις τρόπους. Ο πρώτος είναι η επίδραση του μελλοντικού γεγονότος να οριστεί ίδια με αυτήν του καλύτερου γείτονα. Δηλαδή Impact = Γείτονας1. Ο δεύτερος, η επίδραση να υπολογιστεί ως Impact = 0.5*Γείτονας1 + 0.25*Γείτονας2 + 0.25*Γειτονας3, να περιλαμβάνει δηλαδή τους τρεις καλύτερους γείτονες με βάρη 1/2, 1/4 και 1/4 και ο τρίτος να ορίσει ο χρήστης πόσους γείτονες θέλει να περιληφθούν στον υπολογισμό (έστω N), έχοντας ίσα βάρη (1/N) από τον καλύτερο στο χειρότερο. Παραπάνω όπου Γείτονας εννοούμε την επίδραση του αντίστοιχου υποκειμενικού γεγονότος στη

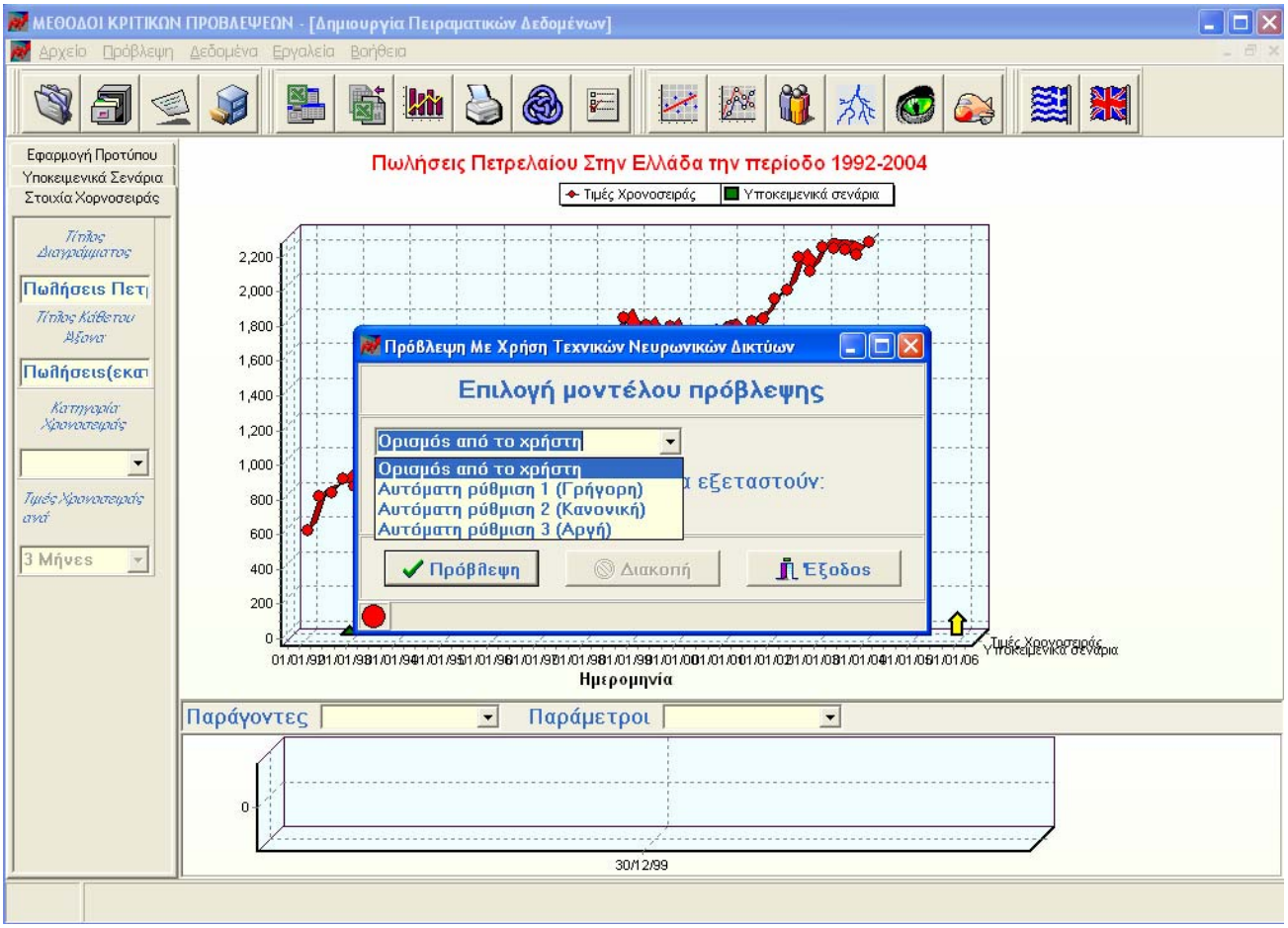
χρονοσειρά. Είναι προφανές ότι κάθε μελλοντικό υποκειμενικό γεγονός θα έχει και διαφορετική λίστα γειτόνων.

Η σύγκριση ενός γείτονα με ένα μελλοντικό υποκειμενικό γεγονός γίνεται με τον εξής τρόπο. Για κάθε κοινή παράμετρο ενός κοινού παράγοντα των δύο γεγονότων η ποσότητα |(Μελλοντικό γεγονός.Παράμετρος – Ιστορικό γεγονός.Παράμετρος)|/|Μελλοντικό γεγονός.Παράμετρος| προστίθεται σε μια μεταβλητή κόστους που δείχνει πόσο «απέχει» ο συγκεκριμένος γείτονας από το γεγονός του οποίου την επίδραση θέλουμε να προβλέψουμε. Για κάθε παράμετρο η οποία παρουσιάζεται μόνο σε ένα από τα δύο γεγονότα η ποσότητα 1 προστίθεται στη μεταβλητή κόστους. Στο τέλος συγκρίνονται οι μεταβλητές κόστους όλων των γειτόνων και αυτοί ταξινομούνται ανάλογα. Παρακάτω φαίνεται η φόρμα επιλογής μιας εκ των τριών μεθόδων πατώντας το κουμπί. 



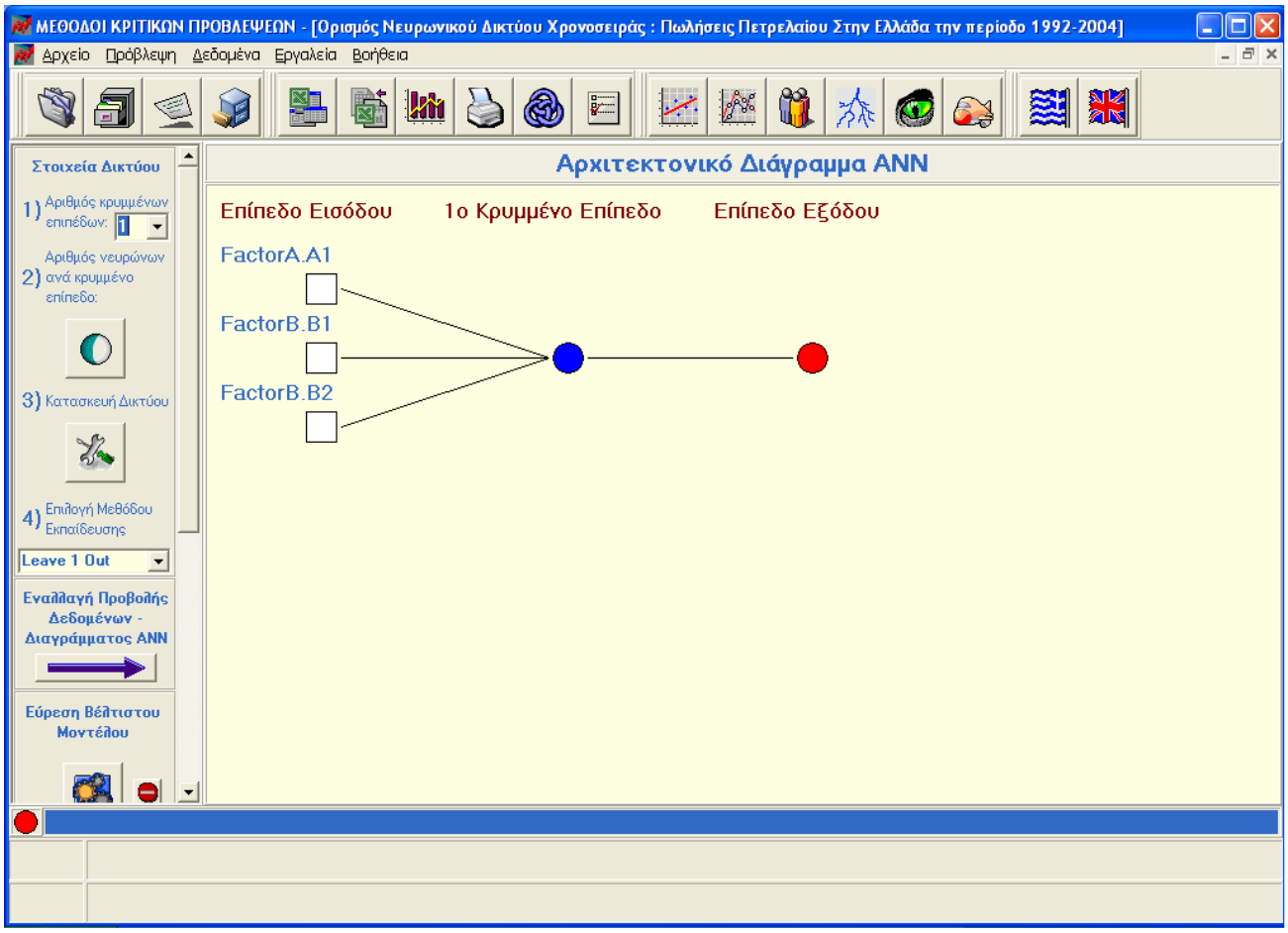
Τώρα για το πρώτο μελλοντικό υποκειμενικό γεγονός με τη μέθοδο του καλύτερου γείτονα έχουμε προβλέψει την τιμή 100.136 και η αναμενόμενη είναι 104+N(0,20) ενώ για το δεύτερο υποκειμενικό γεγονός έχουμε προβλέψει την τιμή 105.145 ενώ η αναμενόμενη τιμή είναι 134+N(0,20).

**Τεχνητά Νευρωνικά Δίκτυα.** Η τέταρτη μέθοδος πρόβλεψης αφορά τα τεχνητά νευρωνικά δίκτυα και εφαρμόζεται θεωρητικά όπως είδαμε στο κεφάλαιο 3. Ο χρήστης πρέπει να πατήσει το κουμπί  ή την αντίστοιχη επιλογή από το βασικό menu οπότε και εμφανίζεται η φόρμα επιλογής μεθόδου εκπαίδευσης και πρόβλεψης που φαίνεται παρακάτω.



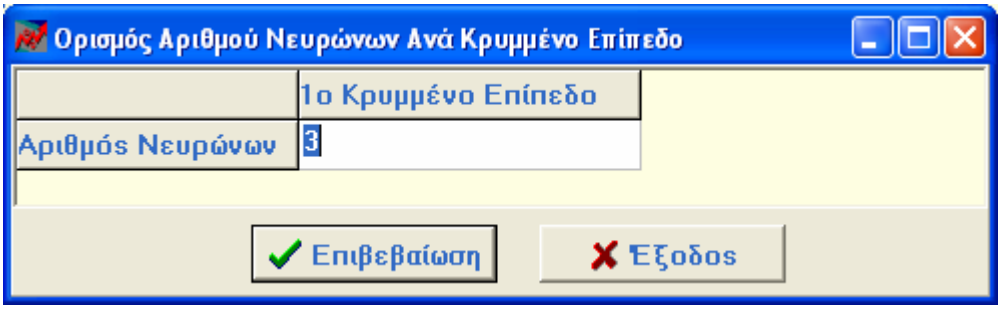

Εδώ ο χρήστης έχει τέσσερις επιλογές. Η πρώτη είναι να ορίσει μόνος το μοντέλο του νευρωνικού δικτύου που θέλει να χρησιμοποιήσει και να το εκπαιδεύσει ενώ οι άλλες τρεις περιλαμβάνουν αυτόματες μεθόδους πρόβλεψης. Από αυτές η Αυτόματη Ρύθμιση 1(Γρήγορη) προσπαθεί να προβλέψει την επίδραση

των μελλοντικών υποκειμενικών γεγονότων με ένα δίκτυο 2 κρυμμένων επιπέδων πλήρως συνδεδεμένο το οποίο και εκπαιδεύει μία μόνο φορά. Ανεξάρτητα με το τοπικό ελάχιστο το οποίο επιτυγχάνεται η πρόβλεψη γίνεται. Η Αυτόματη Ρύθμιση 2(Κανονική) χρησιμοποιεί πάλι ένα δίκτυο 2 κρυμμένων επιπέδων αλλά το εκπαιδεύει 10 φορές και από αυτές κρατάει αυτήν στην οποία το δίκτυο εκπαιδεύτηκε στο μικρότερο τοπικό ελάχιστο. Τέλος η Αυτόματη Ρύθμιση 3(Αργή) εκπαιδεύει όλα τα δίκτυα μέχρι 2 κρυμμένων επιπέδων από 10 φορές το καθένα και κρατάει πάλι το δίκτυο το οποίο εκπαιδεύεται στο μικρότερο τοπικό ελάχιστο. Με βάση τα παραπάνω η τρίτη μέθοδος είναι πιθανό να δώσει και τα καλύτερα αποτελέσματα. Παρακάτω θα εξετάσουμε τη διαδικασία ορισμού ενός νευρωνικού δικτύου από το χρήστη. Επιλέγοντας λοιπόν την επιλογή ορισμός από το χρήστη εμφανίζεται η παρακάτω φόρμα.

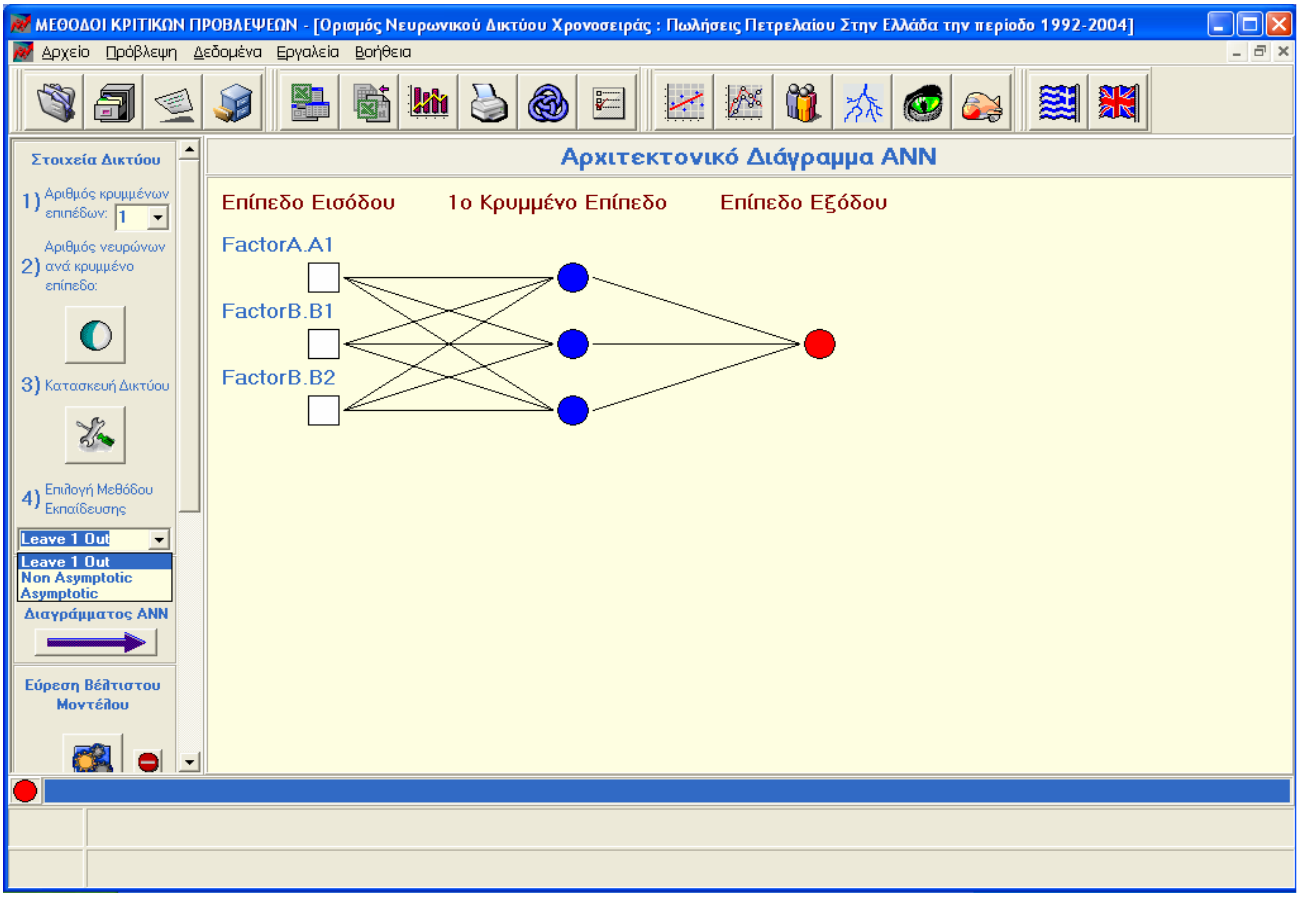



Η επιλογή του μοντέλου πρόβλεψης μπορεί να γίνει με δύο τρόπους. Πρώτον, από τον ίδιο τον χρήστη επιλέγοντας αρχικά τον αριθμό των κρυμμένων επιπέδων από το αντίστοιχο drop down box και

στη συνέχεια πατώντας το κουμπί.  Από τη νέα φόρμα που εμφανίζεται και φαίνεται παρακάτω μπορεί ο χρήστης να ορίσει τους νευρώνες ανά κρυμμένο επίπεδο. Έστω εδώ επιλέγουμε 1 κρυμμένο επίπεδα και τρεις νευρώνες ανά κρυμμένο επίπεδο.



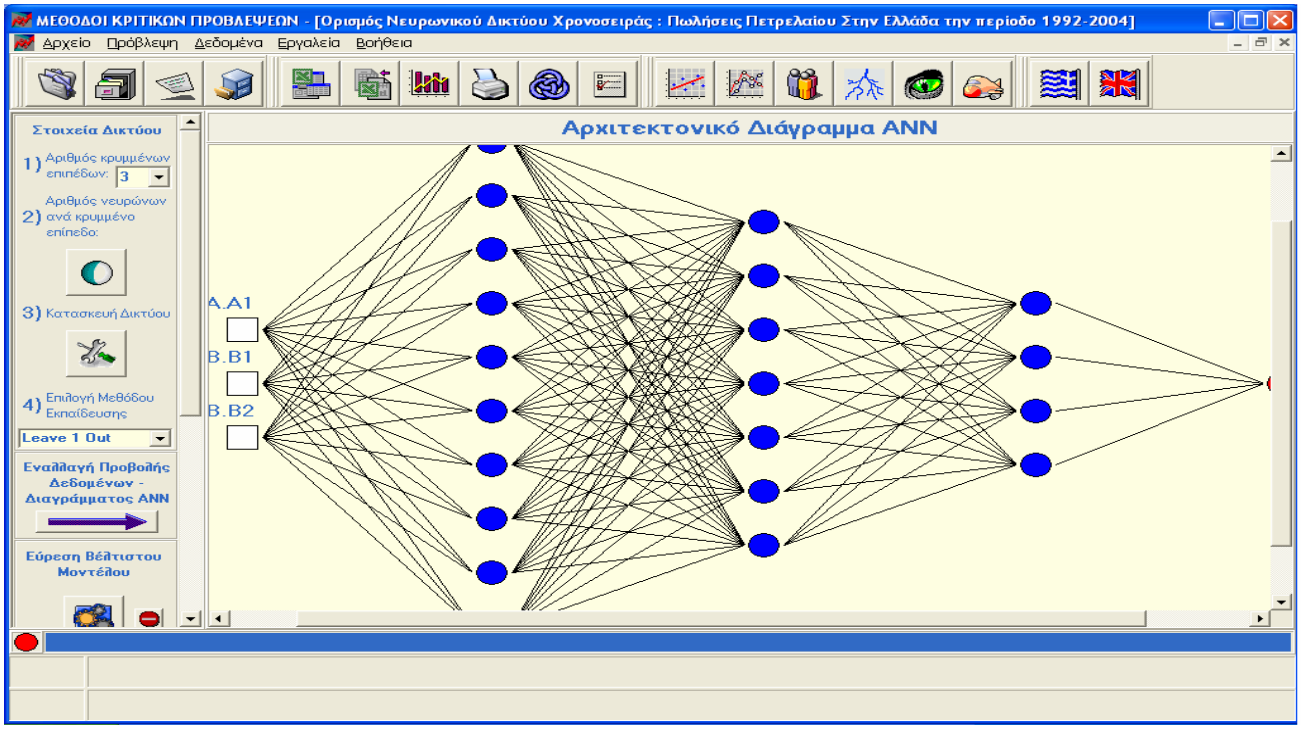

Στη συνέχεια πατώντας το κουμπί της επιβεβαίωσης και έπειτα το κουμπί  ο χρήστης ,μπορεί να κατασκευάσει το επιλεγμένο δίκτυο. Η μορφή του φαίνεται παρακάτω.



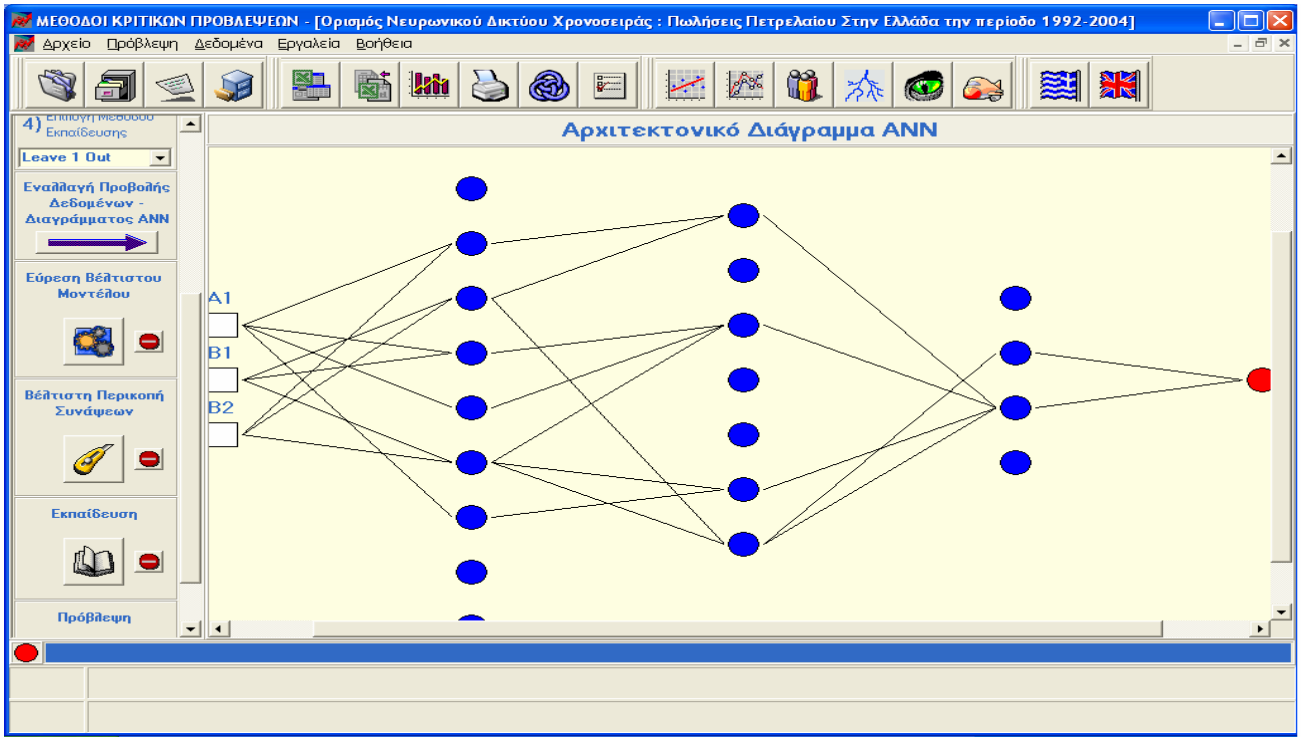
Στη συνέχεια ο χρήστης μπορεί να επιλέξει τη μέθοδο εκπαίδευσης που θέλει να εφαρμόσει. Σύμφωνα και με το κεφάλαιο 3 η καταλληλότερη μέθοδος για την περίπτωσή μας είναι η Leave One Out. Η εκπαίδευση του δικτύου γίνεται σε κάποιο τοπικό ελάχιστο πατώντας το κουμπί.  Τέλος η πρόβλεψη γίνεται πατώντας το κουμπί.  Στη συγκεκριμένη περίπτωση πήραμε για το πρώτο υποκειμενικό γεγονός 97.257 με αναμενόμενη τιμή 104+N(0,20) ενώ για το δεύτερο υποκειμενικό γεγονός πήραμε την τιμή 135.310 με αναμενόμενη τιμή την 134+N(0,20). Τα αποτελέσματα μπορούμε να δούμε πως είναι τα καλύτερα και από τις τρεις μεθόδους. Αυτό είναι φυσικό αφού τα ANN μπόρεσαν και προσομοίωσαν τη μη γραμμική συνάρτηση της επίδρασης με τις παραμέτρους των Υποκειμενικών Παραγόντων.

Η επιλογή του μοντέλου πρόβλεψης μπορεί να γίνει και με αυτόματο τρόπο από την ίδια την εφαρμογή. Η λειτουργία αυτή γίνεται με το πάτημα του κουμπιού.  Η εφαρμογή εκπαιδεύει όλα τα δίκτυα μέχρι δύο κρυμμένα επίπεδα και μέχρι Ν αριθμό νευρώνων ανά κρυμμένο επίπεδο όπου Ν το μέγεθος του διανύσματος εισόδου. Βέβαια θα πρέπει να σημειωθεί ότι διαφορετικά αποτελέσματα είναι δυνατόν να προκύψουν κάθε φορά αφού η εκπαίδευση του δικτύου γίνεται κάθε φορά σε κάποιο τοπικό και όχι ολικό ελάχιστο. Έτσι είναι δυνατόν να επιλεγεί κάποιο δίκτυο με μικρότερο ολικό ελάχιστο από κάποιο άλλο αν τύχει και κατά την εκπαίδευσή του συγκλίνει σε κάποιο τοπικό ελάχιστο μικρότερο από το δεύτερο.

Έστω τώρα φτιάχνουμε ένα πολύπλοκο ANN με τρία κρυμμένα επίπεδα, 10 νευρώνες στο πρώτο κρυμμένο επίπεδο, 7 στο δεύτερο και 4 στο τρίτο. Εκπαιδεύουμε αρχικά το δίκτυο σε κάποιο τοπικό ελάχιστο και στη συνέχεια εφαρμόζουμε τον αλγόριθμο Optimal Brain Surgeon πατώντας το κουμπί.  Τα αποτελέσματα φαίνονται στην επόμενη σελίδα:

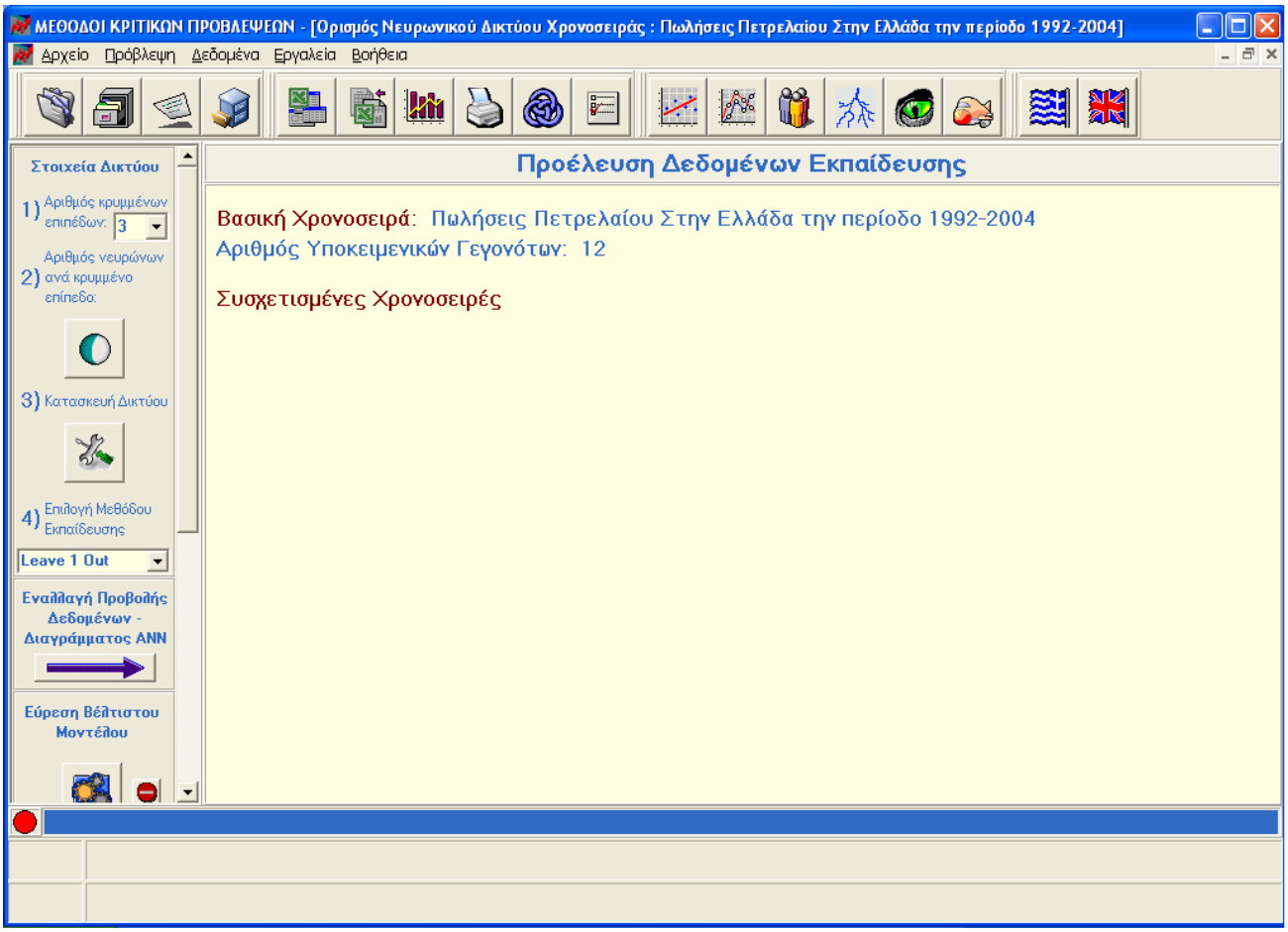



Μετά την περικοπή συνάψεων:



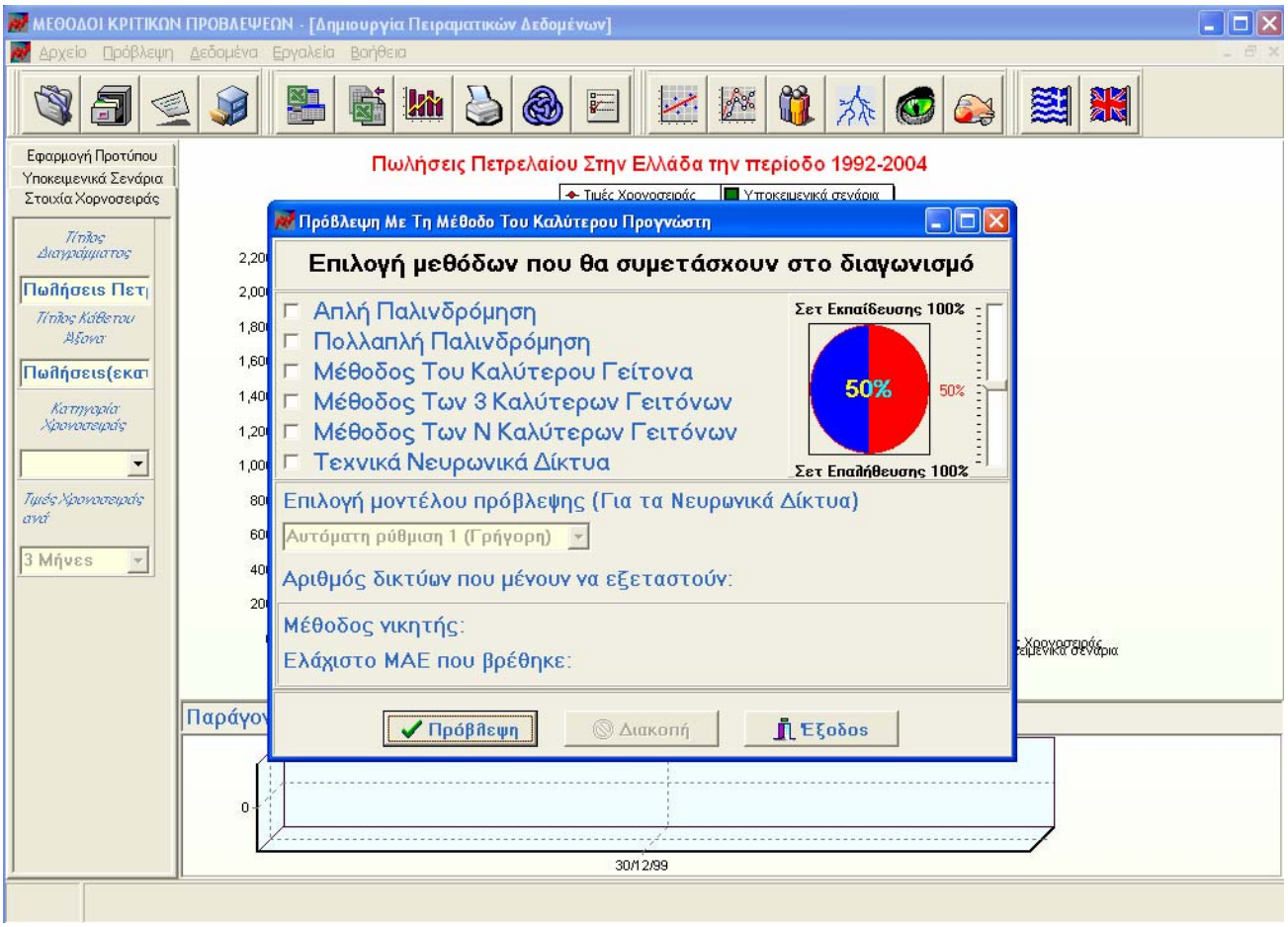
Παρατηρούμε μια δραματική μείωση στον αριθμό των συνάψεων του δικτύου. Βέβαια στη δικιά μας περίπτωση δεν έχει ιδιαίτερο νόημα να επιλέξουμε ένα τόσο πολύπλοκο δίκτυο αφού πολύ μικρότερα από αυτό μπορούν και προσεγγίζουν αρκετά ικανοποιητικά τη μη γραμμική συνάρτηση. Τώρα μπορούμε να εκπαιδεύσουμε το νέο δίκτυο και να προβλέψουμε με αυτό.

Τέλος, μπορούμε να δούμε και την προέλευση των δεδομένων εκπαίδευσης πατώντας πάνω στο πλήκτρο  . Τα αποτελέσματα φαίνονται στην οθόνη ως εξής:




Στη δικιά μας περίπτωση δεν υπάρχουν συσχετισμένες χρονοσειρές.

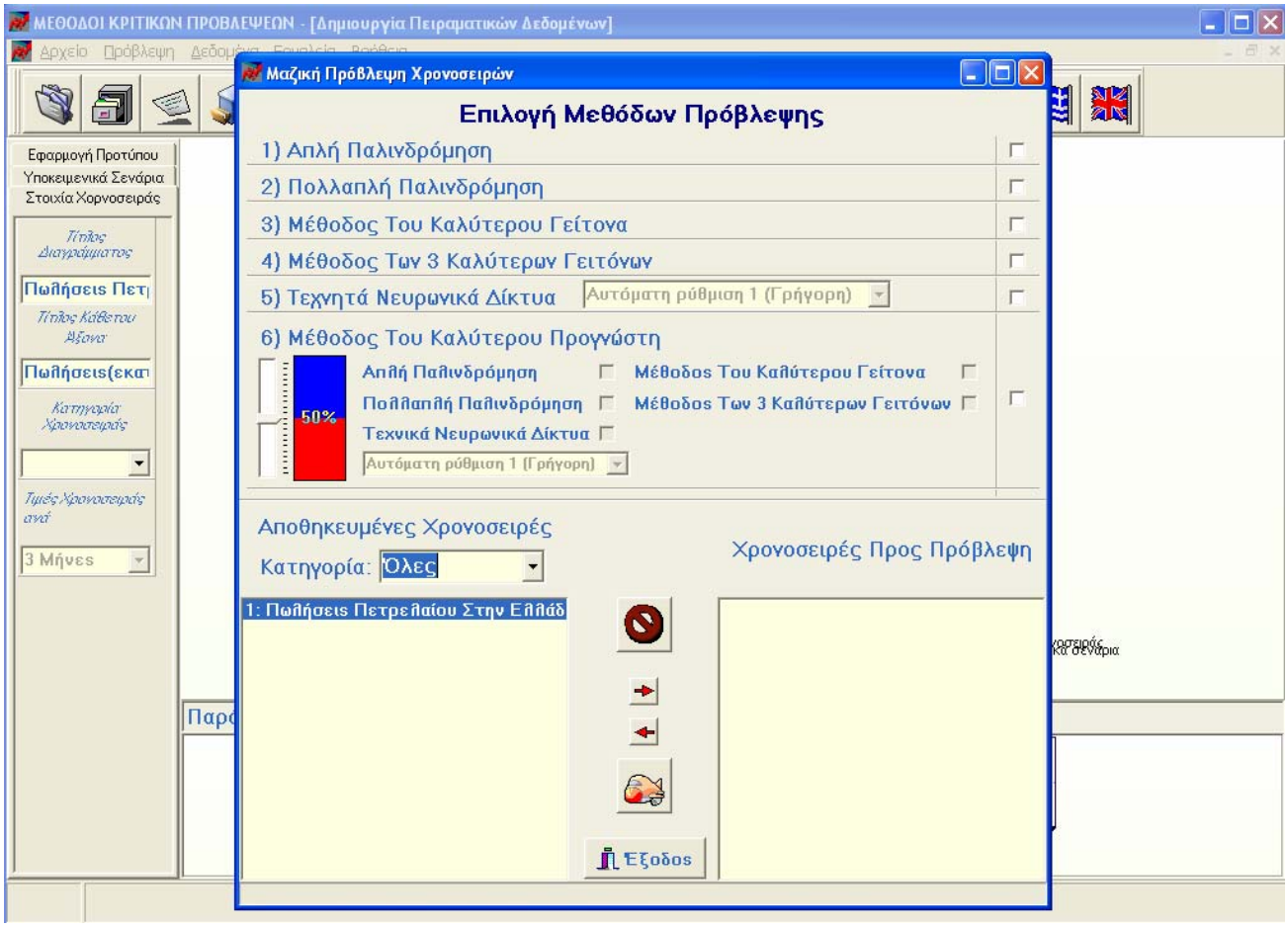
**Ο Καλύτερος Προγνώστης.** Πρόκειται για μία συγκριτική μέθοδο η οποία χρησιμοποιεί ουσιαστικά τις 4 προηγούμενες μεθόδους. Η μέθοδος διαχωρίζει το σύνολο εκπαίδευσης των δεδομένων Γ σε δύο υποσύνολα. Ένα σύνολο εκπαίδευσης Α και ένα σύνολο επαλήθευσης Β με $A \cap B = \varnothing$ και $A \cup B = \Gamma$. Το ποσοστό διαχωρισμού ορίζεται από το χρήστη. Κάθε μέθοδος που εξετάσαμε μέχρι τώρα εκπαιδεύεται με τη σειρά στο σύνολο Α και επαληθεύεται στο σύνολο Β. Για κάθε μέθοδο υπολογίζεται το μέσο απόλυτο σφάλμα και τελικά χρησιμοποιείται αυτή η οποία στο σετ επαλήθευσης Β παρουσιάζει το μικρότερο σφάλμα. Αυτή προβλέπει την επίδραση των μελλοντικών υποκειμενικών γεγονότων. Η μέθοδος του καλύτερου προγνώστη εφαρμόζεται όταν ο χρήστης πατήσει στο κουμπί  ή την κατάλληλη επιλογή από το βασικό menu οπότε και εμφανίζεται η παρακάτω φόρμα.




Ο χρήστης επιλέγει ποια μέθοδος θα συμμετάσχει στο διαγωνισμό σημειώνοντας τα αντίστοιχα checkbox στα αριστερά. Το ποσοστό του σετ επαλήθευσης και του σετ εκπαίδευσης από το αρχικό σετ

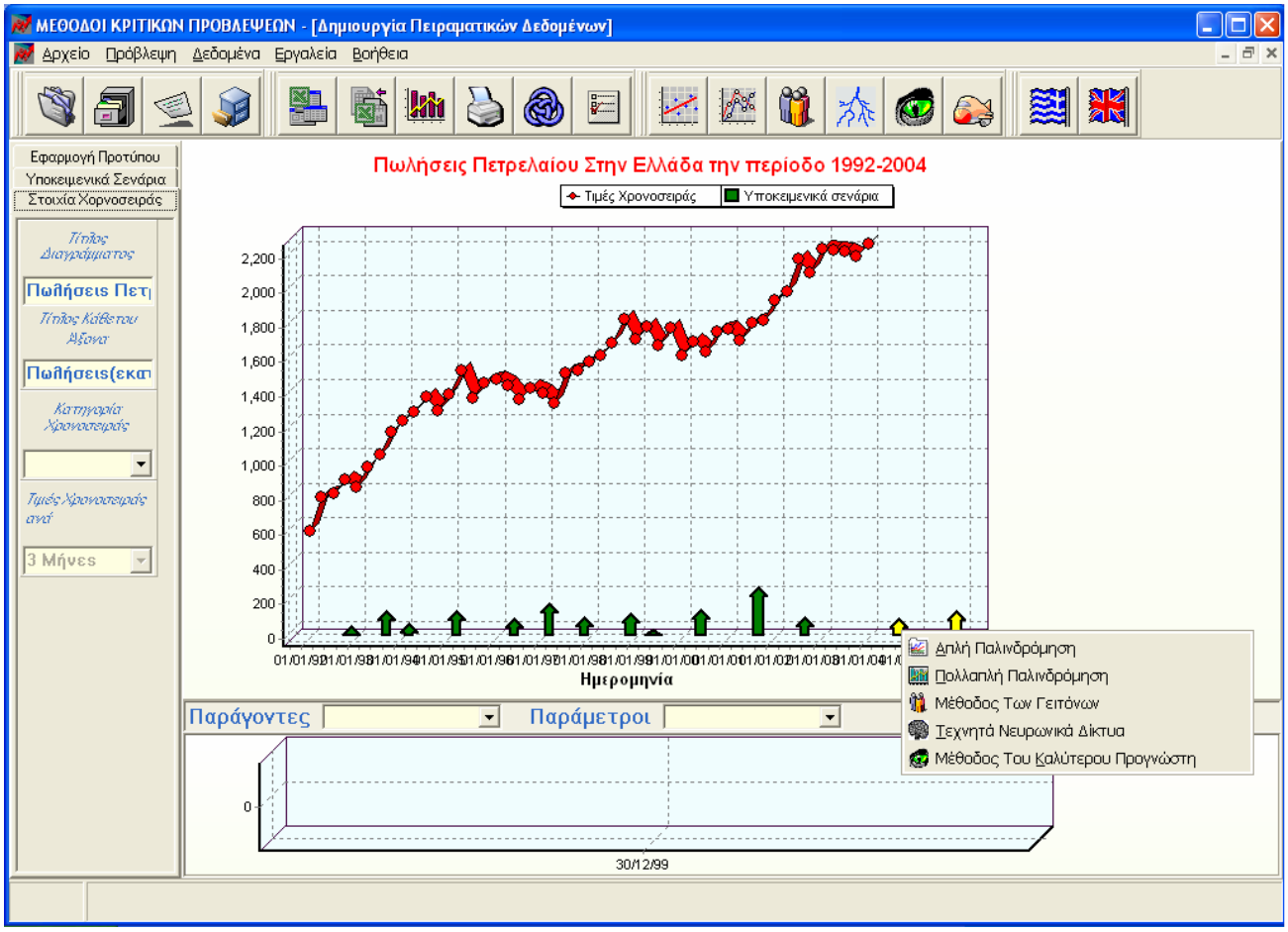
ιστορικών δεδομένων επιλέγεται μετακινώντας την μπάρα στα δεξιά της φόρμας. Π.χ. όταν ο δείκτης βρίσκεται στο 90% αυτό σημαίνει ότι 90% του σετ δεδομένων θα αποδοθεί στο σετ εκπαίδευσης και το υπόλοιπο 10% στο σετ επαλήθευσης. Κάθε φορά που εκτελείται η πρόβλεψη και γίνεται ο διαχωρισμός των δύο υποσυνόλων τα δεδομένα επιλέγονται για το κάθε υποσύνολο με τυχαία σειρά. Τα ακραία ποσοστά 100% και 0% μπορούν να επιλεγούν από το χρήστη χωρίς όμως να έχουν πραγματικό αντίκρισμα αφού το λιγότερο 1 δεδομένο θα ανήκει και στα δύο υποσύνολα προκειμένου να εφαρμοστεί η μέθοδος.. Για την εκπαίδευση των νευρωνικών δικτύων παρέχεται η δυνατότητα επιλογής μιας από τις τρεις αυτόματες μεθόδους. Στο τέλος της διαδικασίας εμφανίζεται η μέθοδος που κέρδισε το διαγωνισμό, το μέσο απόλυτο σφάλμα αυτής στο σετ εκπαίδευσης και εκτελείται η πρόβλεψη.

**Μαζική Πρόβλεψη Χρονοσειρών.** Η εφαρμογή παρέχει στο χρήστη μια επιπλέον δυνατότητα να χρησιμοποιήσει και τις 5 μεθόδους προβλέποντας μαζικά την επίδραση μελλοντικών υποκειμενικών γεγονότων πολλών χρονοσειρών μαζί. Είτε πατώντας το κουμπί  είτε από το βασικό menu την αντίστοιχη επιλογή εμφανίζεται η φόρμα που φαίνεται στην επόμενη σελίδα. Ο χρήστης επιλέγει ποιες μεθόδους θέλει να εφαρμόσει, με ποιες παραμέτρους και σε ποιες χρονοσειρές. Τα αποτελέσματα αποθηκεύονται τελικά σε ένα αρχείο excel με μια συγκεκριμένη μορφή την οποία θα εξετάσουμε σε επόμενη ενότητα. Παράλληλα με τα αποτελέσματα της πρόβλεψης κάθε μεθόδου υπολογίζονται και τα σφάλματα MAPE MdAPE και GR-RAE για κάθε μέθοδο ξεχωριστά και για κάθε χρονοσειρά με βάση την τιμή της επίδρασης των μελλοντικών υποκειμενικών γεγονότων πριν την πρόβλεψη.




Η εφαρμογή κάθε μιας από τις 5 μεθόδους μπορεί να γίνει και ξεχωριστά για κάθε μελλοντικό υποκειμενικό γεγονός κάνοντας δεξί click

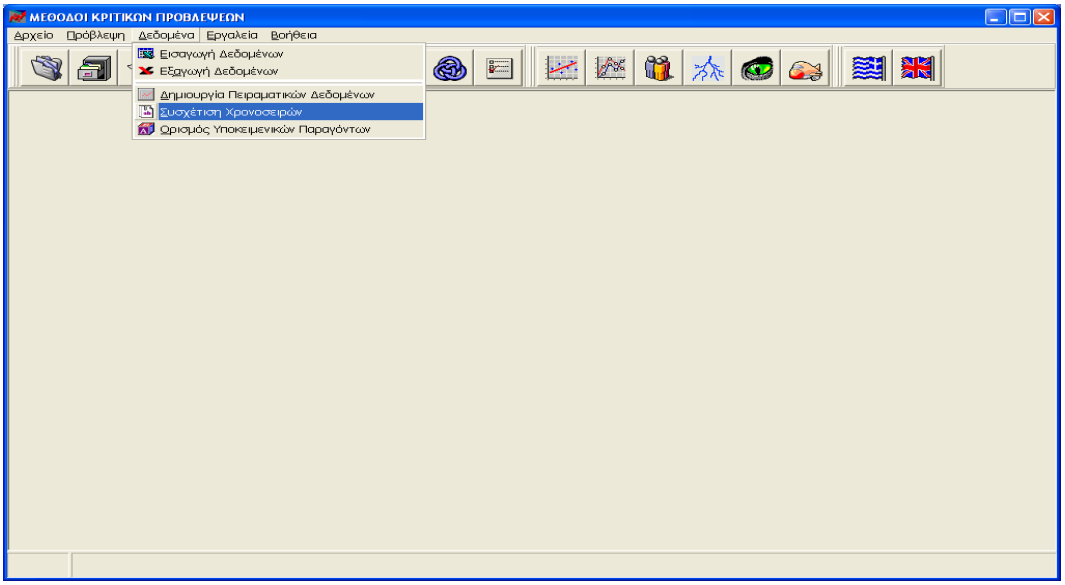
πάνω στην αντίστοιχη μπάρα όπως φαίνεται στο screenshot της επόμενης σελίδας.


ΜΕΘΟΔΟΙ ΚΡΙΤΙΚΩΝ ΠΡΟΒΛΕΨΕΩΝ - [Δημιουργία Πειραματικών Δεδομένων]
Αρχείο Πρόβλεψη Δεδομένα Εργαλεία Βοήθεια
Εφαρμογή Προτύπου
Υποκειμενικά Σενάρια
Στοιχία Χορνοσειράς
Πώλήσεις Πετ
Πώλήσεις(εκα
3 Μήνες
Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004
Τιμές Χρονοσειράς
Υποκειμενικά σενάρια
Ημερομηνία
Παράγοντες
Παράμετροι
Απλή Παλινδρόμηση
Πολλαπλή Παλινδρόμηση
Μέθοδος Των Γειτόνων
Τεχνητά Νευρωνικά Δίκτυα
Μέθοδος Του Καλύτερου Προγνώστη
30/12/99

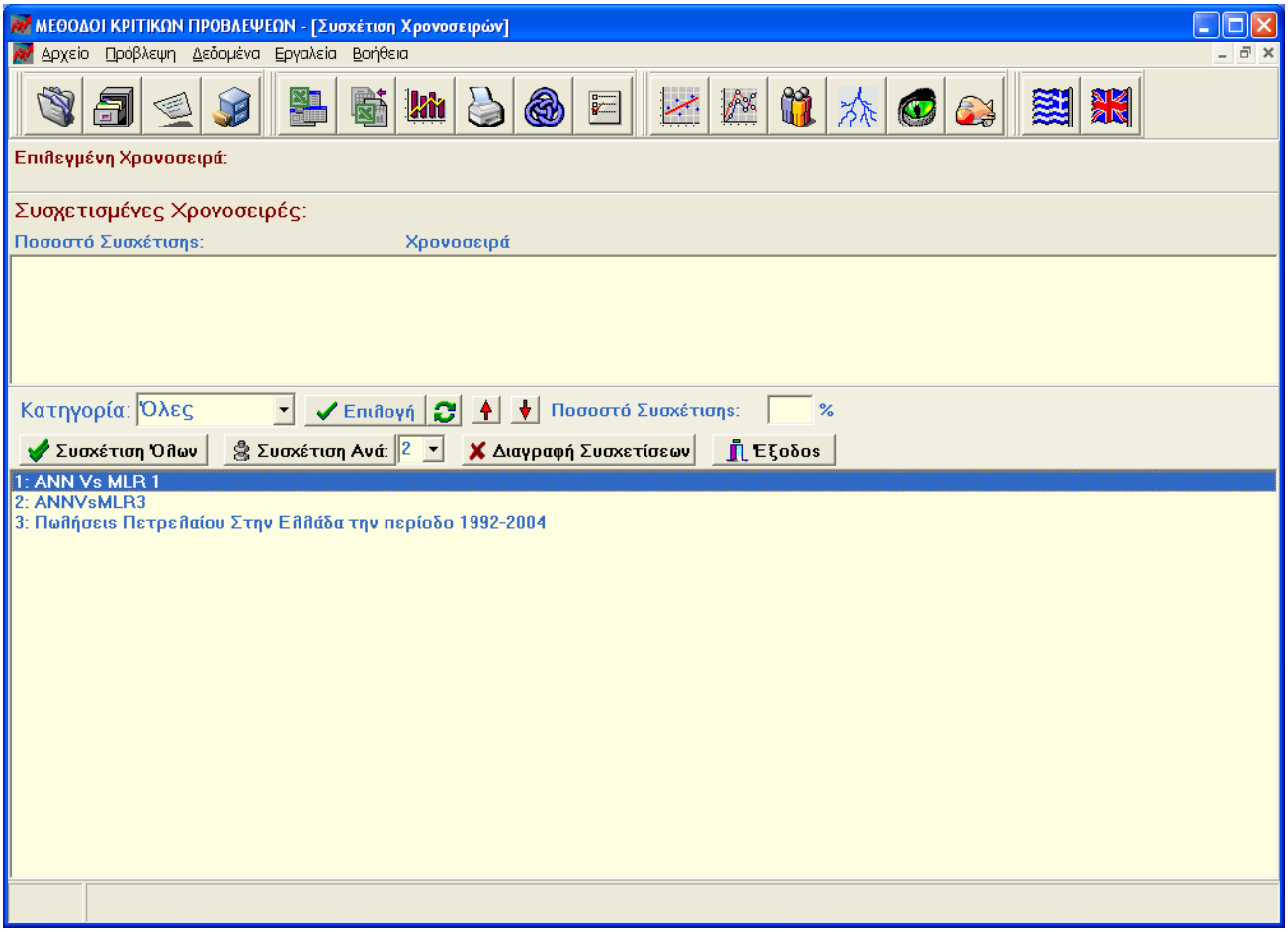
### 4.3.4 Συσχέτιση Χρονοσειρών

Μία πολύ χρήσιμη λειτουργία για την πρόβλεψη με χρήση Τεχνικών Νευρωνικών Δικτύων είναι και η συσχέτιση χρονοσειρών όμοιων ως προς τη συμπεριφορά αντίδρασης σε παρόμοια ή ίδια υποκειμενικά γεγονότα. Η χρήση Τεχνικών Νευρωνικών Δικτύων προϋποθέτει την ύπαρξη αρκετών ιστορικών δεδομένων προκειμένου να μπορέσει το δίκτυο να προσεγγίσει σωστά και σε μεγάλο εύρος την συνάρτηση εξάρτησης της επίδρασης από τις παραμέτρους. Αν τα δεδομένα δεν είναι αρκετά μπορεί να παραπλανηθεί και να προσεγγίσει μεν τα ιστορικά δεδομένα με μεγάλη ακρίβεια αλλά με μια τελείως διαφορετική μη γραμμική συνάρτηση. Η συσχέτιση των χρονοσειρών επιτρέπει τη χρήση ιστορικών δεδομένων από πολλές χρονοσειρές που παρουσιάζουν ομοιότητες, για την πρόβλεψη μελλοντικών γεγονότων μιας χρονοσειράς. Έτσι αυξάνει σημαντικά το πλήθος των δεδομένων εκπαίδευσης του ANN και βελτιώνει κατά πολύ την ποιότητα των αποτελεσμάτων.

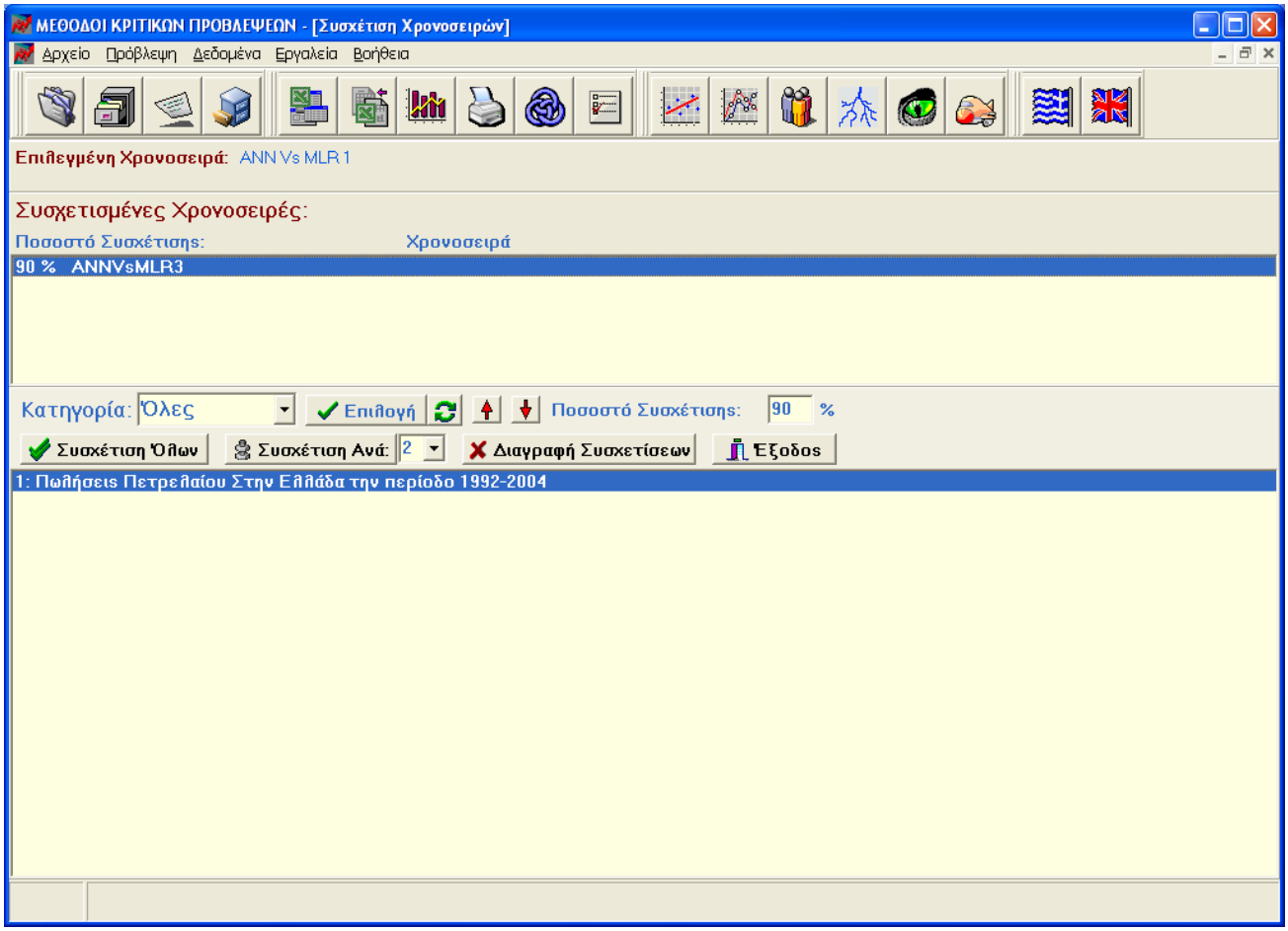
Ο χρήστης της εφαρμογής είναι αποκλειστικά υπεύθυνος για τη συσχέτιση των χρονοσειρών και γι' αυτό θα πρέπει να προσέξει οι συσχετίσεις που κάνει να ανταποκρίνονται στην πραγματικότητα. Λανθασμένες συσχετίσεις είναι πολύ πιθανό να οδηγήσουνε σε μεγάλα λάθη πρόβλεψης. Ενδεικτικά παρακάτω θα δείξουμε πως μπορεί να γίνει η συσχέτιση μερικών χρονοσειρών χωρίς όμως αυτές να σχετίζονται πραγματικά. Ο χρήστης θα πρέπει να πατήσει το κουμπί  ή να επιλέξει την ίδια λειτουργία από το βασικό menu όπως φαίνεται παρακάτω.



Τότε εμφανίζεται η φόρμα της συσχέτισης χρονοσειρών:



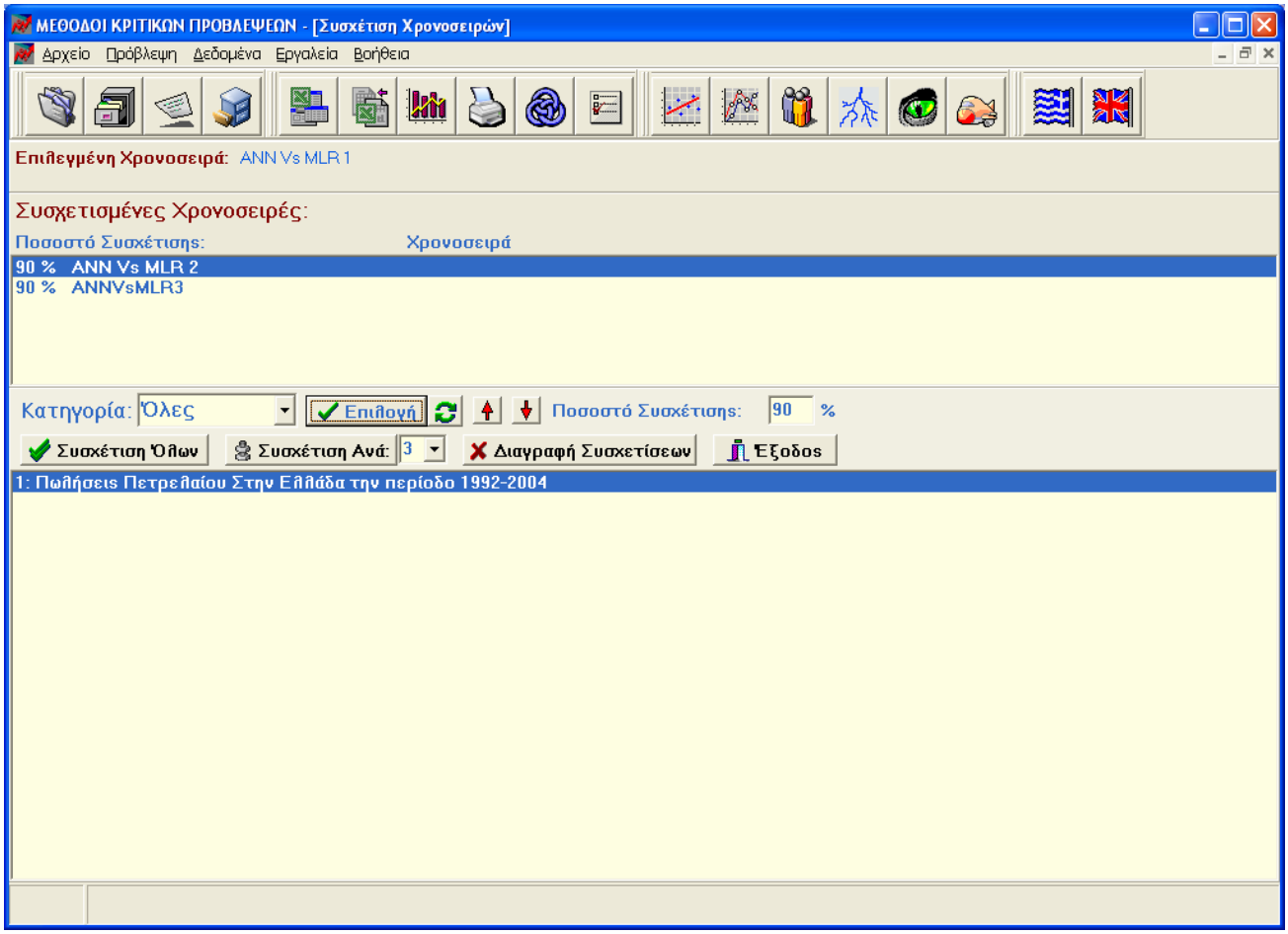

Στη συγκεκριμένη περίπτωση έχουμε αποθηκευμένες στη ΒΔ τρεις μόνο χρονοσειρές τα ονόματα των οποίων φαίνονται στο παραπάνω screenshot. Από το drop down box με τίτλο Κατηγορία μπορούμε να επιλέξουμε να φαίνονται στη λίστα οι χρονοσειρές μόνο μιας συγκεκριμένης κατηγορίας. Έστω θεωρούμε ότι οι χρονοσειρές 1 και 2 σχετίζονται σε ένα βαθμό 90% και ότι οι 2 και 3 με ένα βαθμό 85%. Για να ο δηλώσουμε αυτό στη ΒΔ ακολουθούμε τα εξής βήματα. Επιλέγουμε αρχικά τη χρονοσειρά 1 όπως φαίνεται παραπάνω. Έπειτα, πατάμε στο κουμπί με τίτλο επιλογή. Η χρονοσειρά τότε επιλέγεται και μπορούμε να τη συσχετίσουμε με άλλες. Επιλέγουμε στη συνέχεια τη χρονοσειρά 2 και στο κουτί με το ποσοστό συσχέτιση πληκτρολογούμε 90. Τέλος πατάμε το κουμπί  και η συσχέτιση δηλώνεται. Το αποτέλεσμα φαίνεται στην επόμενη σελίδα.




Για να επανέλθουμε στην αρχική κατάσταση πατάμε το κουμπί της αναίρεσης επιλογής και ακολουθούμε την ίδια διαδικασία για τη δεύτερη συσχέτιση. Το κουμπί χρησιμεύει στην διαγραφή μιας συσχέτισης. Αυτό μεταφέρει τις επιλεγμένες χρονοσειρές από τη λίστα συσχέτισης στη λίστα των χρονοσειρών.

Η φόρμα συσχετίσεων παρέχει ορισμένες ακόμα δυνατότητες. Πατώντας το πλήκτρο με τίτλο 'Διαγραφή Συσχετίσεων' ο χρήστης μπορεί να διαγράψει όλες τις συσχετίσεις που είναι αποθηκευμένες στη ΒΔ. Με το κουμπί 'Συσχέτιση Όλων' συσχετίζονται με βάση το δηλωμένο ποσοστό συσχέτισης όλες οι επιλεγμένες χρονοσειρές στη λίστα χρονοσειρών μεταξύ τους. Με το κουμπί 'Συσχέτιση Ανά' συσχετίζονται μεταξύ τους όλες οι επιλεγμένες χρονοσειρές στη λίστα χρονοσειρών ανά τον αριθμό που επιλέγεται από το αντίστοιχο drop down box δίπλα από το κουμπί. Αν για παράδειγμα έχουν επιλεγεί 10 χρονοσειρές και θέλουμε να τις συσχετίσουμε μεταξύ τους ανά 6 τότε οι 6 πρώτες θα συσχετιστούν κανονικά και οι υπόλοιπες 4 αναγκαστικά ανά 4 μεταξύ τους. Στο παρακάτω παράδειγμα έχουμε εισάγει στη ΒΔ μία

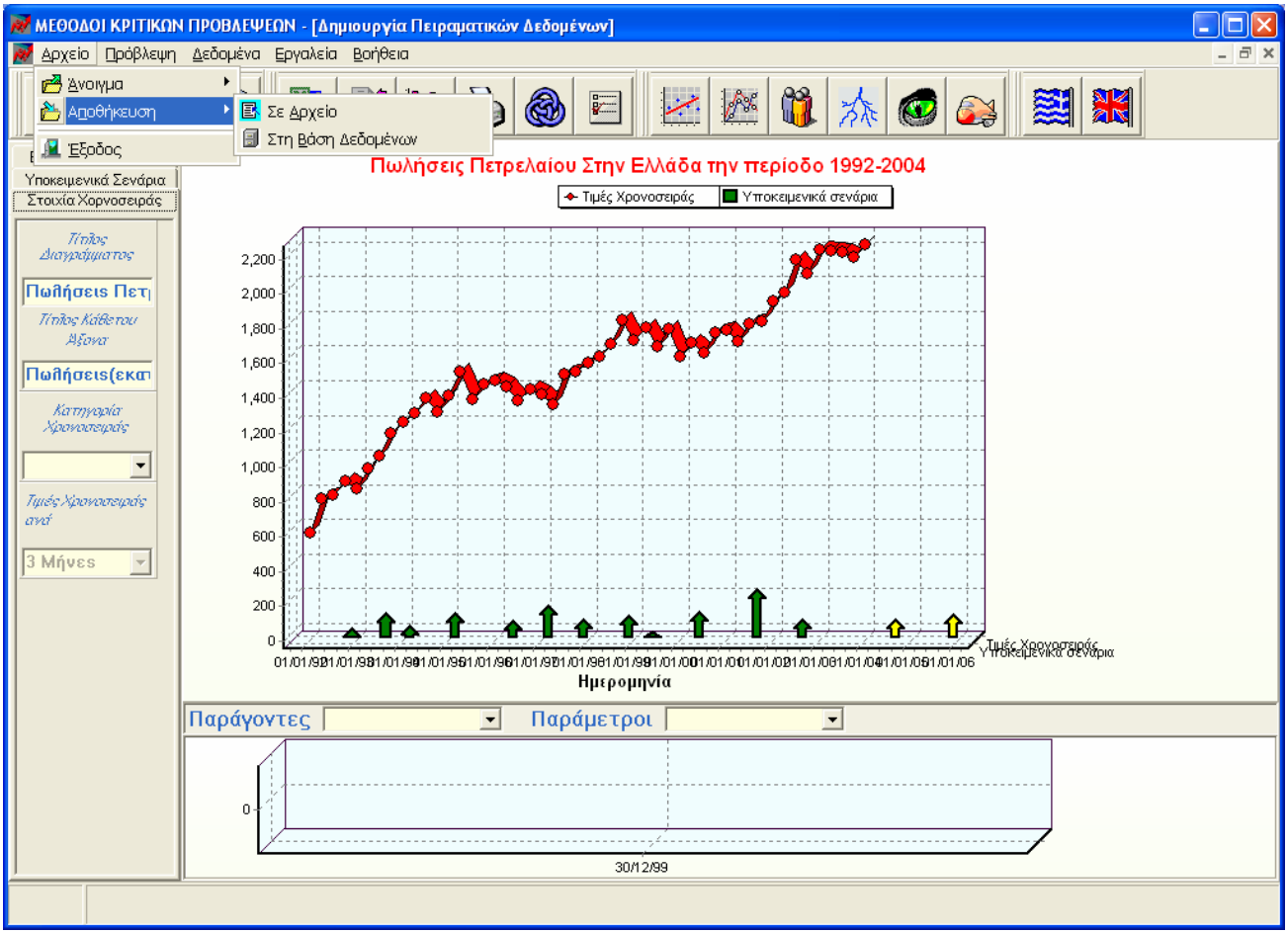
ακόμα χρονοσειρά και έχουμε διαγράψει όλες τις συσχετίσεις από τη ΒΔ. Θέλουμε να συσχετίσουμε όλες τις χρονοσειρές στη ΒΔ μεταξύ τους ανά 3 με ποσοστό συσχέτισης έστω 90%. Για το λόγο αυτό επιλέγουμε όλες τις χρονοσειρές (4 στον αριθμό) από τη λίστα χρονοσειρών και από το drop down box τον αριθμό 3. Τέλος πατάμε το κουμπί 'Συσχέτιση Ανά'. Αφού η εφαρμογή μας ενημερώσει ότι η συσχέτιση εκτελέστηκε με επιτυχία για να δούμε το αποτέλεσμα επιλέγουμε έστω την πρώτη χρονοσειρά και πατάμε το κουμπί επιλογή. Το αποτέλεσμα που παίρνουμε φαίνεται παρακάτω.



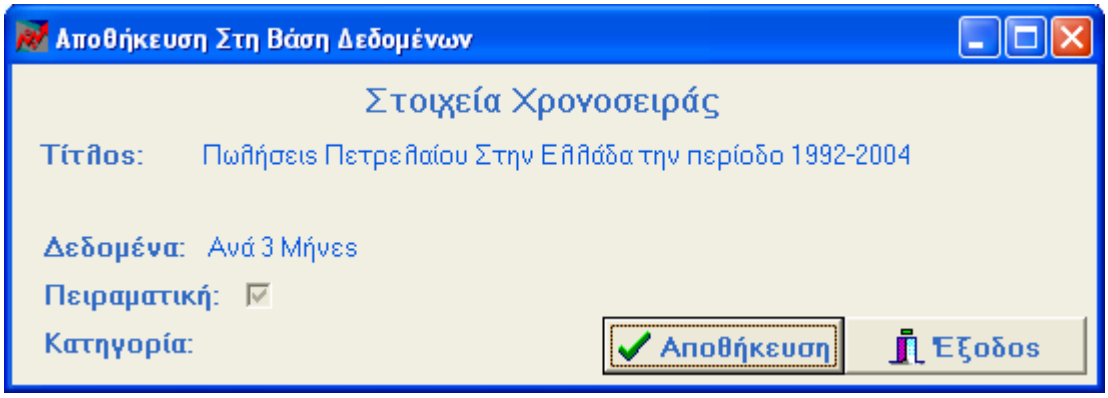

Παρατηρούμε πως η πρώτη χρονοσειρά συσχετίστηκε με τις επόμενες δύο. Το ίδιο θα παρατηρήσουμε αν επιλέξουμε τις χρονοσειρές δύο και τρία ενώ για την τέταρτη θα δούμε πως δεν έχει συσχετιστεί με καμία αφού περίσσεψε τελευταία.

### 4.3.5 Διαχείριση Χρονοσειρών

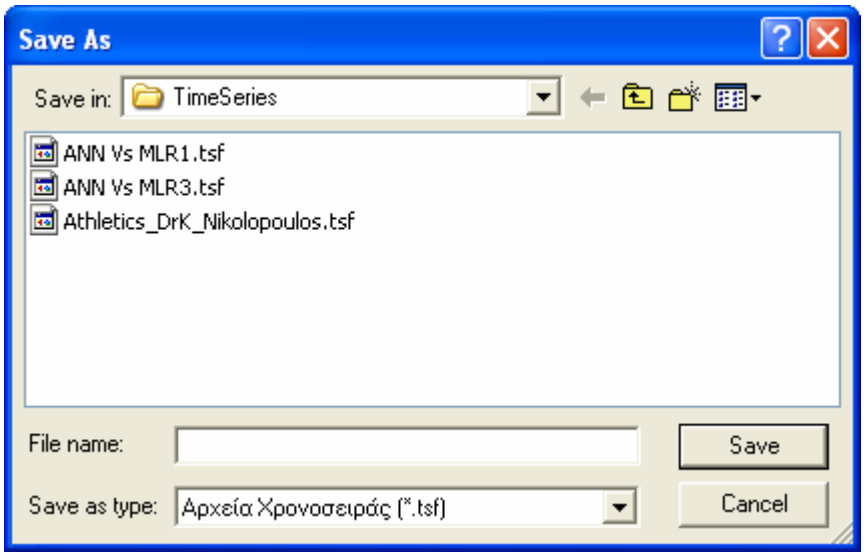
Η διαχείριση των χρονοσειρών αφορά στην αποθήκευσή τους στη ΒΔ ή σε κάποιο αρχείο ή την ανάκτησή τους από τη ΒΔ ή από κάποιο αρχείο. Η αποθήκευση μιας χρονοσειράς γίνεται είτε από το βασικό menu όπως φαίνεται παρακάτω είτε από τα κουμπιά:  'Αποθήκευση στη ΒΔ'  και 'Αποθήκευση σε Αρχείο'.



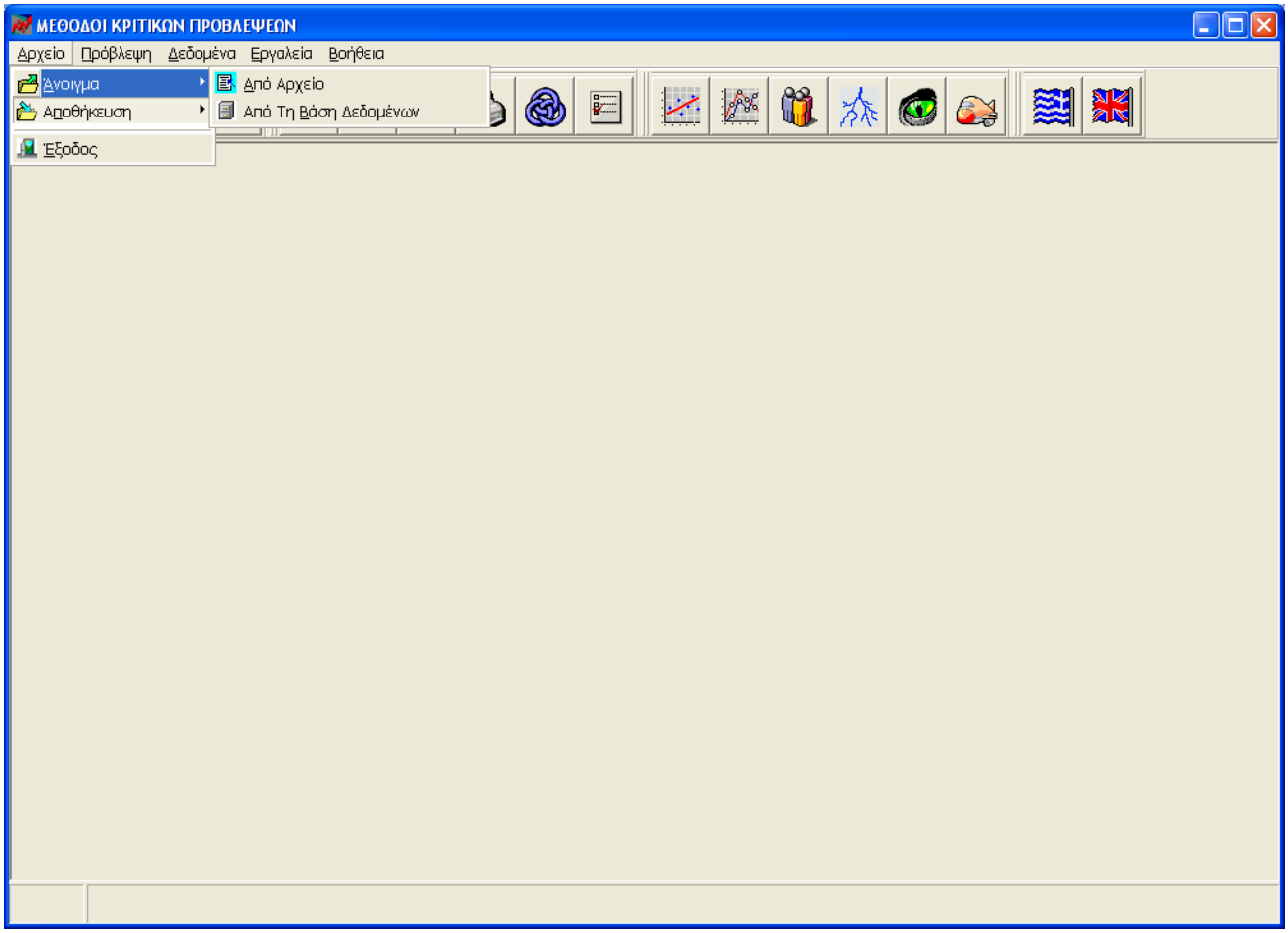

Κατά την επιλογή της αποθήκευσης μια χρονοσειράς στη ΒΔ εμφανίζεται η φόρμα επιβεβαίωσης που φαίνεται στην επόμενη σελίδα και περιέχει βασικές πληροφορίες για τη χρονοσειρά.



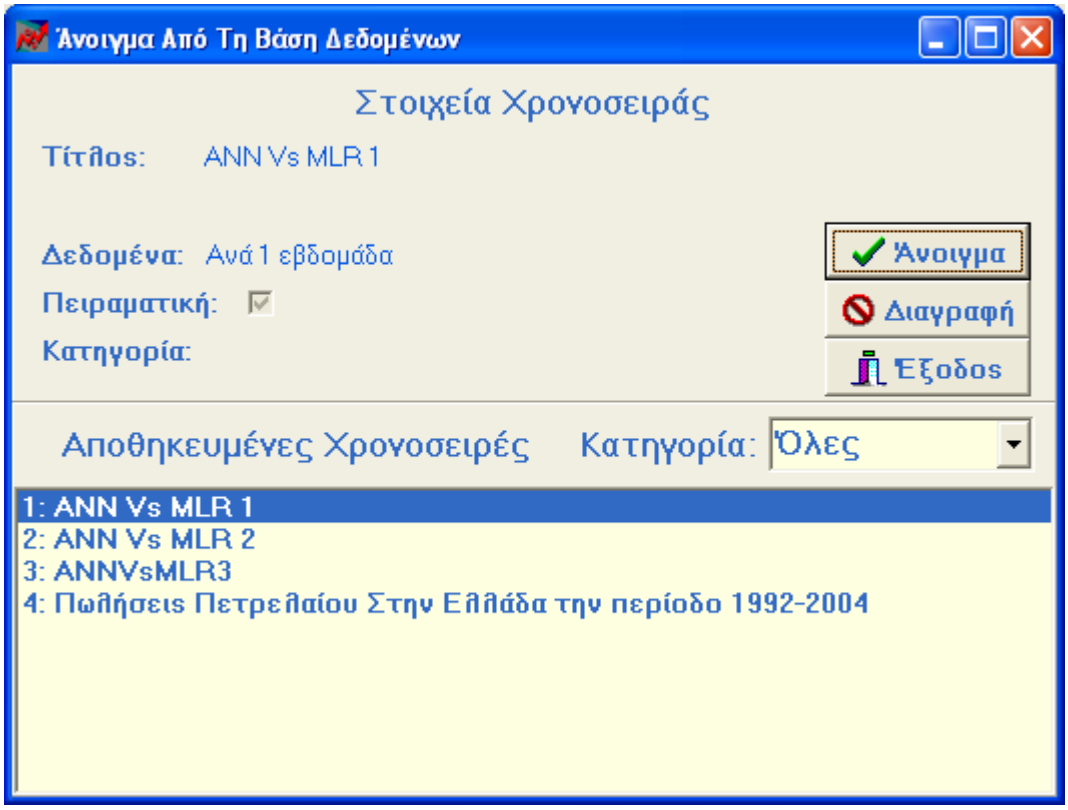

Κατά την επιλογή της αποθήκευσης σε κάποιο αρχείο εμφανίζεται η παρακάτω φόρμα στην οποία ο χρήστης δηλώνει το όνομα της χρονοσειράς με το οποίο θέλει να αποθηκευτεί. Το αρχείο που θα δημιουργήσει η εφαρμογή στο δίσκο είναι δυαδικό και μπορεί να διαβαστεί μόνο από την ίδια την εφαρμογή. Αυτό έχει κατάληξη tsf.



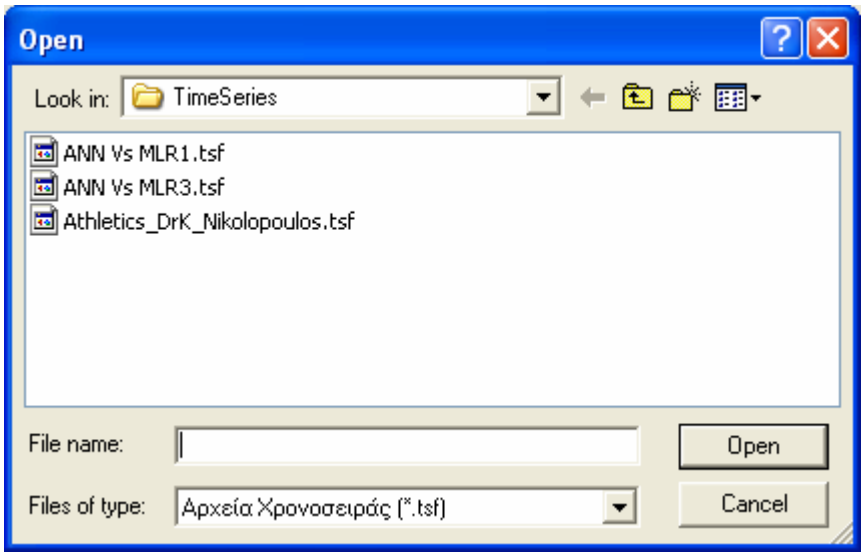

Η λειτουργία της ανάκτησης μιας χρονοσειράς γίνεται είτε από τη ΒΔ είτε από το δίσκο από κάποιο αρχείο της μορφής που περιγράφτηκε προηγουμένως. Αυτή γίνεται από το βασικό menu όπως φαίνεται στην επόμενη σελίδα ή από τα κουμπιά:  'Άνοιγμα από τη ΒΔ' και  'Άνοιγμα από Αρχείο'




Κατά την επιλογή της ανάκτησης από τη ΒΔ εμφανίζεται η φόρμα επιλογής της επόμενης σελίδας μέσα από την οποία μπορούμε να ανοίξουμε ταυτόχρονα πολλές χρονοσειρές καθώς και να διαγράψουμε όποιες επιθυμούμε από τη ΒΔ. Από το drop down box μπορούμε να επιλέξουμε να εμφανίζονται στη λίστα μόνο οι χρονοσειρές κάποιας συγκεκριμένης κατηγορίας. Η πολλαπλή επιλογή είναι δυνατή είτε κρατώντας πατημένο το ctrl και επιλέγοντας τις χρονοσειρές είτε κάνοντας drag το mouse από πάνω και κρατώντας πατημένο το αριστερό κουμπί του.

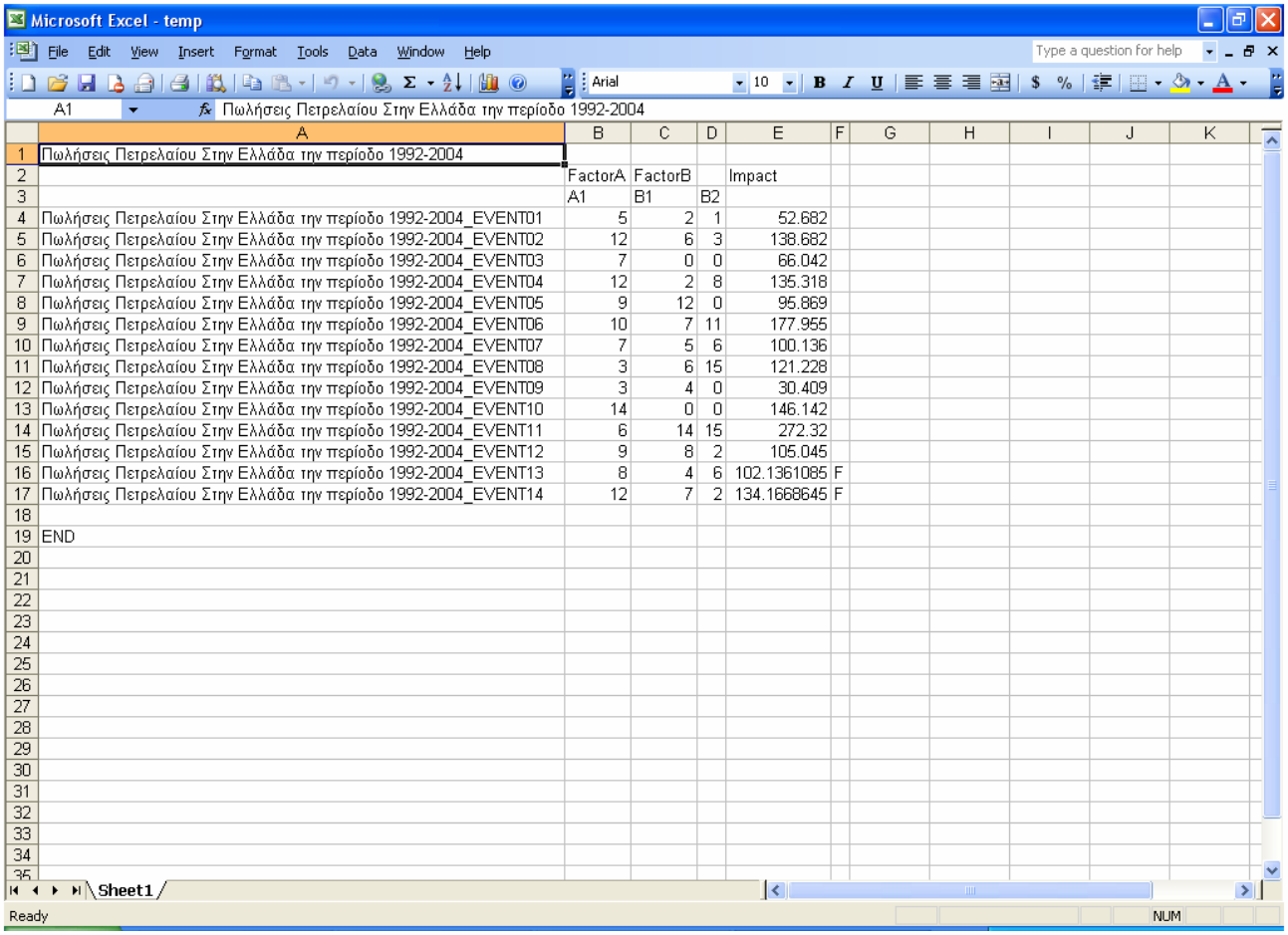



Κατά την επιλογή της ανάκτησης από αρχείο εμφανίζεται η παρακάτω φόρμα η οποία μας επιτρέπει να επιλέξουμε το κατάλληλο αρχείο από το δίσκο.



### 4.3.6 Εισαγωγή – Εξαγωγή Δεδομένων

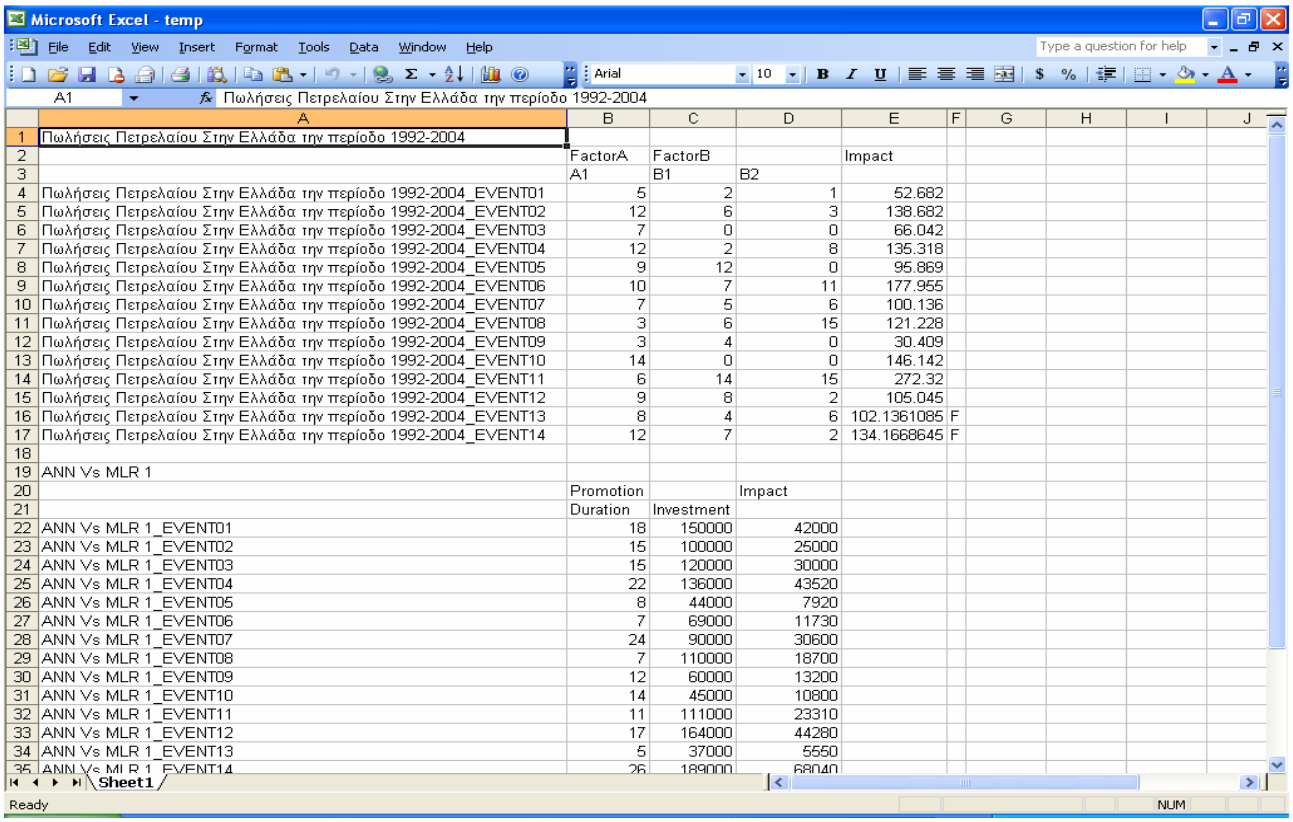
**Εισαγωγή – Εξαγωγή Δεδομένων.** Η Λειτουργία αυτή περιλαμβάνει την εισαγωγή και εξαγωγή χρονοσειρών από και σε αρχεία του excel. Μια χρονοσειρά για να μπορεί να εισαχθεί από αρχείο του excel θα πρέπει να έχει συγκεκριμένη μορφή όπως φαίνεται παρακάτω.




| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004 | | | | | |
| 2 | | FactorA | FactorB | | Impact | |
| 3 | | A1 | B1 | B2 | | |
| 4 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT01 | 5 | 2 | 1 | 52.682 | |
| 5 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT02 | 12 | 6 | 3 | 138.682 | |
| 6 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT03 | 7 | 0 | 0 | 66.042 | |
| 7 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT04 | 12 | 2 | 8 | 135.318 | |
| 8 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT05 | 9 | 12 | 0 | 95.869 | |
| 9 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT06 | 10 | 7 | 11 | 177.955 | |
| 10 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT07 | 7 | 5 | 6 | 100.136 | |
| 11 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT08 | 3 | 6 | 15 | 121.228 | |
| 12 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT09 | 3 | 4 | 0 | 30.409 | |
| 13 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT10 | 14 | 0 | 0 | 146.142 | |
| 14 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT11 | 6 | 14 | 15 | 272.32 | |
| 15 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT12 | 9 | 8 | 2 | 105.045 | |
| 16 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT13 | 8 | 4 | 6 | 102.1361085 | F |
| 17 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT14 | 12 | 7 | 2 | 134.1668645 | F |
| 18 | | | | | | |
| 19 | END | | | | | |

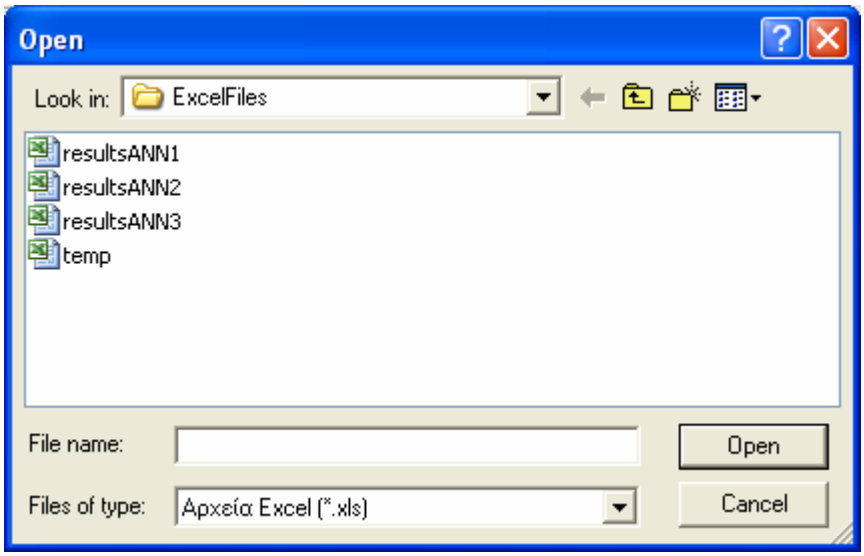
Στο πάνω αριστερό κελί δηλώνεται η ονομασία της χρονοσειράς (A1). Στο κελί B2 αρχίζουμε να δηλώνουμε τους παράγοντες με τις παραμέτρους τους που εμφανίζονται στα ιστορικά και μελλοντικά υποκειμενικά γεγονότα. Για κάθε παράγοντα γράφουμε ακριβώς από κάτω του, τις παραμέτρους από τις οποίες αυτός αποτελείται με τη σειρά. Στο κελί που βρίσκεται ακριβώς κάτω από κάθε παράγοντα δηλώνεται η πρώτη παράμετρός του. Όταν δηλωθούνε όλες οι παράμετροι όλων των παραγόντων στο

αμέσως δεξιότερο κελί και στο ύψος που είναι δηλωμένοι οι παράγοντες πρέπει να υπάρχει η λέξη 'Impact'. Η στήλη αυτή περιέχει την επίδραση των υποκειμενικών γεγονότων. Αν πρόκειται για πραγματικά δεδομένα και δε ξέρουμε την επίδραση των μελλοντικών γεγονότων τότε θα πρέπει να συμπληρωθεί το 0 στην επίδραση κάθε γεγονότος και να μην αφεθεί κενό το αντίστοιχο κελί. Ακριβώς δεξιότερα της επίδρασης των μελλοντικών γεγονότων για να δηλωθεί ότι αυτά είναι μελλοντικά θα πρέπει να υπάρχει το σύμβολο 'F'. Στα κελιά που ακολουθούνε κάτω από τις ονομασίες των παραμέτρων συμπληρώνεται η τιμή της κάθε παραμέτρου για το αντίστοιχο γεγονός. Κάθε γεγονός χαρακτηρίζεται από μια ονομασία που αρχίζει με το όνομα της χρονοσειράς και καταλήγει με τη λέξη 'Event' και τον αριθμό του γεγονότος. Είναι δυνατόν ένα αρχείο excel να περιέχει περισσότερες από μία χρονοσειρές, αρκεί αυτές να διαχωρίζονται από τουλάχιστον μία κενή γραμμή όπως φαίνεται παρακάτω. Σε κάθε περίπτωση θα πρέπει μετά την τελευταία χρονοσειρά και μετά από μια τουλάχιστον κενή γραμμή στην πρώτη στήλη να υπάρχει η λέξη 'END'. Τέλος το όνομα του sheet στο οποίο δηλώνεται μια χρονοσειρά αποτελεί και την ονομασία της κατηγορίας στην οποία ανήκει. Μπορεί ένα αρχείο excel να περιέχει περισσότερα του ενός sheets με περισσότερες από μία χρονοσειρές το καθένα.

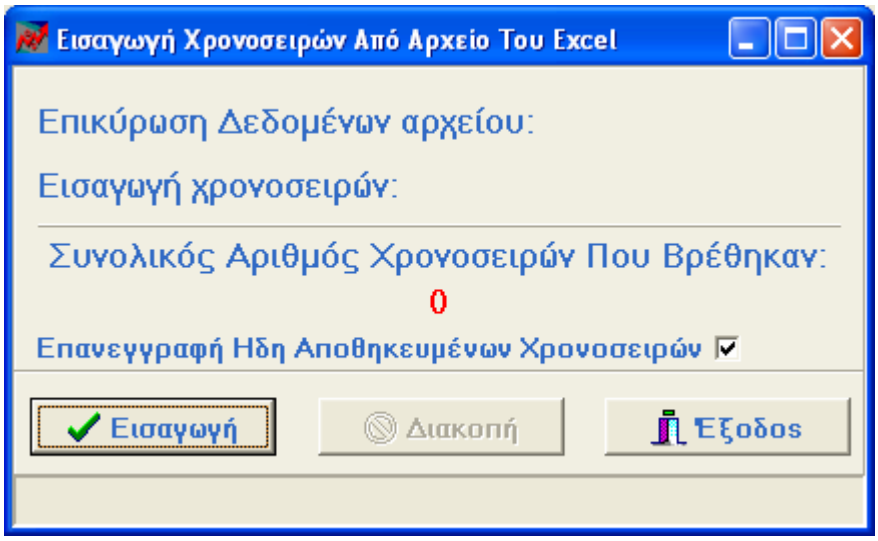



| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004 | | | | | | | | | |
| 2 | | FactorA | FactorB | | Impact | | | | | |
| 3 | | A1 | B1 | B2 | | | | | | |
| 4 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT01 | 5 | 2 | 1 | 52.682 | | | | | |
| 5 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT02 | 12 | 6 | 3 | 138.682 | | | | | |
| 6 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT03 | 7 | 0 | 0 | 66.042 | | | | | |
| 7 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT04 | 12 | 2 | 8 | 135.318 | | | | | |
| 8 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT05 | 9 | 12 | 0 | 95.869 | | | | | |
| 9 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT06 | 10 | 7 | 11 | 177.955 | | | | | |
| 10 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT07 | 7 | 5 | 6 | 100.136 | | | | | |
| 11 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT08 | 3 | 6 | 15 | 121.228 | | | | | |
| 12 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT09 | 3 | 4 | 0 | 30.409 | | | | | |
| 13 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT10 | 14 | 0 | 0 | 146.142 | | | | | |
| 14 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT11 | 6 | 14 | 15 | 272.32 | | | | | |
| 15 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT12 | 9 | 8 | 2 | 105.045 | | | | | |
| 16 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT13 | 8 | 4 | 6 | 102.1361085 | F | | | | |
| 17 | Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004_EVENT14 | 12 | 7 | 2 | 134.1668645 | F | | | | |
| 18 | | | | | | | | | | |
| 19 | ANN Vs MLR 1 | | | | | | | | | |
| 20 | | Promotion | | Impact | | | | | | |
| 21 | | Duration | Investment | | | | | | | |
| 22 | ANN Vs MLR 1_EVENT01 | 18 | 150000 | 42000 | | | | | | |
| 23 | ANN Vs MLR 1_EVENT02 | 15 | 100000 | 25000 | | | | | | |
| 24 | ANN Vs MLR 1_EVENT03 | 15 | 120000 | 30000 | | | | | | |
| 25 | ANN Vs MLR 1_EVENT04 | 22 | 136000 | 43520 | | | | | | |
| 26 | ANN Vs MLR 1_EVENT05 | 8 | 44000 | 7920 | | | | | | |
| 27 | ANN Vs MLR 1_EVENT06 | 7 | 69000 | 11730 | | | | | | |
| 28 | ANN Vs MLR 1_EVENT07 | 24 | 90000 | 30600 | | | | | | |
| 29 | ANN Vs MLR 1_EVENT08 | 7 | 110000 | 18700 | | | | | | |
| 30 | ANN Vs MLR 1_EVENT09 | 12 | 60000 | 13200 | | | | | | |
| 31 | ANN Vs MLR 1_EVENT10 | 14 | 45000 | 10800 | | | | | | |
| 32 | ANN Vs MLR 1_EVENT11 | 11 | 111000 | 23310 | | | | | | |
| 33 | ANN Vs MLR 1_EVENT12 | 17 | 164000 | 44280 | | | | | | |
| 34 | ANN Vs MLR 1_EVENT13 | 5 | 37000 | 5550 | | | | | | |
| 35 | ANN Vs MLR 1_EVENT14 | 26 | 189000 | 68040 | | | | | | |

Η εισαγωγή δεδομένων από ένα αρχείο excel γίνεται όταν ο χρήστης πατήσει το κουμπί  ή την αντίστοιχη επιλογή από το βασικό menu. Τότε εμφανίζεται η φόρμα επιλογής αρχείου:

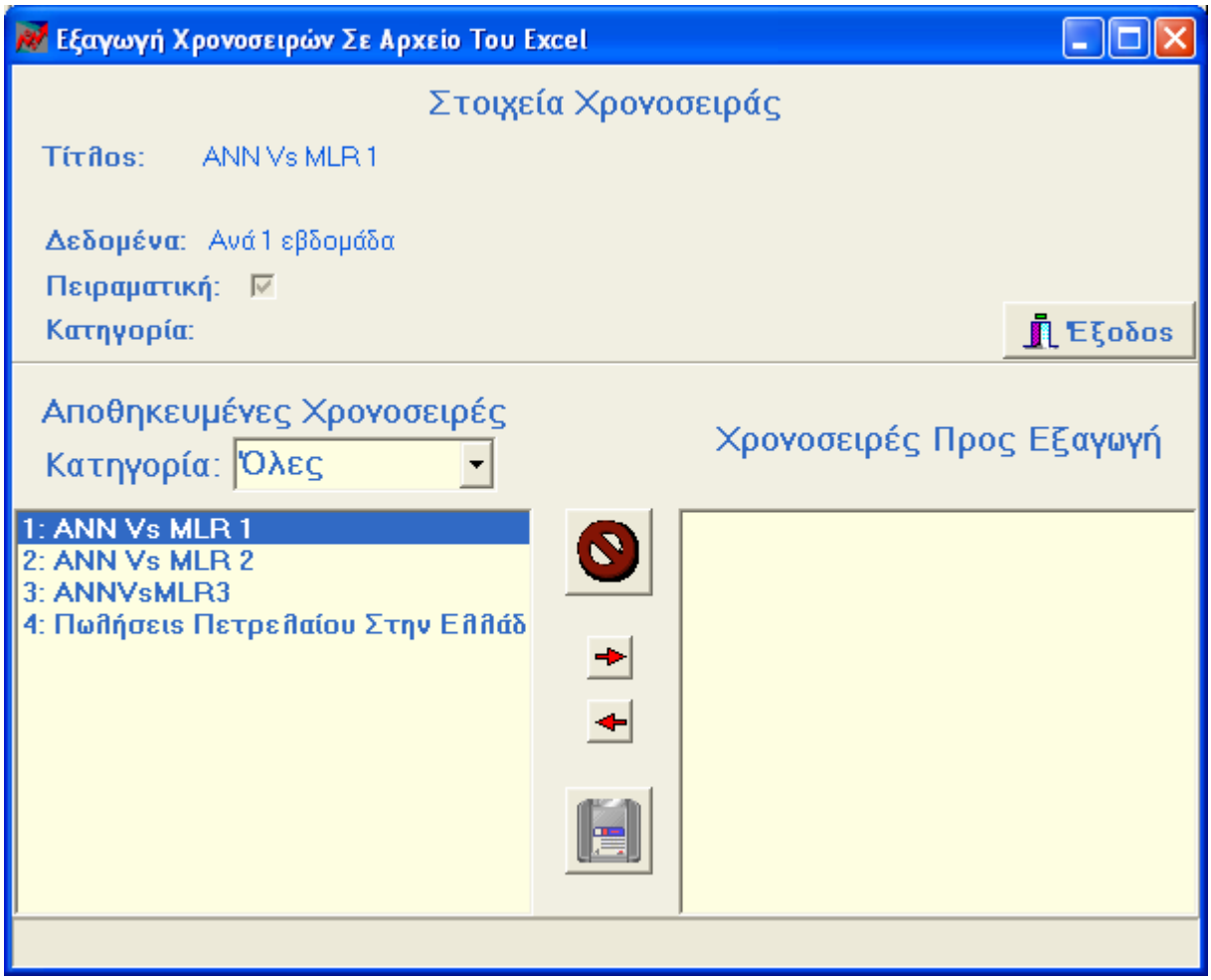



και αμέσως μετά η φόρμα εισαγωγής:

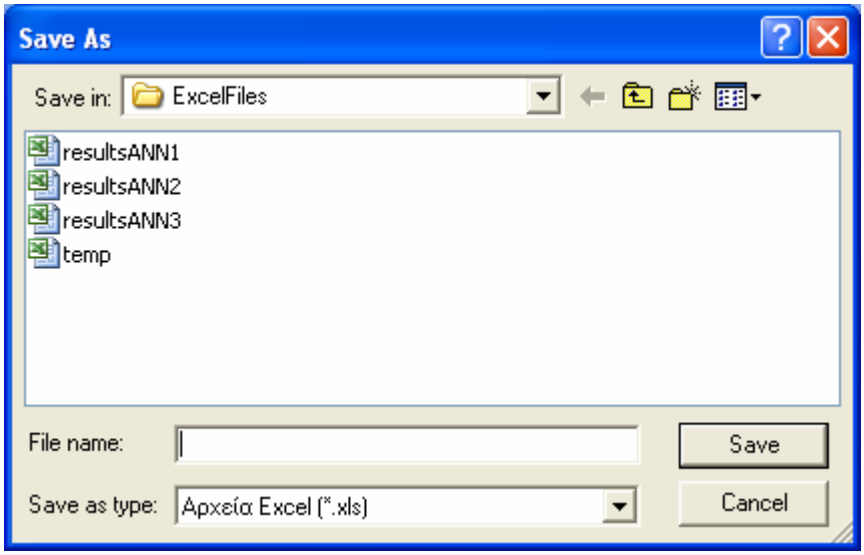



Πριν την εισαγωγή γίνεται επαλήθευση των δεδομένων του αρχείου και αν υπάρχει οποιοδήποτε πρόβλημα μη συμβατότητας εμφανίζεται το κατάλληλο μήνυμα ειδοποιώντας σε ποιο ακριβώς κελί εντοπίστηκε το πρόβλημα. Στην αντίθετη περίπτωση ολοκληρώνεται η εισαγωγή στη ΒΔ.

Η εξαγωγή των δεδομένων γίνεται όταν ο χρήστης πατήσει το κουμπί  ή από το βασικό menu. Τότε εμφανίζεται η φόρμα που φαίνεται παρακάτω.



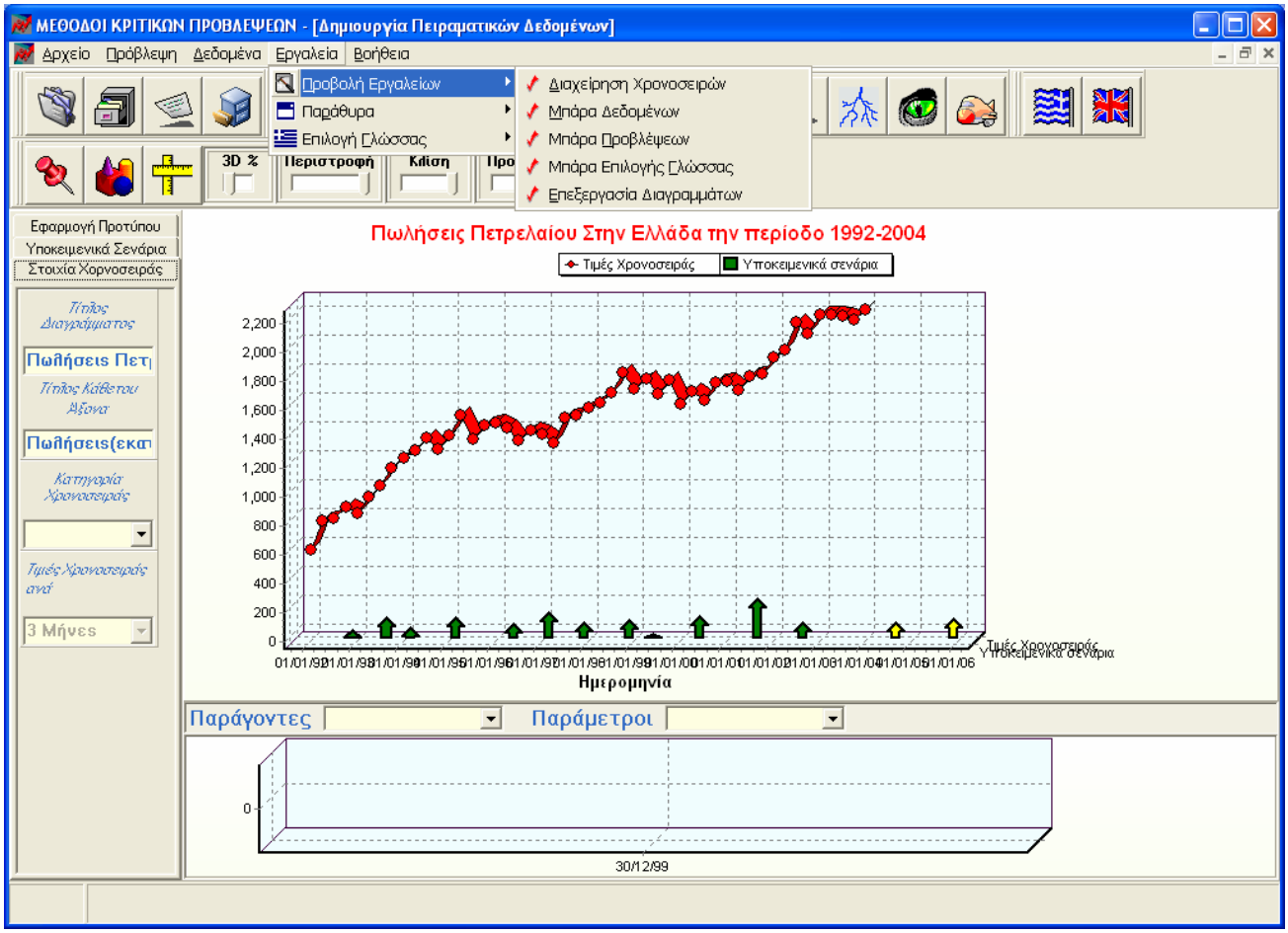

Ο χρήστης θα πρέπει να επιλέξει ποιες χρονοσειρές θέλει να εξάγει και στη συνέχεια να πατήσει το κουμπί με τη δισκέτα. Τότε θα εμφανιστεί η φόρμα της επόμενης σελίδας στην οποία θα δηλώσει το όνομα του αρχείου και την τοποθεσία στην οποία θέλει να αποθηκευτεί.




Το αρχείο που θα δημιουργηθεί θα ακολουθεί όλους τους παραπάνω κανόνες και θα περιέχει ομαδοποιημένες τις χρονοσειρές σε διαφορετικά sheets ανάλογα με την κατηγορία στην οποία αυτές ανήκουν. Κάθε sheet θα ονομαστεί από την αντίστοιχη κατηγορία.

### 4.3.7 Επιπλέον Δυνατότητες

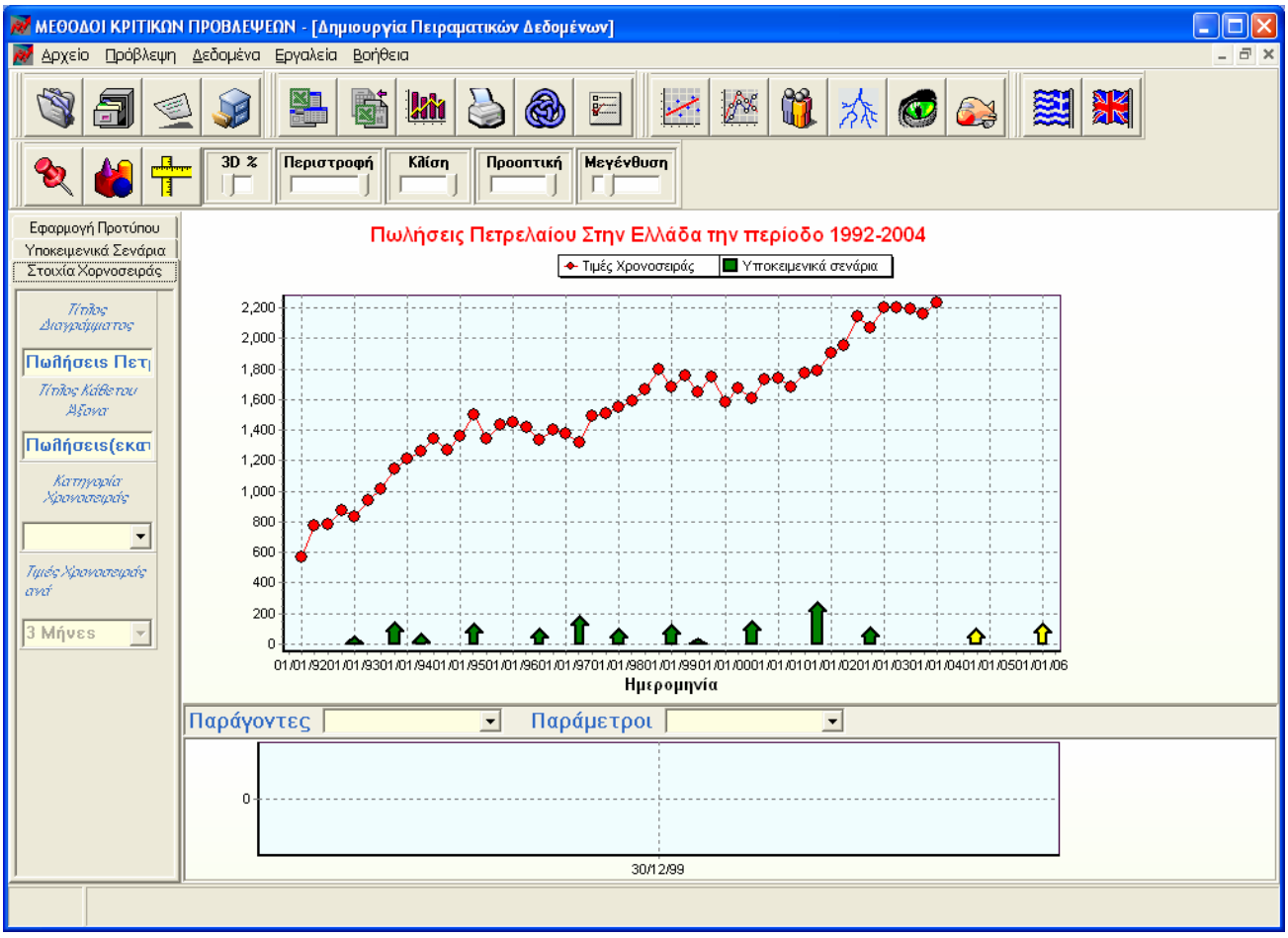
**Επεξεργασία Διαγραμμάτων.** Μέσα από το βασικό menu Εργαλεία->Προβολή Εργαλείων ο χρήστης έχει τη δυνατότητα να επιλέξει ποια εργαλεία θα εμφανίζονται στην οθόνη. Η συγκεκριμένη επιλογή φαίνεται στο παρακάτω screenshot. Μέχρι εδώ έχουμε επεξηγήσει όλες τις μπάρες εργαλείων εκτός από μία. Αυτή της επεξεργασίας διαγραμμάτων. Η τελευταία δίνει στον χρήστη κάποιες επιπλέον δυνατότητες όσον αφορά την προβολή των χρονοσειρών.




Αυτές είναι:

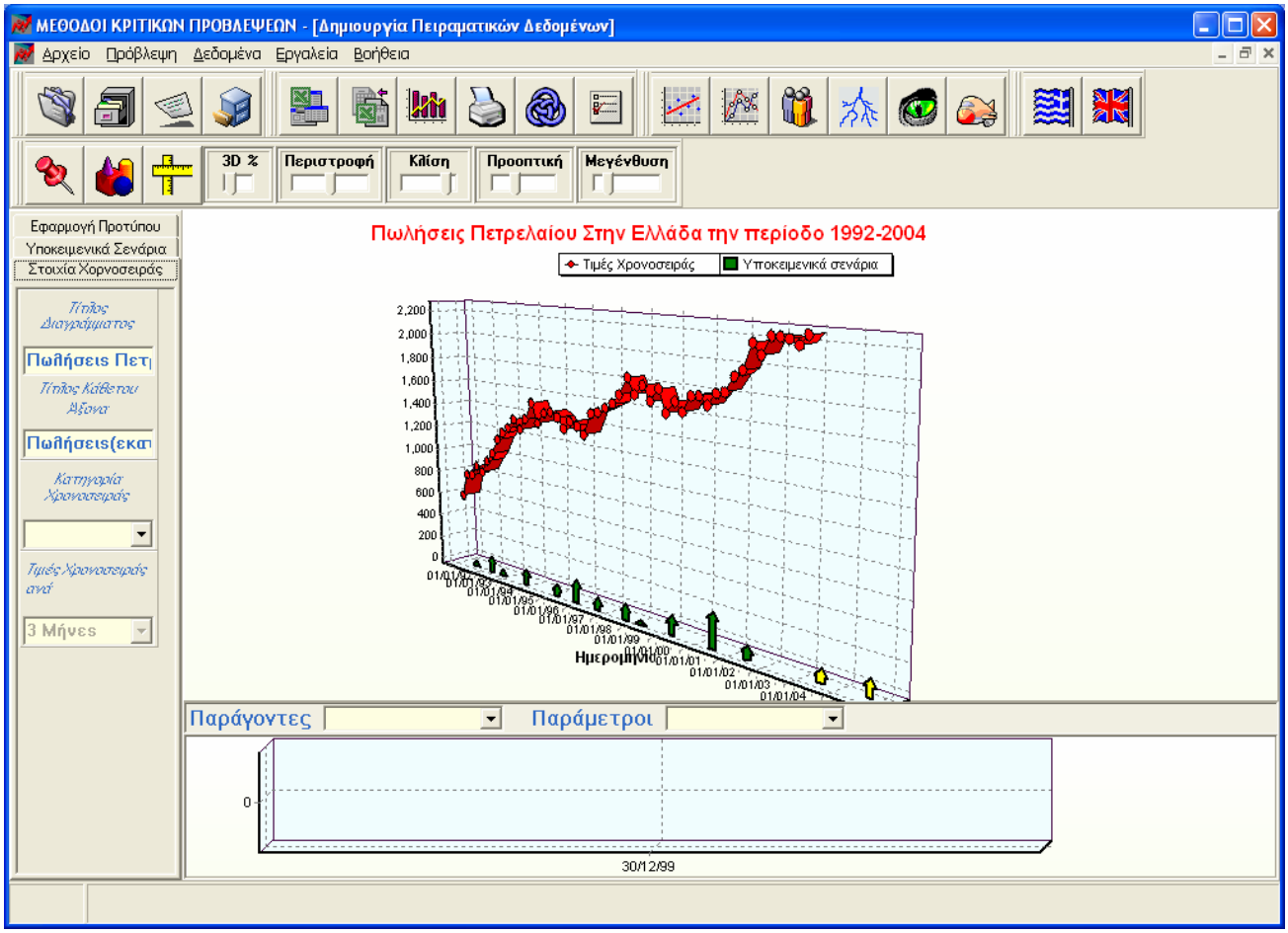
Επιλογή προβολής των τιμών των σημείων της χρονοσειράς πάνω στη γραφική παράσταση.

Εναλλαγή τρισδιάστατης-δυδιάστατης προβολής. Παρακάτω φαίνεται ένα στιγμιότυπο δυδιάστατης προβολής της ίδιας χρονοσειράς.

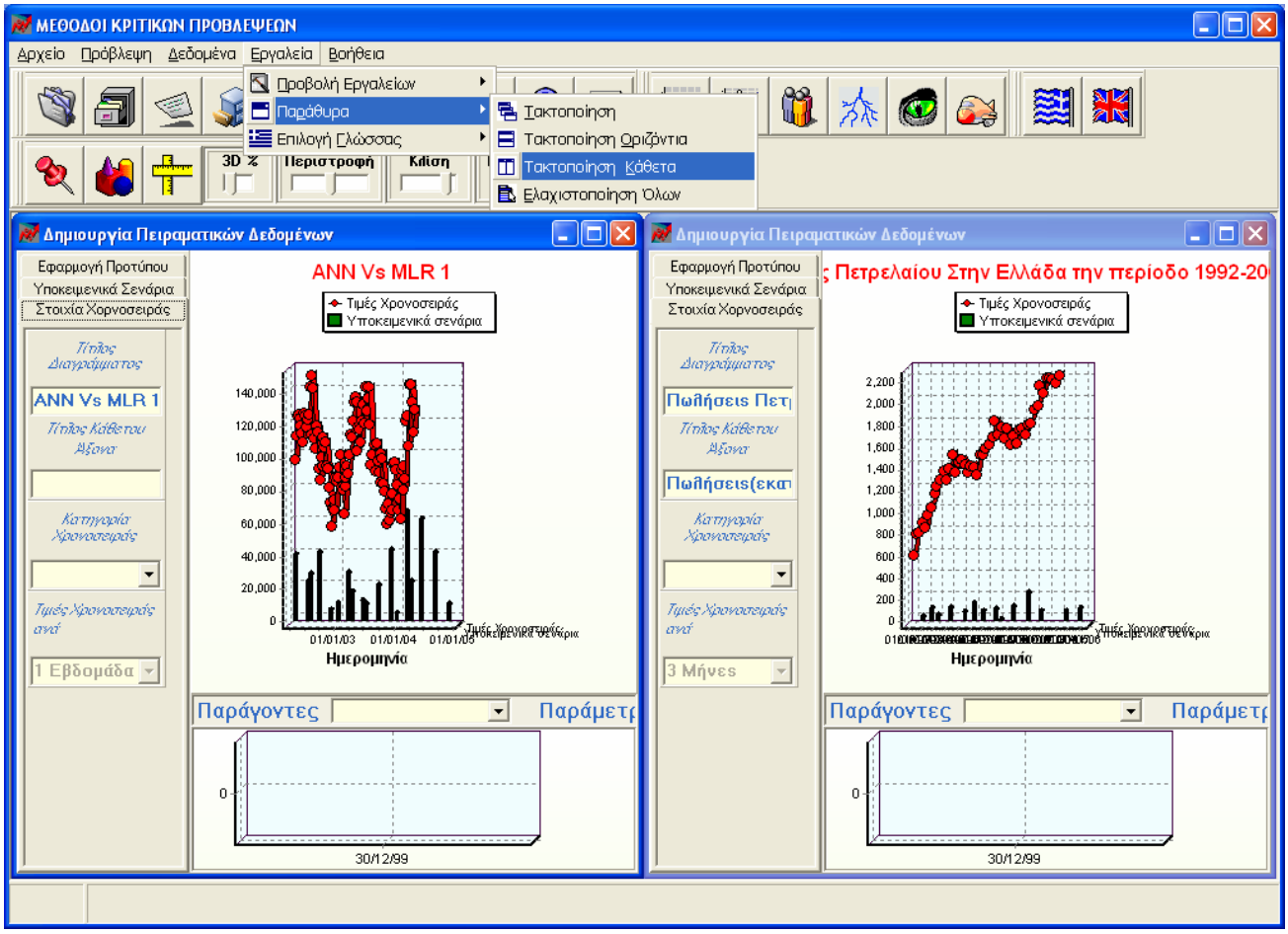



Εναλλαγή προβολής ορθοκανονικού-μη ορθοκανονικού συστήματος. Η συγκεκριμένη επιλογή μας δίνει τη δυνατότητα να δούμε τη χρονοσειρά που επιθυμούμε κάτω από διαφορετικές οπτικές γωνίες τις οποίες μπορούμε να ρυθμίσουμε από την μπάρα:


Ένα τέτοιο στιγμιότυπο φαίνεται στην επόμενη σελίδα.

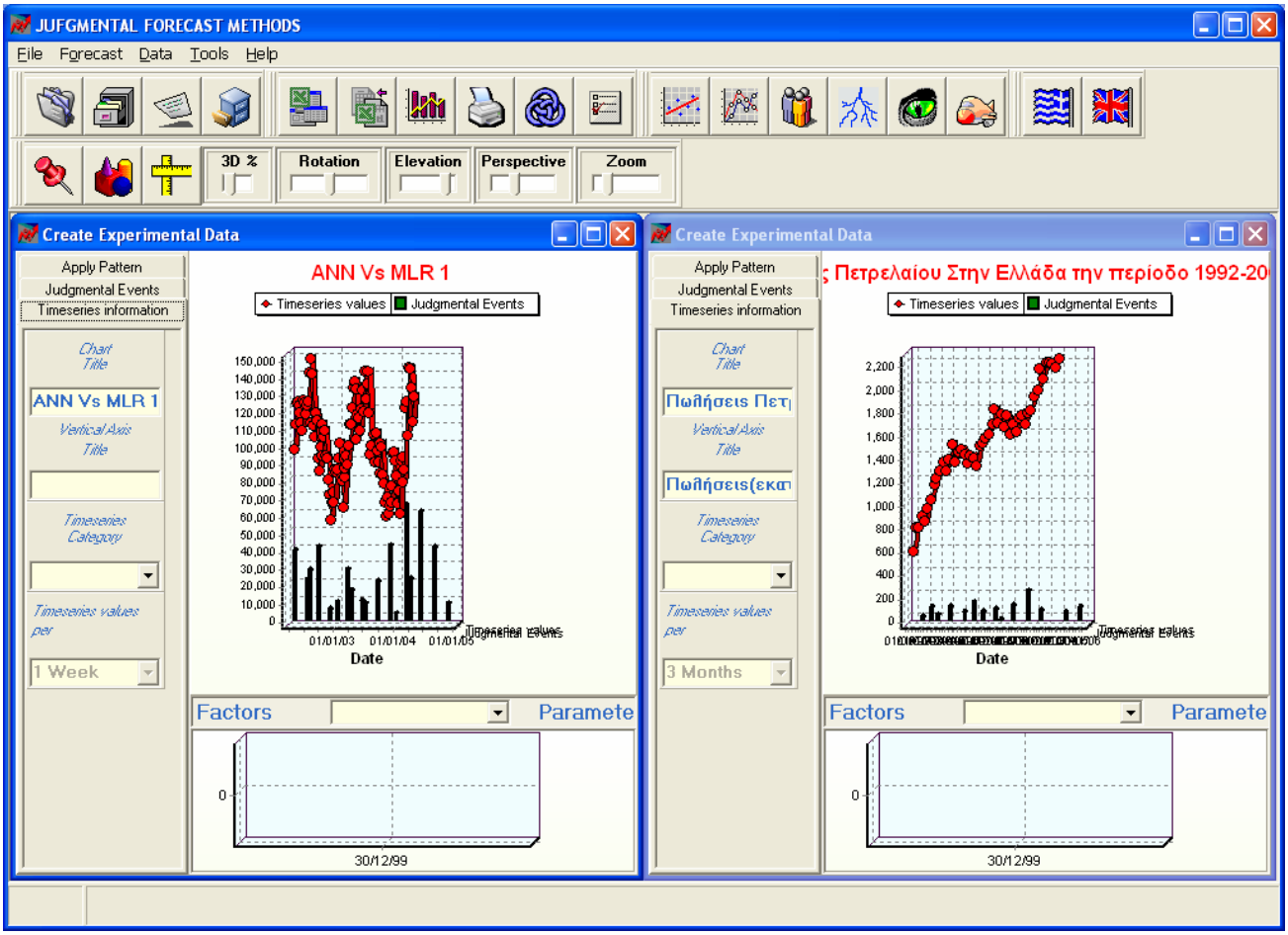

ΜΕΘΟΔΟΙ ΚΡΙΤΙΚΩΝ ΠΡΟΒΛΕΨΕΩΝ - [Δημιουργία Πειραματικών Δεδομένων]
Αρχείο Πρόβλεψη Δεδομένα Εργαλεία Βοήθεια
3D %
Περιστροφή
Κλίση
Προοπτική
Μεγένθυση
Εφαρμογή Προτύπου
Υποκειμενικά Σενάρια
Στοιχία Χορνοσειράς
Πώλήσεις Πετ
Πώλήσεις(εκα
3 Μήνες
Πωλήσεις Πετρελαίου Στην Ελλάδα την περίοδο 1992-2004
Τιμές Χρονοσειράς
Υποκειμενικά σενάρια
Παράγοντες
Παράμετροι
30/12/99

**Τακτοποίηση Παραθύρων.** Από τη συγκεκριμένη επιλογή του βασικού menu ο χρήστης μπορεί να τακτοποιήσει τα παράθυρα τα οποία έχει ανοίξει με τέσσερις διαφορετικούς τρόπους. Ένα τέτοιο παράδειγμα φαίνεται παρακάτω.

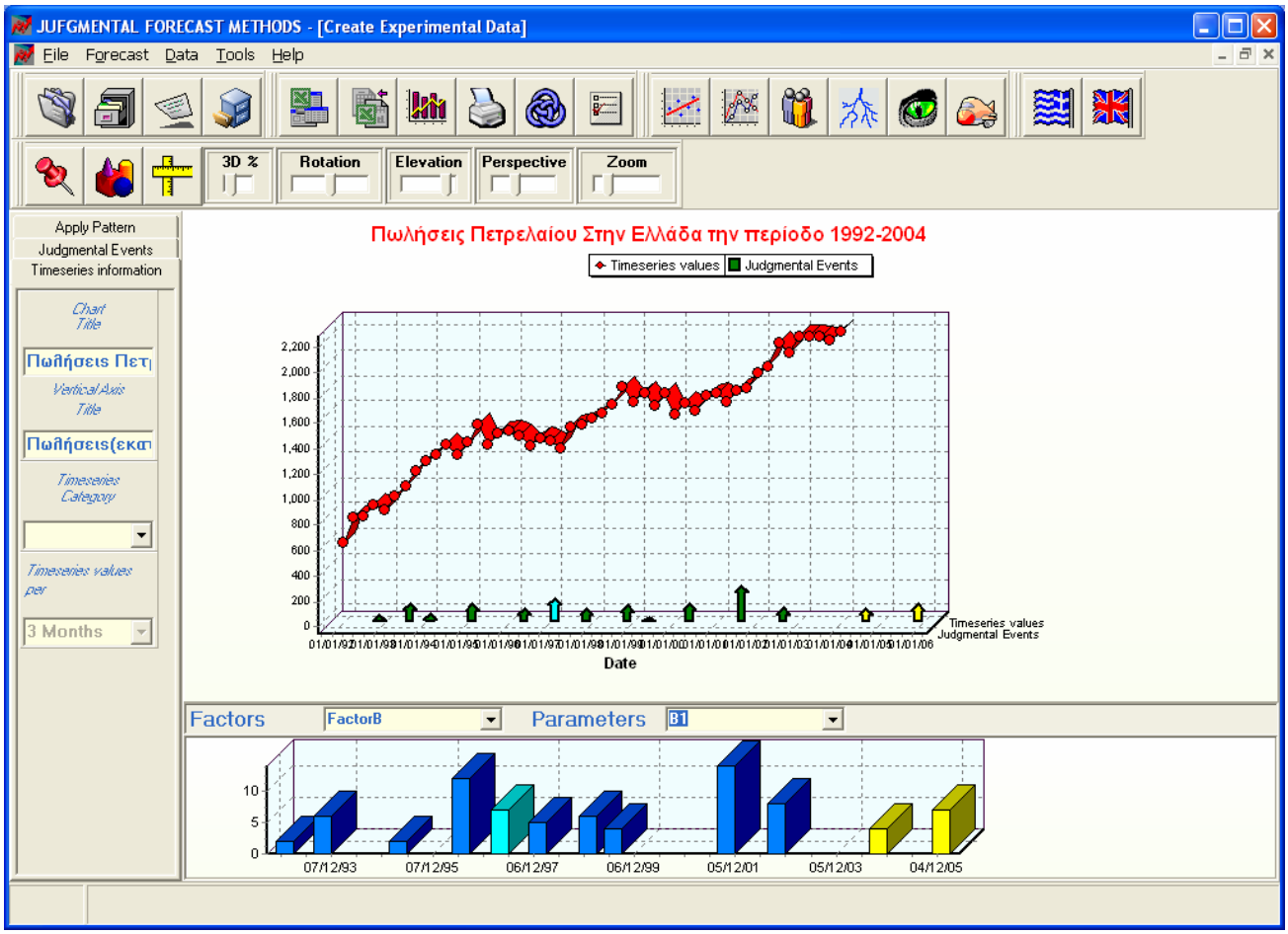


**Επιλογή Γλώσσας.** Ο χρήστης έχει τη δυνατότητα επιλογής ανάμεσα σε Ελληνικά και Αγγλικά μέσα από το βασικό menu ή από τα κουμπιά:  'Επιλογή Ελληνικών' και  'Επιλογή Αγγλικών'

Με το πάτημα ενός εκ των δύο όλα τα κείμενα και τα μηνύματα της εφαρμογής προσαρμόζονται κατάλληλα. Το παραπάνω στιγμιότυπο με επιλεγμένη την Αγγλική γλώσσα φαίνεται παρακάτω.



**Διάγραμμα προβολής παραμέτρων.** Το συγκεκριμένο διάγραμμα βρίσκεται κάτω από κάθε χρονοσειρά και επιτρέπει στο χρήστη να δει παράλληλα με το διάγραμμα της χρονοσειράς, το διάγραμμα κάθε παραμέτρου που εμφανίζεται στα ιστορικά και μελλοντικά υποκειμενικά γεγονότα. Στο παρακάτω στιγμιότυπο φαίνεται το διάγραμμα της παραμέτρου B1 για την χρονοσειρά των προηγούμενων παραδειγμάτων.




Κάθε φορά που ένα γεγονός επιλέγεται χρωματίζεται και η παράμετρος που αντιστοιχεί σε αυτό και το αντίστροφο.

Κεφάλαιο 5

# Πείραμα

## 5.1 Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιάσουμε το πείραμα που εκτελέστηκε με σκοπό τον έλεγχο της αποτελεσματικότητας των μεθόδων που παρουσιάσαμε στο προηγούμενο κεφάλαιο. Το πείραμα σχεδιάστηκε εξολοκλήρου από τον επιβλέποντα αυτής της διπλωματικής εργασίας Κ.Νικολόπουλο. Αυτό περιλαμβάνει την εφαρμογή των προαναφερομένων μεθόδων για την πρόβλεψη της επίδρασης μελλοντικών υποκειμενικών γεγονότων πολλών χρονοσειρών. Πρόκειται για πειραματικές χρονοσειρές οι οποίες κατασκευάστηκαν με κατάλληλο τρόπο ώστε να πληρούν ορισμένες προδιαγραφές. Η σύγκριση των μεθόδων που διαγωνίστηκαν έγινε με βάση τα σφάλματα MAPE MdAPE και GM-RAE. Ακολουθεί μια σύντομη περιγραφή των προδιαγραφών του πειράματος.

**Α) Χρονοσειρές.** Η κάθε χρονοσειρά αποτελείται από 15 υποκειμενικά γεγονότα, 10 ιστορικά τα οποία αποτελούν και το σετ εκπαίδευσης της κάθε μεθόδου και 5 μελλοντικά πάνω στα οποία εκτελούνται οι προβλέψεις. Το κάθε γεγονός μπορεί να είναι απλό ή σύνθετο, να αποτελείται δηλαδή από ένα οι περισσότερους παράγοντες.

**Β) Παράγοντες-Παράμετροι.** Θεωρούμε δύο παράγοντες.:

Ο πρώτος είναι ο ***'Promotions'***, έχει θετική επίδραση και περιγράφεται από τρεις παραμέτρους..

**1) Budget.** Παίρνει τιμές από 50-150 με βήμα 10. (αντιπροσωπεύει 1000 €).

**2) Duration.** Παίρνει τιμές από 1-14 (μέρες).

**3) Media.** Παίρνει τις τιμές: 1 (Έντυπα), 2 (Έντυπα + Ραδιόφωνο), 3 (Έντυπα + Ραδιόφωνο + Τηλεόραση)

Ο δεύτερος είναι ο '***Strike'*** με αρνητική επίδραση και περιγράφεται από δύο παραμέτρους.

**1) Percentage.** Κυμαίνεται από 20%-100%=0.2-1 με βήμα 0.05 (Αντιπροσωπεύει % συμμετοχή)

**2) Duration.** Παίρνει τιμές από 1-7 (μέρες).

Οι παράμετροι έχουν γραφτεί με σειρά προτεραιότητας. Δηλαδή ο παράγοντας Promotions περιγράφεται είτε μόνο από την παράμετρο Budget είτε από τις Budget και Duration είτε και από τις τρεις και όχι για παράδειγμα από τις Budget και Media.

**Γ) Συνάρτηση Επίδρασης.** Η επίδραση των υποκειμενικών γεγονότων έχει στη γενική περίπτωση τη μορφή: Impact = f(Factors, Parameters) + Noise. Για τη συνάρτηση f διακρίνουμε δύο περιπτώσεις. Στην πρώτη, αυτή είναι γραμμική και στη δεύτερη μη γραμμική όπως φαίνεται παρακάτω:

Promotion – Linear : 0.7 x Budget + 5x Duration + 20 x Media
Strike – Linear : -100 x Percentage – 2 x Duration
Promotion – Non Linear : (0.5 x Budget) x ((Duration/14) + Media)
Strike – Non Linear : -50 x Percentage x ((Duration/7) + 1)

Για το θόρυβο διακρίνουμε επίσης δύο περιπτώσεις. Στην πρώτη έχουμε μικρό θόρυβο N(0, 10) και στη δεύτερη μεγάλο N(0, 30).

**Δ) Τρεξίματα.** Θεωρούμε 10 διαφορετικές αρχικοποιήσεις της γεννήτριας θορύβου για κάθε συνδυασμό των παραπάνω (Παραμέτρων, Συνάρτησης επίδρασης, Επίπεδο θορύβου) και 5 τυχαία σετ όσον αφορά τον αριθμό των απλών και σύνθετων γεγονότων σε κάθε χρονοσειρά. Αναλυτικότερα αυτά επεξήγονται στην επόμενη ενότητα.

## 5.2 Πειραματικά Δεδομένα

Η κατασκευή των πειραματικών δεδομένων ακολουθεί τον ακόλουθο κανόνα:

**Part1_Part2_Part3_Part4_Part5**

Κάθε χρονοσειρά αποτελεί ένα στιγμιότυπο του παραπάνω κανόνα και κάθε τμήμα του κανόνα Part(i) μπορεί να πάρει τις τιμές που ακολουθούν:

**Part1:** P1, P2, P3, P1S1, P3S2 όπου το γράμμα P συμβολίζει τον παράγοντα Promotions, το γράμμα S τον παράγοντα Strike και τα νούμερα τον αριθμό των παραμέτρων από τις οποίες αποτελείται ο κάθε παράγοντας. Έτσι για παράδειγμα, για όλες οι χρονοσειρές για τις οποίες το Part1 παίρνει την τιμή P2 τα υποκειμενικά γεγονότα περιλαμβάνουν μόνο τον παράγοντα Promotions ο οποίος αποτελείται μόνο από τις δύο πρώτες παραμέτρους του. Η κάθε παράμετρος αποτελεί μια τυχαία μεταβλητή η οποία παίρνει τιμές μέσα από ένα εύρος τιμών που περιγράψαμε στην εισαγωγή και ακολουθεί ομοιόμορφη κατανομή.

**Part2:** FL, FNL. Διακρίνουμε δύο περιπτώσεις: FL : function Linear και FNL : function non linear σύμφωνα με τις οποίες ορίζεται η επίδραση των υποκειμενικών γεγονότων όπως είδαμε στην εισαγωγή.

**Part3:** NL, NH. Διακρίνουμε άλλες δύο περιπτώσεις ανάλογα με το επίπεδο του θορύβου. NL : noise low και NH : noise high.

**Part4:** R1…R10. Για κάθε συνδυασμό των Part1, Part2 και Part3 δημιουργούμε 10 συνδυασμούς με διαφορετικό θόρυβο πάνω στην επίδραση των υποκειμενικών γεγονότων στον κάθε ένα. Για κάθε λοιπόν συνδυασμό των Part1, Part2 και Part3 παίρνουμε άλλες 10 περιπτώσεις με το Part4, στις οποίες έχουμε διαφοροποίηση του θορύβου πάνω στην συνάρτηση επίδρασης, προκαλώντας έτσι διαφορετική επίδραση κάθε φορά.

**Part5:** H1…H5. Μόνο στην περίπτωση των που το Part1 παίρνει τις τιμές P1S1 και P3S2 δημιουργούμε άλλες 5 διαφορετικές περιπτώσεις οι οποίες διαφέρουν όσον αφορά την κατανομή των απλών και σύνθετων

γεγονότων. Δημιουργούμε δηλαδή κάθε φορά χρονοσειρές οι οποίες περιέχουν διαφορετικό αριθμό απλών και σύνθετων γεγονότων τόσο στα ιστορικά όσο και στα μελλοντικά γεγονότα. Σε κάθε περίπτωση όμως επιβάλουμε τον περιορισμό τόσο στα ιστορικά όσο και στα μελλοντικά γεγονότα να υπάρχει τουλάχιστον ένα σύνθετο γεγονός (που περιέχει και P1S1 και P3S2) αλλά και άλλα δύο απλά (P1 και S1 ή P3 και S3), δεσμεύοντας έτσι στο σύνολο 6 γεγονότα και αφήνοντας ελεύθερα τα υπόλοιπα 9.

Οι χρονοσειρές δημιουργούνται από όλους τους δυνατούς συνδυασμούς των Part1 – Part5. Θα πρέπει όμως να συμπληρώσουμε ότι στην περίπτωση που το Part1 παίρνει τις τιμές P1 και P1S1 το Part2 δεν μπορεί να πάρει την τιμή FNL αφού αυτή εκφυλλίζεται στην FL σύμφωνα με τον ορισμό της συνάρτησης επίδρασης στην εισαγωγή. Μόνο λοιπόν για τις δύο αυτές τιμές του Part1 το Part2 περιορίζεται στην τιμή FL. Επίσης όταν το Part1 δεν παίρνει τις τιμές P1S1 και P3S2 το Part5 δεν έχει νόημα αφού δεν υπάρχουν σύνθετα υποκειμενικά γεγονότα. Συνολικά λοιπόν έχουμε αριθμό χρονοσειρών:

Από το P1: 1 x 2 x 10 = 20 χρονοσειρές
Από το P2: 2 x 2 x 10 = 40 χρονοσειρές
Από το P3: 2 x 2 x 10 = 40 χρονοσειρές
Από το P1S1: 1 x 2 x 10 x 5 = 100 χρονοσειρές
Από το P3S2: 2 x 2 x 10 x5 = 200 χρονοσειρές

Σύνολο : 400 χρονοσειρές ή 400 x 5 = 2000 μελλοντικά υποκειμενικά γεγονότα πάνω στα οποία ελέγχεται η ακρίβεια πρόβλεψης των διαφόρων μεθόδων.

Παρακάτω παραθέτουμε δύο παραδείγματα χρονοσειρών για να γίνει σαφές πως ακριβώς αυτές κατασκευάζονται.

1) **P1_FL_NL_R1**. Η χρονοσειρά αυτή περιέχει γεγονότα που χαρακτηρίζονται μόνο από τον παράγοντα Promotions ο οποίος περιγράφεται μόνο από την παράμετρο Budget. Η συνάρτηση επίδρασης είναι η : Impact = 0.7 x Budget + N(0, 10). Από τις 10 χρονοσειρές που δημιουργούνται με διαφορετικό θόρυβο στα αντίστοιχα υποκειμενικά γεγονότα αυτή είναι η πρώτη (R1).

2) **P1S1_FL_NH_R2_H3**. Η συνάρτηση αυτή περιέχει γεγονότα που χαρακτηρίζονται και από τις δύο παραμέτρους (σύνθετα) ή μόνο από τη μία (απλά). Η επίδραση ακολουθεί το γραμμικό πρότυπο και ο θόρυβος είναι υψηλός. Εδώ ορίζεται ως :
Impact = 0.7 x Budget – 100 x Percentage. Από τις 10 διαφορετικές ομάδες χρονοσειρών με διαφορετικό θόρυβο πάνω στα αντίστοιχα γεγονότα αυτή ανήκει στη δεύτερη ομάδα και από τις 5 διαφορετικές χρονοσειρές ως προς την κατανομή των απλών και σύνθετων γεγονότων αυτή είναι η τρίτη χρονοσειρά.

Κάθε υποκειμενικό γεγονός ονομάζεται με ένα χαρακτηριστικό όνομα που ξεκινάει με την ονομασία της χρονοσειράς και καταλήγει με τη λέξη EVENT και τον αριθμό του γεγονότος. Πχ P1S1_FL_NH_R2_H3_EVENT09.

## 5.3 Αποτελέσματα του πειράματος

Τα αναλυτικά αποτελέσματα του πειράματος βρίσκονται στο αρχείο excel στο συνοδευτικό cd της διπλωματικής εργασίας. Εδώ θα παρουσιάσουμε μόνο τα τελικά συγκεντρωτικά αποτελέσματα πάνω στις 5 διαφορετικές κατηγορίες. Αυτά έχουν παραχθεί με τη μέθοδο της μαζικής πρόβλεψης χρησιμοποιώντας την αργή και ακριβέστερη μέθοδο εκπαίδευσης για τα νευρωνικά δίκτυα. Με κίτρινο χρώμα φαίνεται κάθε φορά η κατηγορία των χρονοσειρών την οποία αφορούν τα υπολογισμένα σφάλματα.

**Για την κατηγορία P3S2:**

| | P3S2_FL_NH | | | | | |
|---|---|---|---|---|---|---|
| | SLR | MLR | BN | B3N | ANN | Expert |
| MAPE | #DIV/0! | 13.38235912 | 47.56809916 | 49.26503662 | 28.92737215 | 26.22916252 |
| MdAPE | #NUM! | 6.737820272 | 25.79361549 | 24.78347292 | 16.04495981 | 11.7245138 |
| GM-RAE | #DIV/0! | 0.27242459 | 1 | 1.061621331 | 0.639646898 | 0.543522441 |
| | P3S2_FL_NL | | | | | |
| | SLR | MLR | BN | B3N | ANN | Expert |
| MAPE | #DIV/0! | 13.33828353 | 46.96598695 | 50.31469319 | 29.72856089 | 25.92215539 |
| MdAPE | #NUM! | 4.264273964 | 24.98958845 | 28.39826077 | 12.33361119 | 8.080467349 |
| GM-RAE | #DIV/0! | 0.182284353 | 1 | 1.047440666 | 0.494187284 | 0.359000609 |

P3S2_FNL_NH

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 80.81494342 | 94.26770298 | 90.08095563 | 66.15712028 | 60.7491079 |
| MdAPE | #NUM! | 34.15660539 | 47.03302705 | 52.83432404 | 31.37512265 | 28.8914195 |
| GM-RAE | #DIV/0! | 0.859192317 | 1 | 1.104843767 | 0.76502963 | 0.608511282 |

P3S2_FNL_NL

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 65.66541944 | 74.79385707 | 78.53261538 | 45.75693495 | 48.89571162 |
| MdAPE | #NUM! | 34.33812855 | 37.90794141 | 43.24607359 | 19.62063931 | 21.10513675 |
| GM-RAE | #DIV/0! | 0.877938696 | 1 | 1.143570695 | 0.532384162 | 0.5979733 |

P3S2_FL

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 13.36032132 | 47.26704306 | 49.78986491 | 29.32796652 | 26.07565895 |
| MdAPE | #NUM! | 5.760018909 | 25.38867437 | 26.50215865 | 14.19982098 | 9.633303863 |
| GM-RAE | #DIV/0! | 0 | 1 | 1.054507162 | 0.562232481 | 0.441729428 |

P3S2_FNL

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 73.24018143 | 84.53078002 | 84.30678551 | 55.95702761 | 54.82240976 |
| MdAPE | #NUM! | 34.17615482 | 42.25455309 | 48.87698666 | 25.767711 | 26.02873501 |
| GM-RAE | #DIV/0! | 0.86851493 | 1 | 1.124040459 | 0.638192494 | 0.60321928 |

P3S2_NH

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 47.09865127 | 70.91790107 | 69.67299612 | 47.54224621 | 43.48913521 |
| MdAPE | #NUM! | 15.04564728 | 33.00010886 | 36.50160125 | 22.26167301 | 18.24680214 |
| GM-RAE | #DIV/0! | 0.483802764 | 1 | 1.083016949 | 0.699534724 | 0.575099589 |

P3S2_NL

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 39.50185148 | 60.87992201 | 64.42365429 | 37.74274792 | 37.4089335 |
| MdAPE | #NUM! | 12.33299195 | 28.59727446 | 35.88101931 | 15.88017267 | 14.54243503 |
| GM-RAE | #DIV/0! | 0.400043106 | 1 | 1.094450753 | 0.512930291 | 0.463327939 |

P3S2

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 43.30025138 | 65.89891154 | 67.04832521 | 42.64249706 | 40.44903436 |
| MdAPE | #NUM! | 13.86882694 | 30.81997159 | 36.24046961 | 18.61141082 | 16.0239679 |
| GM-RAE | #DIV/0! | 0 | 1 | 1.088718841 | 0.59900964 | 0.516197353 |

## Για την κατηγορία P1S1:

| | P1S1_FL_NH | | | | | |
|---|---|---|---|---|---|---|
| | SLR | MLR | BN | B3N | ANN | Expert |
| MAPE | 29.47681913 | 23.26381738 | 61.55370006 | 61.45903888 | 32.94838791 | 33.73009842 |
| MdAPE | 13.47960706 | 10.36084227 | 24.89587006 | 37.92448286 | 15.4372113 | 15.46289217 |
| GM-RAE | 0.513169731 | 0.380972326 | 1 | 1.260960822 | 0.613874837 | 0.547749285 |

| | P1S1_FL_NL | | | | | |
|---|---|---|---|---|---|---|
| | SLR | MLR | BN | B3N | ANN | Expert |
| MAPE | 21.843084 | 19.32362246 | 61.65454424 | 54.9946563 | 28.50774933 | 28.05239489 |
| MdAPE | 7.496553337 | 6.118604084 | 24.35010336 | 31.67083244 | 11.51823767 | 10.42702183 |
| GM-RAE | 0.297293864 | 0.270327087 | 1 | 1.146696692 | 0.485021941 | 0.410159091 |

| | P1S1 | | | | | |
|---|---|---|---|---|---|---|
| | SLR | MLR | BN | B3N | ANN | Expert |
| MAPE | 25.62139735 | 21.29371992 | 61.60412215 | 58.22684759 | 30.72806862 | 30.89124665 |
| MdAPE | 10.01784124 | 8.013082416 | 24.38852675 | 34.99053955 | 12.53078573 | 12.30723627 |
| GM-RAE | 0.389516752 | 0.320916094 | 1 | 1.202472288 | 0.545658103 | 0.473987709 |

## Για την κατηγορία P3:

| | P3_FL_NH | | | | | |
|---|---|---|---|---|---|---|
| | SLR | MLR | BN | B3N | ANN | Expert |
| MAPE | #DIV/0! | 3.243762162 | 14.77218016 | 15.20640898 | 5.998005827 | 5.838240428 |
| MdAPE | #NUM! | 2.340155022 | 12.84767231 | 11.14633683 | 4.865729174 | 5.185310428 |
| GM-RAE | #DIV/0! | 0.230725605 | 1 | 0.910234084 | 0.309736418 | 0.344295934 |

| | P3_FL_NL | | | | | |
|---|---|---|---|---|---|---|
| | SLR | MLR | BN | B3N | ANN | Expert |
| MAPE | #DIV/0! | 2.592509659 | 18.08070314 | 17.98435954 | 5.206978783 | 3.930896061 |
| MdAPE | #NUM! | 2.393994435 | 14.15858051 | 12.59779978 | 4.027242836 | 3.199319635 |
| GM-RAE | #DIV/0! | 0.156081452 | 1 | 0.76001883 | 0.207039096 | 0.182148371 |

| | P3_FNL_NH | | | | | |
|---|---|---|---|---|---|---|
| | SLR | MLR | BN | B3N | ANN | Expert |
| MAPE | #DIV/0! | 22.28482251 | 39.98964096 | 45.38507236 | 11.81040928 | 18.73031109 |
| MdAPE | #NUM! | 7.874190287 | 35.08202576 | 36.86983206 | 8.360361145 | 9.82280409 |
| GM-RAE | #DIV/0! | 0.341571443 | 1 | 1.030696961 | 0.280885474 | 0.336596947 |

| | P3_FNL_NL | | | | | |
|---|---|---|---|---|---|---|
| | SLR | MLR | BN | B3N | ANN | Expert |
| MAPE | #DIV/0! | 13.76049079 | 34.94409277 | 39.05373724 | 12.06210232 | 10.83940999 |
| MdAPE | #NUM! | 9.014150335 | 27.19046949 | 23.52388419 | 8.523330036 | 6.143942562 |
| GM-RAE | #DIV/0! | 0.494331358 | 1 | 1.256778605 | 0.389146011 | 0.338669434 |

P3_FL

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 2.918135911 | 16.42644165 | 16.59538426 | 5.602492305 | 4.884568244 |
| MdAPE | #NUM! | 2.393994435 | 13.77036211 | 12.016669 | 4.586915899 | 3.77010614 |
| GM-RAE | #DIV/0! | 0.189768247 | 1 | 0.831742174 | 0.253234176 | 0.250425525 |

P3_FNL

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 18.02265665 | 37.46686686 | 42.2194048 | 11.9362558 | 14.78486054 |
| MdAPE | #NUM! | 8.858143157 | 28.78446704 | 31.79257865 | 8.523330036 | 8.818743013 |
| GM-RAE | #DIV/0! | 0.410912978 | 1 | 1.138137904 | 0.330613765 | 0.3376316 |

P3_NH

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 12.76429233 | 27.38091056 | 30.29574067 | 8.904207554 | 12.28427576 |
| MdAPE | #NUM! | 3.89965174 | 17.66654112 | 17.79315089 | 5.786712369 | 6.321236006 |
| GM-RAE | #DIV/0! | 0.280729902 | 1 | 0.968594603 | 0.294958405 | 0.340424677 |

P3_NL

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 8.176500224 | 26.51239795 | 28.51904839 | 8.634540551 | 7.385153027 |
| MdAPE | #NUM! | 3.585163225 | 17.24123013 | 14.83903862 | 5.971568829 | 4.280323328 |
| GM-RAE | #DIV/0! | 0.27776961 | 1 | 0.977330755 | 0.2838458 | 0.248370863 |

P3

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 10.47039628 | 26.94665426 | 29.40739453 | 8.769374052 | 9.834714393 |
| MdAPE | #NUM! | 3.764330252 | 17.59178623 | 15.93994512 | 5.946645539 | 5.26866117 |
| GM-RAE | #DIV/0! | 0.279245833 | 1 | 0.972952874 | 0.289348759 | 0.290777528 |

## Για την κατηγορία P2:

P2_FL_NH

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 6.224952024 | 21.08944823 | 23.61587608 | 7.479510957 | 8.323595883 |
| MdAPE | #NUM! | 5.352673413 | 14.21289269 | 15.35374024 | 5.006647883 | 5.514688918 |
| GM-RAE | #DIV/0! | 0.368619089 | 1 | 1.059119769 | 0.337469838 | 0.355115584 |

P2_FL_NL

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 3.054682477 | 19.35622029 | 20.75887625 | 5.812999421 | 3.430838464 |
| MdAPE | #NUM! | 2.125283831 | 13.0330538 | 16.34070519 | 4.633170513 | 3.001246928 |
| GM-RAE | #DIV/0! | 0.16090918 | 1 | 1.026736663 | 0.264026961 | 0.17758962 |

P2_FNL_NH

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 52.33949626 | 50.26588638 | 51.31948298 | 57.85291302 | 46.81283588 |
| MdAPE | #NUM! | 31.07552438 | 28.60767386 | 31.14649616 | 24.59574182 | 29.95052386 |
| GM-RAE | #DIV/0! | 1.091850108 | 1 | 1.112550914 | 0.949533188 | 0.905739152 |

P2_FNL_NL

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 44.37345537 | 37.05496113 | 36.92405139 | 43.72117882 | 35.41311823 |
| MdAPE | #NUM! | 25.52460378 | 26.5895847 | 25.97999562 | 22.08697752 | 16.92846038 |
| GM-RAE | #DIV/0! | 1.142074876 | 1 | 1.097517882 | 0.89617688 | 0.751066894 |

P2_FL

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 4.639817251 | 20.22283426 | 22.18737616 | 6.646255189 | 5.877217174 |
| MdAPE | #NUM! | 3.732120669 | 13.33340018 | 15.77580495 | 4.822437782 | 4.232004503 |
| GM-RAE | #DIV/0! | 0.243545058 | 1 | 1.042802521 | 0.298498134 | 0.251127142 |

P2_FNL

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 48.35647582 | 43.66042376 | 44.12176719 | 50.78704592 | 41.11297705 |
| MdAPE | #NUM! | 28.55005808 | 27.08678124 | 29.09821265 | 24.23892024 | 25.13632369 |
| GM-RAE | #DIV/0! | 1.116680158 | 1 | 1.105008834 | 0.922469344 | 0.82478524 |

P2_NH

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 29.28222414 | 35.67766731 | 37.46767953 | 32.66621199 | 27.56821588 |
| MdAPE | #NUM! | 11.14415235 | 17.45644536 | 20.85226472 | 9.219846554 | 11.16496588 |
| GM-RAE | #DIV/0! | 0.634410586 | 1 | 1.085506641 | 0.56607315 | 0.567134982 |

P2_NL

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 23.71406892 | 28.20559071 | 28.84146382 | 24.76708912 | 19.42197835 |
| MdAPE | #NUM! | 6.05431463 | 18.65836727 | 20.38531578 | 7.125530055 | 5.741826223 |
| GM-RAE | #DIV/0! | 0.428684419 | 1 | 1.061537493 | 0.486430733 | 0.365214573 |

P2

| | SLR | MLR | BN | B3N | ANN | Expert |
|---|---|---|---|---|---|---|
| MAPE | #DIV/0! | 26.49814653 | 31.94162901 | 33.15457167 | 28.71665055 | 23.49509711 |
| MdAPE | #NUM! | 8.367490225 | 18.03967269 | 20.63538189 | 8.261712364 | 7.374315675 |
| GM-RAE | #DIV/0! | 0.521499697 | 1 | 1.073455168 | 0.524743154 | 0.455110932 |

**Για την κατηγορία P1:**

| | P1_FL_NH | | | | | |
|---|---|---|---|---|---|---|
| | SLR | MLR | BN | B3N | ANN | Expert |
| MAPE | 10.14977211 | 10.14977211 | 15.67165423 | 16.46828835 | 10.30596469 | 10.398958 |
| MdAPE | 7.979314011 | 7.979314011 | 11.40835093 | 11.38460253 | 7.729063343 | 7.960019806 |
| GM-RAE | 0.695368302 | 0.695368302 | 1 | 1.007118545 | 0.635493396 | 0.706391303 |

| | P1_FL_NL | | | | | |
|---|---|---|---|---|---|---|
| | SLR | MLR | BN | B3N | ANN | Expert |
| MAPE | 4.273403733 | 4.273403733 | 7.726619093 | 9.052714581 | 6.71659811 | 5.620415018 |
| MdAPE | 3.224272969 | 3.224272969 | 4.974218108 | 4.956563604 | 4.009460346 | 3.511343052 |
| GM-RAE | 0.693990323 | 0.693990323 | 1 | 1.335051584 | 0.91018967 | 0.868081094 |

| | P1 | | | | | |
|---|---|---|---|---|---|---|
| | SLR | MLR | BN | B3N | ANN | Expert |
| MAPE | 7.21158792 | 7.21158792 | 11.69913666 | 12.76050147 | 8.511281399 | 8.009686508 |
| MdAPE | 4.338043491 | 4.338043491 | 7.547209183 | 5.987616478 | 5.643531013 | 4.895938395 |
| GM-RAE | 0.694678971 | 0.694678971 | 1 | 1.159549571 | 0.76053897 | 0.783074029 |

## 5.4 Παρατηρήσεις

Γενικά μπορούμε να παρατηρήσουμε ότι τα αποτελέσματα επαληθεύουν τις αρχικές υποψίες μας. Τα νευρωνικά δίκτυα αποδίδουν καλύτερα στις χρονοσειρές της κατηγορίας FNL όπου η επίδραση εξαρτάται μη γραμμικά από τις παραμέτρους. Αντίθετα, στις χρονοσειρές της κατηγορίας FL αποδίδει καλύτερα η μέθοδος Multiple Regression αφού μπορεί και προσδιορίζει με μεγαλύτερη ακρίβεια τη γραμμική σχέση επίδρασης και παραμέτρων. Αν δεν υπήρχε ο θόρυβος τότε η MLR θα έδινε μηδενικό σφάλμα σε αυτήν την κατηγορία χρονοσειρών. Επίσης στις χρονοσειρές της κατηγορίας NH το σφάλμα όλων των μεθόδων είναι γενικά υψηλότερο από αυτό της κατηγορίας NL των ίδιων χρονοσειρών. Και αυτό το αποτέλεσμα είναι φυσιολογικό και αναμενόμενο.

Στο συγκεκριμένο πείραμα για τη μέθοδο expert χρησιμοποιήθηκε το 60% του σετ εκπαίδευσης ως σετ εκπαίδευσης τόσο για τα νευρωνικά δίκτυα όσο και για την πολλαπλή παλινδρόμηση και το 40% για την σύγκριση και επιλογή της καταλληλότερης μεθόδου. Από τα αναλυτικότερα αποτελέσματα μέσα στο αρχείο των αποτελεσμάτων μπορούμε να παρατηρήσουμε ότι σε πολλές χρονοσειρές της κατηγορίας FL

επιλέγει η πολλαπλή παλινδρόμηση ως καταλληλότερη για την πρόβλεψη ενώ αντίθετα στην κατηγορία FNL επιλέχτηκαν τα νευρωνικά δίκτυα ως η καταλληλότερη μέθοδος. Γι' αυτό και στα συγκεντρωτικά αποτελέσματα παρατηρούμε η μέθοδος expert να παρουσιάζει γενικά στην κατηγορία χρονοσειρών FL μικρότερο σφάλμα από τα νευρωνικά δίκτυα αλλά μεγαλύτερο από αυτό της πολλαπλής παλινδρόμησης. Αν υποθέσουμε ότι η επιλογή της καταλληλότερης μεθόδου ήταν ιδανική και η expert μέθοδος διάλεγε για όλες τις χρονοσειρές της κατηγορίας FL την πολλαπλή παλινδρόμηση ως μέθοδο πρόβλεψης, τότε θα παρουσίαζε το ίδιο ακριβώς σφάλμα με αυτό της πολλαπλής παλινδρόμησης. Το ίδιο ακριβώς ισχύει και για την κατηγορία FNL στην οποία μάλιστα η expert παρουσιάζει σε αρκετές περιπτώσεις μικρότερο σφάλμα και από τα νευρωνικά δίκτυα. Αυτό συμβαίνει γιατί στην κατηγορία FNL η μέθοδος expert βασίζεται κυρίως στα νευρωνικά δίκτυα για την πρόβλεψη, τα οποία δεν εκπαιδεύονται ποτέ στο ίδιο τοπικό ελάχιστο της συνάρτησης σφάλματος, και επομένως δεν παρουσιάζουν ποτέ το ίδιο σφάλμα πρόβλεψης. Συνοψίζοντας, η μέθοδος expert φαίνεται πως αποδίδει καλήτερα από όλες τις μεθόδους σφού μπορεί και επιλέγει με κάποιο σφάλμα βέβαια την καταλληλότερη μέθοδο ανάλογα με τη γραμμικότητα ή μή γραμμικότητα που παρουσιάζει η κάθε χρονοσειρά.

Τα ίδια αποτελέσματα του πειράματος επαληθεύτηκαν αρκετές φορές αφού έγιναν περισσότερα του ενός τρεξίματα δίνοντας κάθε φορά τυχαίες τιμές στις παραμέτρους των παραγόντων των υποκειμενικών γεγονότων. Στη γενική περίπτωση λοιπόν τα νευρωνικά δίκτυα φαίνεται πως μπορούν και προβλέπουν καλύτερα στις μη γραμμικές ομάδες χρονοσειρών. Μάλιστα όταν η εκπαίδευσή τους γίνεται από το χρήστη και δοκιμάζονται και πολυπλοκότερα δίκτυα είναι εφικτό να επιτευχθεί πολύ μικρότερο τοπικό ελάχιστο στην επιφάνεια σφάλματος (κατά την εκπαίδευση) και οι προβλέψεις να προσεγγίζουν σε μεγαλύτερο βαθμό τις πραγματικές τιμές.

Κεφάλαιο 6

# Πηγαίος Κώδικας

Παρακάτω παρατίθεται ο κώδικας της εφαρμογής. Ο κώδικας αυτός αποτελεί μικρό τμήμα του κώδικα της εφαρμογής ο οποίος ανέρχεται περίπου στις 35.000 γραμμές. Παρατίθονται κυρίως οι αλγόριθμοι που αφορούν στα νευρωνικά δίκτυα και στις άλλες μεθόδους πρόβλεψεις που εξετάσαμε στα προηγούμενα κεφάλαια. Κανένα τμήμα του κώδικα δεν έχει αντιγραφεί από οποιαδήποτε άλλη πηγή. Το μόνο που έχει χρησιμοποιηθεί έτοιμο είναι ο ακόλουθος αλγόριθμος αντιστροφής πίνακα ο οποίος έχει ληφθεί από το διαδίκτυο.

```
double **darray_invert(double **a1, int n)
{
   int     i, j, k;
   int    *p = ivector_alloc(0, n);
   double **a = darray_alloc(0, 0, n, n);
   double  h, q, s, sup, pivot;


   for (i = 0; i < n; i++)
            for (j = 0; j < n; j++)
               a[i][j] = a1[i][j];

   for (k = 0; k < n; k++)
   {
            sup = 0.0;
            p[k] = 0;
            for (i = k; i < n; i++)
            {
               s = 0.0;
               for (j = k; j < n; j++)
                        s += fabs(a[i][j]);
               q = fabs(a[i][k]) / s;
               if (sup < q)
               {
                        sup = q;
                        p[k] = i;
               }
            }
            if (sup == 0.0)
               return (NULL);
            if (p[k] != k)
```

```
            for (j = 0; j < n; j++)
            {
                    h = a[k][j];
                    a[k][j] = a[p[k]][j];
                    a[p[k]][j] = h;
            }
        pivot = a[k][k];
        for (j = 0; j < n; j++)
            if (j != k)
            {
                    a[k][j] = -a[k][j] / pivot;
                    for (i = 0; i < n; i++)
                        if (i != k)
                                a[i][j] += a[i][k] * a[k][j];
            }
        for (i = 0; i < n; i++)
            a[i][k] = a[i][k] / pivot;
        a[k][k] = 1.0 / pivot;
    }
    for (k = n - 1; k >= 0; k--)
        if (p[k] != k)
            for (i = 0; i < n; i++)
            {
                    h = a[i][k];
                    a[i][k] = a[i][p[k]];
                    a[i][p[k]] = h;
            }

    ivector_free((void *) p, 0);
    return (a);
}
```

Η πηγή προέλευσης του παραπάνω αλγορίθμου είναι η ακόλουθη:

GNU LIBRARY

Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA  02111-1307  USA

## Δηλώσεις τάξης Μη Επαναληπτικών Πολυστρωματικών Νευρωνικών Δικτύων

# ANNHeaderFile.h

```
//---------------------------------------------------------------------------

#ifndef ANNHeaderFileH
#define ANNHeaderFileH
//---------------------------------------------------------------------------
#include <list.h>
#include <deque.h>
#include <math.h>
//---------------------------------------------------------------------------
typedef enum{asymptotic, nonAsymptotic, leaveOneOut} TTrainingMethod;
//---------------------------------------------------------------------------
class TNode;
class TSynapse;
double HyperbolicTangent(double U);
double HyperbolicTangentDerivative(double Y);
double NDistribution(double X, double s, double m);
bool MatrixTranspoze(long double **a, int aRows, int aCols,
                     long double **b, int bRows, int bCols);
bool MatrixSubtract(long double **a, int aRows, int aCols,
                    long double **b, int bRows, int bCols,
                    long double **c, int cRows, int cCols);
bool MatrixMultiply(long double **a, int aRows, int aCols,
                    long double **b, int bRows, int bCols,
                    long double **c, int cRows, int cCols);
//---------------------------------------------------------------------------
class TSynapse
{
public:
  __fastcall TSynapse();
  __fastcall ~TSynapse();
  __property double synapticWeight = {read = FSynapticWeight, write = FSynapticWeight};
  __property double synapticWeightDif = {read = FSynapticWeightDif, write = FSynapticWeightDif};
  __property TNode *inputNode = {read = FInputNode, write = FInputNode};
  __property TNode *outputNode = {read = FOutputNode, write = FOutputNode};
private:
  double FSynapticWeight, FSynapticWeightDif;
  TNode *FInputNode, *FOutputNode;
};
//---------------------------------------------------------------------------
class TNode
{
public:
  __fastcall TNode();
  __fastcall ~TNode();
  __property double response = {read = FResponse, write = FResponse};
  __property double inducedLocalField = {read = FInducedLocalField, write = FInducedLocalField};
  __property double error = {read = FError, write = FError};
  __property double Dj = {read = FDj, write = FDj};
  __property double SUMDjW = {read = FSUMDjW, write = FSUMDjW};
  __property double a = {read = Fa, write = Fa};
  __property double n = {read = Fn, write = Fn};
  __property AnsiString name = {read = FName, write = FName};
  __property list<TSynapse> synapses = {read = FSynapses, write = FSynapses};
  __property double (*activationFunction)(double) = {read = FActivationFunction, write = FActivationFunction};
  __property double (*activationFunctionDerivative)(double) = {read = FActivationFunctionDerivative, write = FActivationFunctionDerivative};
  __property TShape *referenceShape = {read = FReferenceShape, write = FReferenceShape};
private:
```

```
  TNode(const TNode &);
  TNode & operator=(const TNode &);
  double FResponse, FInducedLocalField, FError, FDj, FSUMDjW, Fa, Fn;
  AnsiString FName;
  list<TSynapse> FSynapses;
  double (*FActivationFunction)(double);
  double (*FActivationFunctionDerivative)(double);
  TShape *FReferenceShape;
};
//---------------------------------------------------------------------------
class TANeuralNetwork
{
public:
  __fastcall TANeuralNetwork(int *NeuronsPerHiddenLayer, int InputSize = 1, int HiddenLayers = 1, int OutputSize = 1);
  __fastcall ~TANeuralNetwork();
  TANeuralNetwork(const TANeuralNetwork &rhs);
  TANeuralNetwork & operator=(const TANeuralNetwork &rhs);
  bool __fastcall TrainANN(TObject *Sender, double **tData, double **dResponce, int nOfExamples, bool *stop, double *MeanSquareError, int
maxNumberOfEpochs);
  void __fastcall PruneANN(double **tData, double **dResponse, int nOfExamples, bool *stop);
  void __fastcall Forecast(double **tData, int nOfExamples, double *inputVector, double *outputVector);
  void __fastcall RandomizeTrainingData(double **trainingData, double **randomizedTrainingData,
                                        double **desiredResponse, double **randomizedDesiredResponse, int nOfExamples);
  double __fastcall SumCalculation(int layersLeft, TNode *currentNode, TNode *endNode);
  __property double *minResponse = {read = FMinResponse, write = FMinResponse};
  __property double *maxResponse = {read = FMaxResponse, write = FMaxResponse};
  __property double *minInput = {read = FMinInput, write = FMinInput};
  __property double *maxInput = {read = FMaxInput, write = FMaxInput};
  __property int inputSize = {read = FInputSize, write = FInputSize};
  __property int outputSize = {read = FOutputSize, write = FOutputSize};
  __property int hiddenLayers = {read = FHiddenLayers, write = FHiddenLayers};
  __property TTrainingMethod forecastMethod = {read = FForecastMethod, write = FForecastMethod};
  __property TNode fixedInput = {read = FFixedInput, write = FFixedInput};
  __property int *neuronsPerHiddenLayer = {read = FNeuronsPerHiddenLayer, write = FNeuronsPerHiddenLayer};
  __property TNode **nodes = {read = FNodes, write = FNodes};
private:
  double *FMinResponse, *FMaxResponse, *FMinInput, *FMaxInput;
  int FInputSize, FOutputSize, FHiddenLayers;
  TTrainingMethod FForecastMethod;
  TNode FFixedInput;
  int *FNeuronsPerHiddenLayer;
  TNode **FNodes;
};
//---------------------------------------------------------------------------
#endif
```

## Υλοποίηση των κυριότερων αλγορίθμων που αφορούν στα Νευρωνικά Δίκτυα.

# ANNHeaderFile.cpp

```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop

#include "ANNHeaderFile.h"
#include "ANNFormUnit.h"
#include "DataModuleUnit.h"

//---------------------------------------------------------------------------

#pragma package(smart_init)

//---------------------------------------------------------------------------
__fastcall TANeuralNetwork::TANeuralNetwork(int *NeuronsPerHiddenLayer, int InputSize, int HiddenLayers, int OutputSize)
{
  TSynapse tempSynapse;

  neuronsPerHiddenLayer = 0;
  nodes = 0;
  minResponse = 0;
  maxResponse = 0;
  minInput = 0;
  maxInput = 0;

  if((InputSize < 1) || (HiddenLayers < 1) || (OutputSize < 1)) return;
  for(int i = 0; i < HiddenLayers; i++)
  {
    if(NeuronsPerHiddenLayer[i] < 1) return;
  }

  inputSize = InputSize;
  hiddenLayers = HiddenLayers;
  outputSize = OutputSize;
  fixedInput.response = 1;

  neuronsPerHiddenLayer = new int[HiddenLayers];
  for(int i = 0; i < hiddenLayers; i++)
  {
    neuronsPerHiddenLayer[i] = NeuronsPerHiddenLayer[i];
  }

  nodes = new TNode*[hiddenLayers + 2];
  for(int i = 0; i < hiddenLayers + 2; i++)
  {
    if(i == 0)
    {
      nodes[i] = new TNode[inputSize];
    }
    else if(i == (hiddenLayers + 1))
    {
      nodes[i] = new TNode[outputSize];
    }
    else
    {
      nodes[i] = new TNode[neuronsPerHiddenLayer[i - 1]];
    }
  }
```

```
  minResponse = new double[outputSize];
  maxResponse = new double[outputSize];
  minInput = new double[inputSize];
  maxInput = new double[inputSize];

  for(int i = 0; i < hiddenLayers + 2; i++)
  {
    if(i == 0)
    {
      for(int j = 0; j < inputSize; j++)
      {
        nodes[i][j].name = AnsiString(IntToStr(i) + "-" + IntToStr(j));
        nodes[i][j].activationFunction = 0;
        nodes[i][j].activationFunctionDerivative = 0;
        nodes[i][j].response = 0;
        nodes[i][j].inducedLocalField = 0;
        nodes[i][j].Dj = 0;
        nodes[i][j].SUMDjW = 0;
        nodes[i][j].a = 0;
        nodes[i][j].n = 0;
      }
    }
    else if(i == (hiddenLayers + 1))
    {
      for(int j = 0; j < outputSize; j++)
      {
        nodes[i][j].name = AnsiString(IntToStr(i) + "-" + IntToStr(j));
        nodes[i][j].activationFunction = HyperbolicTangent;
        nodes[i][j].activationFunctionDerivative = HyperbolicTangentDerivative;
        nodes[i][j].response = 0;
        nodes[i][j].inducedLocalField = 0;
        nodes[i][j].Dj = 0;
        nodes[i][j].SUMDjW = 0;
        nodes[i][j].a = 0.3;
        nodes[i][j].n = 0.1;
      }
    }
    else
    {
      for(int j = 0; j < neuronsPerHiddenLayer[i - 1]; j++)
      {
        nodes[i][j].name = AnsiString(IntToStr(i) + "-" + IntToStr(j));
        nodes[i][j].activationFunction = HyperbolicTangent;
        nodes[i][j].activationFunctionDerivative = HyperbolicTangentDerivative;
        nodes[i][j].response = 0;
        nodes[i][j].inducedLocalField = 0;
        nodes[i][j].Dj = 0;
        nodes[i][j].SUMDjW = 0;
        nodes[i][j].a = 0.3;
        nodes[i][j].n = 0.1;
      }
    }
  }

  for(int i = 1; i < hiddenLayers + 2; i++)
  {
    if(i == 1)
    {
      for(int j = 0; j < neuronsPerHiddenLayer[i - 1]; j++)
      {
        tempSynapse.synapticWeight = 0;
        tempSynapse.synapticWeightDif = 0;
        tempSynapse.inputNode = &fixedInput;
        tempSynapse.outputNode = &nodes[i][j];
        nodes[i][j].synapses.push_back(tempSynapse);

        for(int k = 0; k < inputSize; k++)
```

```
        {
          tempSynapse.synapticWeight = 0;
          tempSynapse.synapticWeightDif = 0;
          tempSynapse.inputNode = &nodes[i - 1][k];
          tempSynapse.outputNode = &nodes[i][j];
          nodes[i][j].synapses.push_back(tempSynapse);
        }
      }
    }
    else if(i == (hiddenLayers + 1))
    {
      for(int j = 0; j < outputSize; j++)
      {
        tempSynapse.synapticWeight = 0;
        tempSynapse.synapticWeightDif = 0;
        tempSynapse.inputNode = &fixedInput;
        tempSynapse.outputNode = &nodes[i][j];
        nodes[i][j].synapses.push_back(tempSynapse);

        for(int k = 0; k < neuronsPerHiddenLayer[i - 2]; k++)
        {
          tempSynapse.synapticWeight = 0;
          tempSynapse.synapticWeightDif = 0;
          tempSynapse.inputNode = &nodes[i - 1][k];
          tempSynapse.outputNode = &nodes[i][j];
          nodes[i][j].synapses.push_back(tempSynapse);
        }
      }
    }
    else
    {
      for(int j = 0; j < neuronsPerHiddenLayer[i - 1]; j++)
      {
        tempSynapse.synapticWeight = 0;
        tempSynapse.synapticWeightDif = 0;
        tempSynapse.inputNode = &fixedInput;
        tempSynapse.outputNode = &nodes[i][j];
        nodes[i][j].synapses.push_back(tempSynapse);

        for(int k = 0; k < neuronsPerHiddenLayer[i - 2]; k++)
        {
          tempSynapse.synapticWeight = 0;
          tempSynapse.synapticWeightDif = 0;
          tempSynapse.inputNode = &nodes[i - 1][k];
          tempSynapse.outputNode = &nodes[i][j];
          nodes[i][j].synapses.push_back(tempSynapse);
        }
      }
    }
  }
}
//---------------------------------------------------------------------------
TANeuralNetwork::TANeuralNetwork(const TANeuralNetwork &rhs)
{
  TSynapse tempSynapse;

  neuronsPerHiddenLayer = 0;
  nodes = 0;
  minResponse = 0;
  maxResponse = 0;
  minInput = 0;
  maxInput = 0;

  inputSize = rhs.inputSize;
  hiddenLayers = rhs.hiddenLayers;
  outputSize = rhs.outputSize;
  fixedInput.response = 1;
```

```
  forecastMethod = rhs.forecastMethod;

  neuronsPerHiddenLayer = new int[rhs.hiddenLayers];
  for(int i = 0; i < hiddenLayers; i++)
  {
    neuronsPerHiddenLayer[i] = rhs.neuronsPerHiddenLayer[i];
  }

  minResponse = new double[outputSize];
  maxResponse = new double[outputSize];
  minInput = new double[inputSize];
  maxInput = new double[inputSize];

  for(int i = 0; i < outputSize; i++)
  {
    minResponse[i] = rhs.minResponse[i];
    maxResponse[i] = rhs.maxResponse[i];
  }

  for(int i = 0; i < inputSize; i++)
  {
    minInput[i] = rhs.minInput[i];
    maxInput[i] = rhs.maxInput[i];
  }

  nodes = new TNode*[hiddenLayers + 2];
  for(int i = 0; i < hiddenLayers + 2; i++)
  {
    if(i == 0)
    {
      nodes[i] = new TNode[inputSize];
    }
    else if(i == (hiddenLayers + 1))
    {
      nodes[i] = new TNode[outputSize];
    }
    else
    {
      nodes[i] = new TNode[neuronsPerHiddenLayer[i - 1]];
    }
  }

  for(int i = 0; i < hiddenLayers + 2; i++)
  {
    if(i == 0)
    {
      for(int j = 0; j < inputSize; j++)
      {
        nodes[i][j].name = rhs.nodes[i][j].name;
        nodes[i][j].activationFunction = rhs.nodes[i][j].activationFunction;
        nodes[i][j].activationFunctionDerivative = rhs.nodes[i][j].activationFunctionDerivative;
        nodes[i][j].response = rhs.nodes[i][j].response;
        nodes[i][j].inducedLocalField = rhs.nodes[i][j].inducedLocalField;
        nodes[i][j].Dj = rhs.nodes[i][j].Dj;
        nodes[i][j].SUMDjW = rhs.nodes[i][j].SUMDjW;
        nodes[i][j].a = rhs.nodes[i][j].a;
        nodes[i][j].n = rhs.nodes[i][j].n;
      }
    }
    else if(i == (hiddenLayers + 1))
    {
      for(int j = 0; j < outputSize; j++)
      {
        nodes[i][j].name = rhs.nodes[i][j].name;
        nodes[i][j].activationFunction = rhs.nodes[i][j].activationFunction;
        nodes[i][j].activationFunctionDerivative = rhs.nodes[i][j].activationFunctionDerivative;
        nodes[i][j].response = rhs.nodes[i][j].response;
```

```
        nodes[i][j].inducedLocalField = rhs.nodes[i][j].inducedLocalField;
        nodes[i][j].Dj = rhs.nodes[i][j].Dj;
        nodes[i][j].SUMDjW = rhs.nodes[i][j].SUMDjW;
        nodes[i][j].a = rhs.nodes[i][j].a;
        nodes[i][j].n = rhs.nodes[i][j].n;
      }
    }
    else
    {
      for(int j = 0; j < neuronsPerHiddenLayer[i - 1]; j++)
      {
        nodes[i][j].name = rhs.nodes[i][j].name;
        nodes[i][j].activationFunction = rhs.nodes[i][j].activationFunction;
        nodes[i][j].activationFunctionDerivative = rhs.nodes[i][j].activationFunctionDerivative;
        nodes[i][j].response = rhs.nodes[i][j].response;
        nodes[i][j].inducedLocalField = rhs.nodes[i][j].inducedLocalField;
        nodes[i][j].Dj = rhs.nodes[i][j].Dj;
        nodes[i][j].SUMDjW = rhs.nodes[i][j].SUMDjW;
        nodes[i][j].a = rhs.nodes[i][j].a;
        nodes[i][j].n = rhs.nodes[i][j].n;
      }
    }
  }

  for(int i = 1; i < hiddenLayers + 2; i++)
  {
    if(i == 1)
    {
      for(int j = 0; j < neuronsPerHiddenLayer[i - 1]; j++)
      {
        for(list<TSynapse>::iterator k = rhs.nodes[i][j].synapses.begin(); k != rhs.nodes[i][j].synapses.end(); k++)
        {
          if((*k).inputNode != &(rhs.fixedInput))
          {
            tempSynapse.synapticWeight = (*k).synapticWeight;
            tempSynapse.synapticWeightDif = (*k).synapticWeightDif;
            for(int l = 0; l < inputSize; l++)
            {
              if(nodes[i-1][l].name == (*k).inputNode->name)
              {
                tempSynapse.inputNode = &nodes[i - 1][l];
                break;
              }
            }
            tempSynapse.outputNode = &nodes[i][j];
            nodes[i][j].synapses.push_back(tempSynapse);
          }
          else
          {
            tempSynapse.synapticWeight = (*k).synapticWeight;
            tempSynapse.synapticWeightDif = (*k).synapticWeightDif;
            tempSynapse.inputNode = &fixedInput;
            tempSynapse.outputNode = &nodes[i][j];
            nodes[i][j].synapses.push_back(tempSynapse);
          }
        }
      }
    }
    else if(i == (hiddenLayers + 1))
    {
      for(int j = 0; j < outputSize; j++)
      {
        for(list<TSynapse>::iterator k = rhs.nodes[i][j].synapses.begin(); k != rhs.nodes[i][j].synapses.end(); k++)
        {
          if((*k).inputNode != &(rhs.fixedInput))
          {
            tempSynapse.synapticWeight = (*k).synapticWeight;
```

```
            tempSynapse.synapticWeightDif = (*k).synapticWeightDif;
            for(int l = 0; l < neuronsPerHiddenLayer[i - 2]; l++)
            {
              if(nodes[i-1][l].name == (*k).inputNode->name)
              {
                tempSynapse.inputNode = &nodes[i - 1][l];
                break;
              }
            }
            tempSynapse.outputNode = &nodes[i][j];
            nodes[i][j].synapses.push_back(tempSynapse);
          }
          else
          {
            tempSynapse.synapticWeight = (*k).synapticWeight;
            tempSynapse.synapticWeightDif = (*k).synapticWeightDif;
            tempSynapse.inputNode = &fixedInput;
            tempSynapse.outputNode = &nodes[i][j];
            nodes[i][j].synapses.push_back(tempSynapse);
          }
        }
      }
    }
    else
    {
      for(int j = 0; j < neuronsPerHiddenLayer[i - 1]; j++)
      {
        for(list<TSynapse>::iterator k = rhs.nodes[i][j].synapses.begin(); k != rhs.nodes[i][j].synapses.end(); k++)
        {
          if((*k).inputNode != &(rhs.fixedInput))
          {
            tempSynapse.synapticWeight = (*k).synapticWeight;
            tempSynapse.synapticWeightDif = (*k).synapticWeightDif;
            for(int l = 0; l < neuronsPerHiddenLayer[i - 2]; l++)
            {
              if(nodes[i-1][l].name == (*k).inputNode->name)
              {
                tempSynapse.inputNode = &nodes[i - 1][l];
                break;
              }
            }
            tempSynapse.outputNode = &nodes[i][j];
            nodes[i][j].synapses.push_back(tempSynapse);
          }
          else
          {
            tempSynapse.synapticWeight = (*k).synapticWeight;
            tempSynapse.synapticWeightDif = (*k).synapticWeightDif;
            tempSynapse.inputNode = &fixedInput;
            tempSynapse.outputNode = &nodes[i][j];
            nodes[i][j].synapses.push_back(tempSynapse);
          }
        }
      }
    }
  }
}
//---------------------------------------------------------------------------
TANeuralNetwork & TANeuralNetwork::operator=(const TANeuralNetwork &rhs)
{
  TSynapse tempSynapse;

  if(this == &rhs) return *this;

  if(neuronsPerHiddenLayer) delete [] neuronsPerHiddenLayer;

  if(nodes)
```

```
{
  for(int i = 0; i < hiddenLayers + 2; i++)
  {
    delete [] nodes[i];
  }
  delete [] nodes;
}

if(minResponse) delete [] minResponse;
if(maxResponse) delete [] maxResponse;
if(minInput) delete [] minInput;
if(maxInput) delete [] maxInput;

neuronsPerHiddenLayer = 0;
nodes = 0;
minResponse = 0;
maxResponse = 0;
minInput = 0;
maxInput = 0;

inputSize = rhs.inputSize;
hiddenLayers = rhs.hiddenLayers;
outputSize = rhs.outputSize;
fixedInput.response = 1;
forecastMethod = rhs.forecastMethod;

neuronsPerHiddenLayer = new int[rhs.hiddenLayers];
for(int i = 0; i < hiddenLayers; i++)
{
  neuronsPerHiddenLayer[i] = rhs.neuronsPerHiddenLayer[i];
}

minResponse = new double[outputSize];
maxResponse = new double[outputSize];
minInput = new double[inputSize];
maxInput = new double[inputSize];

for(int i = 0; i < outputSize; i++)
{
  minResponse[i] = rhs.minResponse[i];
  maxResponse[i] = rhs.maxResponse[i];
}

for(int i = 0; i < inputSize; i++)
{
  minInput[i] = rhs.minInput[i];
  maxInput[i] = rhs.maxInput[i];
}

nodes = new TNode*[hiddenLayers + 2];
for(int i = 0; i < hiddenLayers + 2; i++)
{
  if(i == 0)
  {
    nodes[i] = new TNode[inputSize];
  }
  else if(i == (hiddenLayers + 1))
  {
    nodes[i] = new TNode[outputSize];
  }
  else
  {
    nodes[i] = new TNode[neuronsPerHiddenLayer[i - 1]];
  }
}

for(int i = 0; i < hiddenLayers + 2; i++)
```

```
{
  if(i == 0)
  {
    for(int j = 0; j < inputSize; j++)
    {
      nodes[i][j].name = rhs.nodes[i][j].name;
      nodes[i][j].activationFunction = rhs.nodes[i][j].activationFunction;
      nodes[i][j].activationFunctionDerivative = rhs.nodes[i][j].activationFunctionDerivative;
      nodes[i][j].response = rhs.nodes[i][j].response;
      nodes[i][j].inducedLocalField = rhs.nodes[i][j].inducedLocalField;
      nodes[i][j].Dj = rhs.nodes[i][j].Dj;
      nodes[i][j].SUMDjW = rhs.nodes[i][j].SUMDjW;
      nodes[i][j].a = rhs.nodes[i][j].a;
      nodes[i][j].n = rhs.nodes[i][j].n;
    }
  }
  else if(i == (hiddenLayers + 1))
  {
    for(int j = 0; j < outputSize; j++)
    {
      nodes[i][j].name = rhs.nodes[i][j].name;
      nodes[i][j].activationFunction = rhs.nodes[i][j].activationFunction;
      nodes[i][j].activationFunctionDerivative = rhs.nodes[i][j].activationFunctionDerivative;
      nodes[i][j].response = rhs.nodes[i][j].response;
      nodes[i][j].inducedLocalField = rhs.nodes[i][j].inducedLocalField;
      nodes[i][j].Dj = rhs.nodes[i][j].Dj;
      nodes[i][j].SUMDjW = rhs.nodes[i][j].SUMDjW;
      nodes[i][j].a = rhs.nodes[i][j].a;
      nodes[i][j].n = rhs.nodes[i][j].n;
    }
  }
  else
  {
    for(int j = 0; j < neuronsPerHiddenLayer[i - 1]; j++)
    {
      nodes[i][j].name = rhs.nodes[i][j].name;
      nodes[i][j].activationFunction = rhs.nodes[i][j].activationFunction;
      nodes[i][j].activationFunctionDerivative = rhs.nodes[i][j].activationFunctionDerivative;
      nodes[i][j].response = rhs.nodes[i][j].response;
      nodes[i][j].inducedLocalField = rhs.nodes[i][j].inducedLocalField;
      nodes[i][j].Dj = rhs.nodes[i][j].Dj;
      nodes[i][j].SUMDjW = rhs.nodes[i][j].SUMDjW;
      nodes[i][j].a = rhs.nodes[i][j].a;
      nodes[i][j].n = rhs.nodes[i][j].n;
    }
  }
}

for(int i = 1; i < hiddenLayers + 2; i++)
{
  if(i == 1)
  {
    for(int j = 0; j < neuronsPerHiddenLayer[i - 1]; j++)
    {
      for(list<TSynapse>::iterator k = rhs.nodes[i][j].synapses.begin(); k != rhs.nodes[i][j].synapses.end(); k++)
      {
        if((*k).inputNode != &(rhs.fixedInput))
        {
          tempSynapse.synapticWeight = (*k).synapticWeight;
          tempSynapse.synapticWeightDif = (*k).synapticWeightDif;
          for(int l = 0; l < inputSize; l++)
          {
            if(nodes[i-1][l].name == (*k).inputNode->name)
            {
              tempSynapse.inputNode = &nodes[i - 1][l];
              break;
            }
```

```
            }
            tempSynapse.outputNode = &nodes[i][j];
            nodes[i][j].synapses.push_back(tempSynapse);
          }
          else
          {
            tempSynapse.synapticWeight = (*k).synapticWeight;
            tempSynapse.synapticWeightDif = (*k).synapticWeightDif;
            tempSynapse.inputNode = &fixedInput;
            tempSynapse.outputNode = &nodes[i][j];
            nodes[i][j].synapses.push_back(tempSynapse);
          }
        }
      }
    }
    else if(i == (hiddenLayers + 1))
    {
      for(int j = 0; j < outputSize; j++)
      {
        for(list<TSynapse>::iterator k = rhs.nodes[i][j].synapses.begin(); k != rhs.nodes[i][j].synapses.end(); k++)
        {
          if((*k).inputNode != &(rhs.fixedInput))
          {
            tempSynapse.synapticWeight = (*k).synapticWeight;
            tempSynapse.synapticWeightDif = (*k).synapticWeightDif;
            for(int l = 0; l < neuronsPerHiddenLayer[i - 2]; l++)
            {
              if(nodes[i-1][l].name == (*k).inputNode->name)
              {
                tempSynapse.inputNode = &nodes[i - 1][l];
                break;
              }
            }
            tempSynapse.outputNode = &nodes[i][j];
            nodes[i][j].synapses.push_back(tempSynapse);
          }
          else
          {
            tempSynapse.synapticWeight = (*k).synapticWeight;
            tempSynapse.synapticWeightDif = (*k).synapticWeightDif;
            tempSynapse.inputNode = &fixedInput;
            tempSynapse.outputNode = &nodes[i][j];
            nodes[i][j].synapses.push_back(tempSynapse);
          }
        }
      }
    }
    else
    {
      for(int j = 0; j < neuronsPerHiddenLayer[i - 1]; j++)
      {
        for(list<TSynapse>::iterator k = rhs.nodes[i][j].synapses.begin(); k != rhs.nodes[i][j].synapses.end(); k++)
        {
          if((*k).inputNode != &(rhs.fixedInput))
          {
            tempSynapse.synapticWeight = (*k).synapticWeight;
            tempSynapse.synapticWeightDif = (*k).synapticWeightDif;
            for(int l = 0; l < neuronsPerHiddenLayer[i - 2]; l++)
            {
              if(nodes[i-1][l].name == (*k).inputNode->name)
              {
                tempSynapse.inputNode = &nodes[i - 1][l];
                break;
              }
            }
            tempSynapse.outputNode = &nodes[i][j];
            nodes[i][j].synapses.push_back(tempSynapse);
```

```
            }
            else
            {
              tempSynapse.synapticWeight = (*k).synapticWeight;
              tempSynapse.synapticWeightDif = (*k).synapticWeightDif;
              tempSynapse.inputNode = &fixedInput;
              tempSynapse.outputNode = &nodes[i][j];
              nodes[i][j].synapses.push_back(tempSynapse);
            }
          }
        }
      }
    }

  return *this;
}
//---------------------------------------------------------------------------
__fastcall TANeuralNetwork::~TANeuralNetwork()
{
  if(neuronsPerHiddenLayer) delete [] neuronsPerHiddenLayer;

  if(nodes)
  {
    for(int i = 0; i < hiddenLayers + 2; i++)
    {
      delete [] nodes[i];
    }
    delete [] nodes;
  }

  if(minResponse) delete [] minResponse;
  if(maxResponse) delete [] maxResponse;
  if(minInput) delete [] minInput;
  if(maxInput) delete [] maxInput;
}
//---------------------------------------------------------------------------
__fastcall TNode::TNode()
{
}
//---------------------------------------------------------------------------
__fastcall TNode::~TNode()
{
}
//---------------------------------------------------------------------------
__fastcall TSynapse::TSynapse()
{
}
//---------------------------------------------------------------------------
__fastcall TSynapse::~TSynapse()
{
}
//---------------------------------------------------------------------------
bool __fastcall TANeuralNetwork::TrainANN(TObject *Sender, double **tData, double **dResponse, int nOfExamples, bool *stop, double
*MeanSquareError, int maxNumberOfEpochs)
{
  try
  {
    int totalSynapses, numberOfEpochs, estimationSize, validationSize, period;
    double c, s, m, U1, U2, X, Y, result, rOptimal;
    double tempDouble, AROASE, ASE, pASE, MSE, pMSE;
    double **trainingData, **desiredResponse, *meanVector;
    double **randomizedTrainingData, **randomizedDesiredResponse;
    double **estimationSubset, **validationSubset, **estimationResponse, **validationResponse;
    bool valueFound;
    TANNForm *myForm;
    TMainForm *myForm2;
```

```
trainingData = 0;
desiredResponse = 0;
meanVector = 0;
randomizedTrainingData = 0;
randomizedDesiredResponse = 0;
estimationSubset = 0;
validationSubset = 0;
estimationResponse = 0;
validationResponse = 0;

myForm = dynamic_cast<TANNForm *>(Sender);
myForm2 = dynamic_cast<TMainForm *>(Sender);

// 1) INITIALIZATION

trainingData = new double*[nOfExamples];
for(int i = 0; i < nOfExamples; i++)
{
  trainingData[i] = new double[inputSize];
}

desiredResponse = new double*[nOfExamples];
for(int i = 0; i < nOfExamples; i++)
{
  desiredResponse[i] = new double[outputSize];
}

meanVector = new double[inputSize];

// 1A) INITIALIZE SYNAPTIC WEIGHTS AND THRESHOLDS FROM A UNIFORM DISTRIBUTION

RandSeed = rand();

if(myForm)
{
  myForm->Timer1->Enabled = false;
  myForm->curAction = countAction;
  if(DataModule1->curLanguage == Greek)
  {
    myForm->StatusBar1->Panels->Items[1]->Text = "Αρχικοποίηση δικτύου.";
  }
  else if(DataModule1->curLanguage == English)
  {
    myForm->StatusBar1->Panels->Items[1]->Text = "Network initialization.";
  }
  myForm->StatusBar1->Invalidate();
}
if(myForm2)
{
  DataModule1->Timer1->Enabled = false;
  myForm2->curAction = countAction;
  if(DataModule1->curLanguage == Greek)
  {
    myForm2->StatusBar1->Panels->Items[1]->Text = "Αρχικοποίηση δικτύου.";
  }
  else if(DataModule1->curLanguage == English)
  {
    myForm2->StatusBar1->Panels->Items[1]->Text = "Network initialization.";
  }
  myForm2->StatusBar1->Invalidate();
}

for(int i = 0; i < hiddenLayers; i++)
{
  for(int j = 0; j < neuronsPerHiddenLayer[i]; j++)
  {
    Application->ProcessMessages();
```

```
      m = 0;
      if(nodes[i + 1][j].synapses.size() > 0) s = double(1)/sqrt(double(nodes[i + 1][j].synapses.size()));
      for(list<TSynapse>::iterator k = nodes[i + 1][j].synapses.begin(); k != nodes[i + 1][j].synapses.end(); k++)
      {
        valueFound = false;
        while(!valueFound)
        {
          c = double(1)/(s*sqrt(double(2)*M_PI));
          U1 = double(rand()) / double(RAND_MAX);
          U2 = double(rand()) / double(RAND_MAX);
          X = m - double(1000*s) + double(2000*s) * U1;
          Y = c * U2;
          if(Y < NDistribution(X, s, m))
          {
            result = X;
            valueFound = true;
          }
        }
        (*k).synapticWeight = result;

        if(*stop)
        {
          if(trainingData)
          {
            for(int i = 0; i < nOfExamples; i++)
            {
              delete [] trainingData[i];
            }
            delete [] trainingData;
          }

          if(desiredResponse)
          {
            for(int i = 0; i < nOfExamples; i++)
            {
              delete [] desiredResponse[i];
            }
            delete [] desiredResponse;
          }

          if(meanVector)
          {
            delete [] meanVector;
          }

          return false;
        }
      }
    }
  }

  for(int i = 0; i < outputSize; i++)
  {
    Application->ProcessMessages();
    m = 0;
    if(nodes[hiddenLayers + 1][i].synapses.size() > 0) s = double(1)/sqrt(double(nodes[hiddenLayers + 1][i].synapses.size()));
    for(list<TSynapse>::iterator j = nodes[hiddenLayers + 1][i].synapses.begin(); j != nodes[hiddenLayers + 1][i].synapses.end(); j++)
    {
      valueFound = false;
      while(!valueFound)
      {
        c = double(1)/(s*sqrt(double(2)*M_PI));
        U1 = double(rand()) / double(RAND_MAX);
        U2 = double(rand()) / double(RAND_MAX);
        X = m - double(1000*s) + double(2000*s) * U1;
        Y = c * U2;
        if(Y < NDistribution(X, s, m))
```

```
        {
          result = X;
          valueFound = true;
        }
      }
      (*j).synapticWeight = result;

      if(*stop)
      {
        if(trainingData)
        {
          for(int i = 0; i < nOfExamples; i++)
          {
            delete [] trainingData[i];
          }
          delete [] trainingData;
        }

        if(desiredResponse)
        {
          for(int i = 0; i < nOfExamples; i++)
          {
            delete [] desiredResponse[i];
          }
          delete [] desiredResponse;
        }

        if(meanVector)
        {
          delete [] meanVector;
        }

        return false;
      }
    }
  }

  // 1B) TRANFORM TARGET VALUES AND NORMALIZE INPUTS

  for(int i = 0; i < outputSize; i++)
  {
    minResponse[i] = dResponse[0][i];
    maxResponse[i] = dResponse[0][i];
  }

  for(int i = 0; i < inputSize; i++)
  {
    meanVector[i] = 0;
  }

  for(int i = 1; i < nOfExamples; i++)
  {
    for(int j = 0; j < outputSize; j++)
    {
      if(dResponse[i][j] > maxResponse[j]) maxResponse[j] = dResponse[i][j];
      if(dResponse[i][j] < minResponse[j]) minResponse[j] = dResponse[i][j];
    }
  }

  for(int i = 0; i < nOfExamples; i++)
  {
    for(int j = 0; j < outputSize; j++)
    {
      desiredResponse[i][j] = double(-1) + double(2)*((dResponse[i][j] - minResponse[j])/(maxResponse[j] - minResponse[j]));
    }

    for(int j = 0; j < inputSize; j++)
```

```
    {
      meanVector[j] += tData[i][j];
    }
  }

  for(int i = 0; i < inputSize; i++)
  {
    meanVector[i] = meanVector[i]/double(nOfExamples);
  }

  for(int i = 0; i < nOfExamples; i++)
  {
    for(int j = 0; j < inputSize; j++)
    {
      trainingData[i][j] = tData[i][j] - meanVector[j];
    }
  }

  for(int i = 0; i < inputSize; i++)
  {
    minInput[i] = trainingData[0][i];
    maxInput[i] = trainingData[0][i];
  }

  for(int i = 1; i < nOfExamples; i++)
  {
    for(int j = 0; j < inputSize; j++)
    {
      if(trainingData[i][j] > maxInput[j]) maxInput[j] = trainingData[i][j];
      if(trainingData[i][j] < minInput[j]) minInput[j] = trainingData[i][j];
    }
  }

  for(int i = 0; i < nOfExamples; i++)
  {
    for(int j = 0; j < inputSize; j++)
    {
      trainingData[i][j] = double(-1) + double(2)*((trainingData[i][j] - minInput[j])/(maxInput[j] - minInput[j]));
    }
  }

  // 2)CALCULATE TOTAL NTEWORK SYNAPSES

  totalSynapses = 0;
  for(int i = 0; i < hiddenLayers + 1; i++)
  {
    if(i != hiddenLayers)
    {
      for(int j = 0; j < neuronsPerHiddenLayer[i]; j++)
      {
        totalSynapses += nodes[i + 1][j].synapses.size();
      }
    }
    else
    {
      for(int j = 0; j < outputSize; j++)
      {
        totalSynapses += nodes[i + 1][j].synapses.size();
      }
    }
  }

  // 3) TRAIN THE NETWORK USING THE BACK-PROPAGATION ALGORITHM AND THE APPROPRIATE STOPPING CRITERIA

  if(forecastMethod == asymptotic)
  {
    //ASYMPTOTIC MODE
```

```
AROASE = 1;
ASE = 0;
numberOfEpochs = 0;

randomizedTrainingData = new double*[nOfExamples];
for(int i = 0; i < nOfExamples; i++)
{
  randomizedTrainingData[i] = new double[inputSize];
}

randomizedDesiredResponse = new double*[nOfExamples];
for(int i = 0; i < nOfExamples; i++)
{
  randomizedDesiredResponse[i] = new double[outputSize];
}

while(AROASE >= 0.01)
{
  //EPOCH START - RANDOMIZE TRAINING DATA

  RandomizeTrainingData(trainingData, randomizedTrainingData, desiredResponse, randomizedDesiredResponse, nOfExamples);
  pASE = ASE;
  ASE = 0;

  for(int i = 0; i < nOfExamples; i++)
  {
    Application->ProcessMessages();

    //FORWARD COMPUTATION
    for(int j = 0; j < inputSize; j++)
    {
      nodes[0][j].response = randomizedTrainingData[i][j];
    }

    for(int j = 0; j < hiddenLayers; j++)
    {
      for(int k = 0; k < neuronsPerHiddenLayer[j]; k++)
      {
        nodes[j + 1][k].inducedLocalField = 0;

        for(list<TSynapse>::iterator l = nodes[j + 1][k].synapses.begin(); l != nodes[j + 1][k].synapses.end(); l++)
        {
          nodes[j + 1][k].inducedLocalField += ((*l).synapticWeight) * ((*l).inputNode->response);
        }

        nodes[j + 1][k].response = nodes[j + 1][k].activationFunction(nodes[j + 1][k].inducedLocalField);
      }
    }

    for(int j = 0; j < outputSize; j++)
    {
      nodes[hiddenLayers + 1][j].inducedLocalField = 0;

      for(list<TSynapse>::iterator k = nodes[hiddenLayers + 1][j].synapses.begin(); k != nodes[hiddenLayers + 1][j].synapses.end(); k++)
      {
        nodes[hiddenLayers + 1][j].inducedLocalField += ((*k).synapticWeight) * ((*k).inputNode->response);
      }

      nodes[hiddenLayers + 1][j].response = nodes[hiddenLayers + 1][j].activationFunction(nodes[hiddenLayers + 1][j].inducedLocalField);
      nodes[hiddenLayers + 1][j].error = randomizedDesiredResponse[i][j] - nodes[hiddenLayers + 1][j].response;
    }

    //BACKWARD COMPUTATION

    for(int j = 0; j < neuronsPerHiddenLayer[hiddenLayers - 1]; j++)
    {
```

```
            nodes[hiddenLayers][j].SUMDjW = 0;
          }
          for(int j = 0; j < outputSize; j++)
          {
            nodes[hiddenLayers + 1][j].Dj = nodes[hiddenLayers + 1][j].error * nodes[hiddenLayers +
1][j].activationFunctionDerivative(nodes[hiddenLayers + 1][j].response);
            for(list<TSynapse>::iterator k = nodes[hiddenLayers + 1][j].synapses.begin(); k!= nodes[hiddenLayers + 1][j].synapses.end(); k++)
            {
              tempDouble = (*k).synapticWeight;
              (*k).synapticWeight = (*k).synapticWeight + (nodes[hiddenLayers + 1][j].a * ((*k).synapticWeightDif)) + (nodes[hiddenLayers + 1][j].n
* nodes[hiddenLayers + 1][j].Dj * ((*k).inputNode->response));
              (*k).synapticWeightDif = (*k).synapticWeight - tempDouble;
              (*k).inputNode->SUMDjW += (*k).synapticWeight * nodes[hiddenLayers + 1][j].Dj;
            }
          }

          for(int j = hiddenLayers; j > 0; j--)
          {
            if(j > 1)
            {
              for(int k = 0; k < neuronsPerHiddenLayer[j - 2]; k++)
              {
                nodes[j - 1][k].SUMDjW = 0;
              }
            }
            else
            {
              for(int k = 0; k < inputSize; k++)
              {
                nodes[j - 1][k].SUMDjW = 0;
              }
            }

            for(int k = 0; k < neuronsPerHiddenLayer[j - 1]; k++)
            {
              nodes[j][k].Dj = nodes[j][k].SUMDjW * nodes[j][k].activationFunctionDerivative(nodes[j][k].response);

              for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l!= nodes[j][k].synapses.end(); l++)
              {
                tempDouble = (*l).synapticWeight;
                (*l).synapticWeight = (*l).synapticWeight + (nodes[j][k].a * ((*l).synapticWeightDif)) + (nodes[j][k].n * nodes[j][k].Dj *
((*l).inputNode->response));
                (*l).synapticWeightDif = (*l).synapticWeight - tempDouble;
                (*l).inputNode->SUMDjW += (*l).synapticWeight * nodes[j][k].Dj;
              }
            }
          }

          //CALCULATE AVERAGE SQUARED ERROR

          for(int j = 0; j < outputSize; j++)
          {
            ASE += pow(nodes[hiddenLayers + 1][j].error, double(2))/double(2);
          }

          if(*stop)
          {
            if(trainingData)
            {
              for(int i = 0; i < nOfExamples; i++)
              {
                delete [] trainingData[i];
              }
              delete [] trainingData;
            }

            if(desiredResponse)
```

```
        {
          for(int i = 0; i < nOfExamples; i++)
          {
            delete [] desiredResponse[i];
          }
          delete [] desiredResponse;
        }

        if(meanVector)
        {
          delete [] meanVector;
        }

        if(randomizedTrainingData)
        {
          for(int i = 0; i < nOfExamples; i++)
          {
            delete [] randomizedTrainingData[i];
          }
          delete [] randomizedTrainingData;
        }

        if(randomizedDesiredResponse)
        {
          for(int i = 0; i < nOfExamples; i++)
          {
            delete [] randomizedDesiredResponse[i];
          }
          delete [] randomizedDesiredResponse;
        }

        ASE = ASE / double(i + 1);
        if(myForm)
        {
          myForm->curAction = trainAction;
          if(DataModule1->curLanguage == Greek)
          {
            myForm->StatusBar1->Panels->Items[1]->Text = "Η εκπαίδευση του δικτύου διακόπηκε.  MSE: " + FloatToStrF(ASE, ffExponent, 4,
3);
          }
          else if(DataModule1->curLanguage == English)
          {
            myForm->StatusBar1->Panels->Items[1]->Text = "Network training was interrupted.  MSE: " + FloatToStrF(ASE, ffExponent, 4, 3);
          }
          myForm->StatusBar1->Invalidate();
          myForm->Timer1->Enabled = true;
        }
        if(myForm2)
        {
          DataModule1->Timer1->Enabled = true;
        }

        return true;
      }
    }

    ASE = ASE / double(nOfExamples);
    if(numberOfEpochs > 0) AROASE = double(100) * (fabs(ASE - pASE)/ASE);
    else AROASE = 1;
    numberOfEpochs++;

    if(maxNumberOfEpochs !=0)
    {
      if(numberOfEpochs > maxNumberOfEpochs) *stop = true;
    }

    if(myForm)
```

```
        {
          myForm->curAction = countAction;
          if(DataModule1->curLanguage == Greek)
          {
            myForm->StatusBar1->Panels->Items[1]->Text = "Αριθμός Εποχών: " + IntToStr(numberOfEpochs);
          }
          else if(DataModule1->curLanguage == English)
          {
            myForm->StatusBar1->Panels->Items[1]->Text = "Number Of Epochs: " + IntToStr(numberOfEpochs);
          }
          myForm->StatusBar1->Invalidate();
        }
        if(myForm2)
        {
          myForm2->curAction = countAction;
          if(DataModule1->curLanguage == Greek)
          {
            myForm2->StatusBar1->Panels->Items[1]->Text = "Αριθμός Εποχών: " + IntToStr(numberOfEpochs);
          }
          else if(DataModule1->curLanguage == English)
          {
            myForm2->StatusBar1->Panels->Items[1]->Text = "Number Of Epochs: " + IntToStr(numberOfEpochs);
          }
          myForm2->StatusBar1->Invalidate();
        }
      }

      (*MeanSquareError) = ASE;

      if(randomizedTrainingData)
      {
        for(int i = 0; i < nOfExamples; i++)
        {
          delete [] randomizedTrainingData[i];
        }
        delete [] randomizedTrainingData;
      }

      if(randomizedDesiredResponse)
      {
        for(int i = 0; i < nOfExamples; i++)
        {
          delete [] randomizedDesiredResponse[i];
        }
        delete [] randomizedDesiredResponse;
      }

      if(myForm)
      {
        myForm->curAction = trainAction;
        if(DataModule1->curLanguage == Greek)
        {
          myForm->StatusBar1->Panels->Items[1]->Text = "Η εκπαίδευση του δικτύου ολοκληρώθηκε με επιτυχία.  MSE: " + FloatToStrF(ASE,
ffExponent, 4, 3);
        }
        else if(DataModule1->curLanguage == English)
        {
          myForm->StatusBar1->Panels->Items[1]->Text = "Network training completed successfully.  MSE: " + FloatToStrF(ASE, ffExponent, 4, 3);
        }
        myForm->StatusBar1->Invalidate();
        myForm->Timer1->Enabled = true;
      }
      if(myForm2)
      {
        DataModule1->Timer1->Enabled = true;
      }
    }
```

```
else if(forecastMethod == nonAsymptotic)
{
  //NONASYMPTOTIC MODE

  rOptimal = (sqrt(double(2)*double(totalSynapses) - double(1)) - double(1))/(double(2)*(double(totalSynapses) - double(1)));
  validationSize = int(double(nOfExamples) * rOptimal);
  if(validationSize == 0) validationSize = 1;
  if(validationSize == nOfExamples) validationSize = nOfExamples - 1;
  estimationSize = nOfExamples - validationSize;

  estimationSubset = new double*[estimationSize];
  for(int i = 0; i < estimationSize; i++)
  {
    estimationSubset[i] = new double[inputSize];
  }

  estimationResponse = new double*[estimationSize];
  for(int i = 0; i < estimationSize; i++)
  {
    estimationResponse[i] = new double[outputSize];
  }

  validationSubset = new double*[validationSize];
  for(int i = 0; i < validationSize; i++)
  {
    validationSubset[i] = new double[inputSize];
  }

  validationResponse = new double*[validationSize];
  for(int i = 0; i < validationSize; i++)
  {
    validationResponse[i] = new double[outputSize];
  }

  randomizedTrainingData = new double*[estimationSize];
  for(int i = 0; i < estimationSize; i++)
  {
    randomizedTrainingData[i] = new double[inputSize];
  }

  randomizedDesiredResponse = new double*[estimationSize];
  for(int i = 0; i < estimationSize; i++)
  {
    randomizedDesiredResponse[i] = new double[outputSize];
  }

  for(int i = 0; i < validationSize; i++)
  {
    for(int j = 0; j < inputSize; j++)
    {
      validationSubset[i][j] = trainingData[nOfExamples - 1 - i][j];
    }
    for(int j = 0; j < outputSize; j++)
    {
      validationResponse[i][j] = desiredResponse[nOfExamples - 1 - i][j];
    }
  }

  for(int i = 0; i < estimationSize; i++)
  {
    for(int j = 0; j < inputSize; j++)
    {
      estimationSubset[i][j] = trainingData[i][j];
    }
    for(int j = 0; j < outputSize; j++)
    {
      estimationResponse[i][j] = desiredResponse[i][j];
```

```
    }
  }

  pMSE = -1;
  MSE = -2;
  numberOfEpochs = 1;
  period = 20;

  while(MSE <= pMSE)
  {
    //EPOCH START - RANDOMIZE TRAINING DATA

    RandomizeTrainingData(estimationSubset, randomizedTrainingData, estimationResponse, randomizedDesiredResponse, estimationSize);
    pMSE = MSE;

    for(int i = 0; i < estimationSize; i++)
    {
      Application->ProcessMessages();

      if(*stop)
      {
        if(trainingData)
        {
          for(int i = 0; i < nOfExamples; i++)
          {
            delete [] trainingData[i];
          }
          delete [] trainingData;
        }

        if(desiredResponse)
        {
          for(int i = 0; i < nOfExamples; i++)
          {
            delete [] desiredResponse[i];
          }
          delete [] desiredResponse;
        }

        if(meanVector)
        {
          delete [] meanVector;
        }

        if(estimationSubset)
        {
          for(int i = 0; i < estimationSize; i++)
          {
            delete [] estimationSubset[i];
          }
          delete [] estimationSubset;
        }

        if(estimationResponse)
        {
          for(int i = 0; i < estimationSize; i++)
          {
            delete [] estimationResponse[i];
          }
          delete [] estimationResponse;
        }

        if(validationSubset)
        {
          for(int i = 0; i < validationSize; i++)
          {
            delete [] validationSubset[i];
```

```
      }
      delete [] validationSubset;
    }

    if(validationResponse)
    {
      for(int i = 0; i < validationSize; i++)
      {
        delete [] validationResponse[i];
      }
      delete [] validationResponse;
    }

    if(randomizedTrainingData)
    {
      for(int i = 0; i < estimationSize; i++)
      {
        delete [] randomizedTrainingData[i];
      }
      delete [] randomizedTrainingData;
    }

    if(randomizedDesiredResponse)
    {
      for(int i = 0; i < estimationSize; i++)
      {
        delete [] randomizedDesiredResponse[i];
      }
      delete [] randomizedDesiredResponse;
    }

    if(numberOfEpochs > period)
    {
      if(myForm)
      {
        myForm->curAction = trainAction;
        if(DataModule1->curLanguage == Greek)
        {
          myForm->StatusBar1->Panels->Items[1]->Text = "Η εκπαίδευση του δικτύου διακόπηκε.  MSE: " + FloatToStrF(pMSE, ffExponent,
4, 3);
        }
        else if(DataModule1->curLanguage == English)
        {
          myForm->StatusBar1->Panels->Items[1]->Text = "Network training was interrupted.  MSE: " + FloatToStrF(pMSE, ffExponent, 4,
3);
        }
        myForm->StatusBar1->Invalidate();
        myForm->Timer1->Enabled = true;
      }
      if(myForm2)
      {
        DataModule1->Timer1->Enabled = true;
      }

      return true;
    }
    else return false;
  }

  //FORWARD COMPUTATION
  for(int j = 0; j < inputSize; j++)
  {
    nodes[0][j].response = randomizedTrainingData[i][j];
  }

  for(int j = 0; j < hiddenLayers; j++)
  {
```

```
            for(int k = 0; k < neuronsPerHiddenLayer[j]; k++)
            {
              nodes[j + 1][k].inducedLocalField = 0;

              for(list<TSynapse>::iterator l = nodes[j + 1][k].synapses.begin(); l != nodes[j + 1][k].synapses.end(); l++)
              {
                nodes[j + 1][k].inducedLocalField += ((*l).synapticWeight) * ((*l).inputNode->response);
              }

              nodes[j + 1][k].response = nodes[j + 1][k].activationFunction(nodes[j + 1][k].inducedLocalField);
            }
          }

          for(int j = 0; j < outputSize; j++)
          {
            nodes[hiddenLayers + 1][j].inducedLocalField = 0;

            for(list<TSynapse>::iterator k = nodes[hiddenLayers + 1][j].synapses.begin(); k != nodes[hiddenLayers + 1][j].synapses.end(); k++)
            {
              nodes[hiddenLayers + 1][j].inducedLocalField += ((*k).synapticWeight) * ((*k).inputNode->response);
            }

            nodes[hiddenLayers + 1][j].response = nodes[hiddenLayers + 1][j].activationFunction(nodes[hiddenLayers + 1][j].inducedLocalField);
            nodes[hiddenLayers + 1][j].error = randomizedDesiredResponse[i][j] - nodes[hiddenLayers + 1][j].response;
          }

          //BACKWARD COMPUTATION

          for(int j = 0; j < neuronsPerHiddenLayer[hiddenLayers - 1]; j++)
          {
            nodes[hiddenLayers][j].SUMDjW = 0;
          }
          for(int j = 0; j < outputSize; j++)
          {
            nodes[hiddenLayers + 1][j].Dj = nodes[hiddenLayers + 1][j].error * nodes[hiddenLayers +
1][j].activationFunctionDerivative(nodes[hiddenLayers + 1][j].response);
            for(list<TSynapse>::iterator k = nodes[hiddenLayers + 1][j].synapses.begin(); k!= nodes[hiddenLayers + 1][j].synapses.end(); k++)
            {
              tempDouble = (*k).synapticWeight;
              (*k).synapticWeight = (*k).synapticWeight + (nodes[hiddenLayers + 1][j].a * ((*k).synapticWeightDif)) + (nodes[hiddenLayers + 1][j].n
* nodes[hiddenLayers + 1][j].Dj * ((*k).inputNode->response));
              (*k).synapticWeightDif = (*k).synapticWeight - tempDouble;
              (*k).inputNode->SUMDjW += (*k).synapticWeight * nodes[hiddenLayers + 1][j].Dj;
            }
          }

          for(int j = hiddenLayers; j > 0; j--)
          {
            if(j > 1)
            {
              for(int k = 0; k < neuronsPerHiddenLayer[j - 2]; k++)
              {
                nodes[j - 1][k].SUMDjW = 0;
              }
            }
            else
            {
              for(int k = 0; k < inputSize; k++)
              {
                nodes[j - 1][k].SUMDjW = 0;
              }
            }

            for(int k = 0; k < neuronsPerHiddenLayer[j - 1]; k++)
            {
              nodes[j][k].Dj = nodes[j][k].SUMDjW * nodes[j][k].activationFunctionDerivative(nodes[j][k].response);
```

```
            for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l!= nodes[j][k].synapses.end(); l++)
            {
              tempDouble = (*l).synapticWeight;
              (*l).synapticWeight = (*l).synapticWeight + (nodes[j][k].a * ((*l).synapticWeightDif)) + (nodes[j][k].n * nodes[j][k].Dj *
((*l).inputNode->response));
              (*l).synapticWeightDif = (*l).synapticWeight - tempDouble;
              (*l).inputNode->SUMDjW += (*l).synapticWeight * nodes[j][k].Dj;
            }
          }
        }
      }

      //PRESENT VALIDATION SUBSET EVERY PERIOD OF EPOCHS

      if((numberOfEpochs % period) == 0)
      {
        MSE = 0;

        for(int i = 0; i < validationSize; i++)
        {
          Application->ProcessMessages();

          //FORWARD COMPUTATION
          for(int j = 0; j < inputSize; j++)
          {
            nodes[0][j].response = validationSubset[i][j];
          }

          for(int j = 0; j < hiddenLayers; j++)
          {
            for(int k = 0; k < neuronsPerHiddenLayer[j]; k++)
            {
              nodes[j + 1][k].inducedLocalField = 0;

              for(list<TSynapse>::iterator l = nodes[j + 1][k].synapses.begin(); l != nodes[j + 1][k].synapses.end(); l++)
              {
                nodes[j + 1][k].inducedLocalField += ((*l).synapticWeight) * ((*l).inputNode->response);
              }

              nodes[j + 1][k].response = nodes[j + 1][k].activationFunction(nodes[j + 1][k].inducedLocalField);
            }
          }

          for(int j = 0; j < outputSize; j++)
          {
            nodes[hiddenLayers + 1][j].inducedLocalField = 0;

            for(list<TSynapse>::iterator k = nodes[hiddenLayers + 1][j].synapses.begin(); k != nodes[hiddenLayers + 1][j].synapses.end(); k++)
            {
              nodes[hiddenLayers + 1][j].inducedLocalField += ((*k).synapticWeight) * ((*k).inputNode->response);
            }

            nodes[hiddenLayers + 1][j].response = nodes[hiddenLayers + 1][j].activationFunction(nodes[hiddenLayers + 1][j].inducedLocalField);
            nodes[hiddenLayers + 1][j].error = validationResponse[i][j] - nodes[hiddenLayers + 1][j].response;
          }

          //CALCULATE MEAN SQUARED ERROR OVER VALIDATION SUBSET

          for(int j = 0; j < outputSize; j++)
          {
            MSE += pow(nodes[hiddenLayers + 1][j].error, double(2))/double(2);
          }
        }

        MSE = MSE / double(validationSize);
      }
```

```
    if(numberOfEpochs == period) pMSE = MSE;

    numberOfEpochs++;

    if(maxNumberOfEpochs !=0)
    {
      if(numberOfEpochs > maxNumberOfEpochs) *stop = true;
    }

    if(myForm)
    {
      myForm->curAction = countAction;
      if(DataModule1->curLanguage == Greek)
      {
        myForm->StatusBar1->Panels->Items[1]->Text = "Αριθμός Εποχών: " + IntToStr(numberOfEpochs);
      }
      else if(DataModule1->curLanguage == English)
      {
        myForm->StatusBar1->Panels->Items[1]->Text = "Number Of Epochs: " + IntToStr(numberOfEpochs);
      }
      myForm->StatusBar1->Invalidate();
    }
    if(myForm2)
    {
      myForm2->curAction = countAction;
      if(DataModule1->curLanguage == Greek)
      {
        myForm2->StatusBar1->Panels->Items[1]->Text = "Αριθμός Εποχών: " + IntToStr(numberOfEpochs);
      }
      else if(DataModule1->curLanguage == English)
      {
        myForm2->StatusBar1->Panels->Items[1]->Text = "Number Of Epochs: " + IntToStr(numberOfEpochs);
      }
      myForm2->StatusBar1->Invalidate();
    }
  }

  (*MeanSquareError) = MSE;

  if(estimationSubset)
  {
    for(int i = 0; i < estimationSize; i++)
    {
      delete [] estimationSubset[i];
    }
    delete [] estimationSubset;
  }

  if(estimationResponse)
  {
    for(int i = 0; i < estimationSize; i++)
    {
      delete [] estimationResponse[i];
    }
    delete [] estimationResponse;
  }

  if(validationSubset)
  {
    for(int i = 0; i < validationSize; i++)
    {
      delete [] validationSubset[i];
    }
    delete [] validationSubset;
  }

  if(validationResponse)
```

```
      {
        for(int i = 0; i < validationSize; i++)
        {
          delete [] validationResponse[i];
        }
        delete [] validationResponse;
      }

      if(randomizedTrainingData)
      {
        for(int i = 0; i < estimationSize; i++)
        {
          delete [] randomizedTrainingData[i];
        }
        delete [] randomizedTrainingData;
      }

      if(randomizedDesiredResponse)
      {
        for(int i = 0; i < estimationSize; i++)
        {
          delete [] randomizedDesiredResponse[i];
        }
        delete [] randomizedDesiredResponse;
      }

      if(myForm)
      {
        myForm->curAction = trainAction;
        if(DataModule1->curLanguage == Greek)
        {
          myForm->StatusBar1->Panels->Items[1]->Text = "Η εκπαίδευση του δικτύου ολοκληρώθηκε με επιτυχία.  MSE: " + FloatToStrF(MSE,
ffExponent, 4, 3);
        }
        else if(DataModule1->curLanguage == English)
        {
          myForm->StatusBar1->Panels->Items[1]->Text = "Network training completed successfully.  MSE: " + FloatToStrF(MSE, ffExponent, 4, 3);
        }
        myForm->StatusBar1->Invalidate();
        myForm->Timer1->Enabled = true;
      }
      if(myForm2)
      {
        DataModule1->Timer1->Enabled = true;
      }
    }
    else if(forecastMethod == leaveOneOut)
    {
      //LEAVE-ONE-OUT METHOD

      validationSize = 1;
      estimationSize = nOfExamples - validationSize;

      estimationSubset = new double*[estimationSize];
      for(int i = 0; i < estimationSize; i++)
      {
        estimationSubset[i] = new double[inputSize];
      }

      estimationResponse = new double*[estimationSize];
      for(int i = 0; i < estimationSize; i++)
      {
        estimationResponse[i] = new double[outputSize];
      }

      validationSubset = new double*[validationSize];
      for(int i = 0; i < validationSize; i++)
```

```
{
  validationSubset[i] = new double[inputSize];
}

validationResponse = new double*[validationSize];
for(int i = 0; i < validationSize; i++)
{
  validationResponse[i] = new double[outputSize];
}

randomizedTrainingData = new double*[estimationSize];
for(int i = 0; i < estimationSize; i++)
{
  randomizedTrainingData[i] = new double[inputSize];
}

randomizedDesiredResponse = new double*[estimationSize];
for(int i = 0; i < estimationSize; i++)
{
  randomizedDesiredResponse[i] = new double[outputSize];
}

pMSE = -1;
MSE = -2;
numberOfEpochs = 1;
period = 2;

while(MSE <= pMSE)
{
  //EPOCH START - RANDOMIZE TRAINING DATA

  pMSE = MSE;
  if((numberOfEpochs % period) == 0)
  {
    MSE = 0;
  }

  for(int z = 0; z < nOfExamples; z++)
  {
    if(*stop)
    {
      if(trainingData)
      {
        for(int i = 0; i < nOfExamples; i++)
        {
          delete [] trainingData[i];
        }
        delete [] trainingData;
      }

      if(desiredResponse)
      {
        for(int i = 0; i < nOfExamples; i++)
        {
          delete [] desiredResponse[i];
        }
        delete [] desiredResponse;
      }

      if(meanVector)
      {
        delete [] meanVector;
      }

      if(estimationSubset)
      {
        for(int i = 0; i < estimationSize; i++)
```

```
    {
      delete [] estimationSubset[i];
    }
    delete [] estimationSubset;
  }

  if(estimationResponse)
  {
    for(int i = 0; i < estimationSize; i++)
    {
      delete [] estimationResponse[i];
    }
    delete [] estimationResponse;
  }

  if(validationSubset)
  {
    for(int i = 0; i < validationSize; i++)
    {
      delete [] validationSubset[i];
    }
    delete [] validationSubset;
  }

  if(validationResponse)
  {
    for(int i = 0; i < validationSize; i++)
    {
      delete [] validationResponse[i];
    }
    delete [] validationResponse;
  }

  if(randomizedTrainingData)
  {
    for(int i = 0; i < estimationSize; i++)
    {
      delete [] randomizedTrainingData[i];
    }
    delete [] randomizedTrainingData;
  }

  if(randomizedDesiredResponse)
  {
    for(int i = 0; i < estimationSize; i++)
    {
      delete [] randomizedDesiredResponse[i];
    }
    delete [] randomizedDesiredResponse;
  }

  if(numberOfEpochs > period)
  {
    if(myForm)
    {
      myForm->curAction = trainAction;
      if(DataModule1->curLanguage == Greek)
      {
        myForm->StatusBar1->Panels->Items[1]->Text = "Η εκπαίδευση του δικτύου διακόπηκε.  pMSE: " + FloatToStrF(MSE, ffExponent,
4, 3);
      }
      else if(DataModule1->curLanguage == English)
      {
        myForm->StatusBar1->Panels->Items[1]->Text = "Network training was interrupted.  pMSE: " + FloatToStrF(MSE, ffExponent, 4,
3);
      }
      myForm->StatusBar1->Invalidate();
```

```
        myForm->Timer1->Enabled = true;
      }
      if(myForm2)
      {
        DataModule1->Timer1->Enabled = true;
      }

      return true;
    }
    else return false;
  }

  for(int i = 0; i < validationSize; i++)
  {
    for(int j = 0; j < inputSize; j++)
    {
      validationSubset[i][j] = trainingData[nOfExamples - 1 - z][j];
    }
    for(int j = 0; j < outputSize; j++)
    {
      validationResponse[i][j] = desiredResponse[nOfExamples - 1 - z][j];
    }
  }

  for(int i = 0; i < estimationSize; i++)
  {
    for(int j = 0; j < inputSize; j++)
    {
      if(i < (nOfExamples - 1 - z)) estimationSubset[i][j] = trainingData[i][j];
      else estimationSubset[i][j] = trainingData[i + 1][j];
    }
    for(int j = 0; j < outputSize; j++)
    {
      if(i < (nOfExamples - 1 - z)) estimationResponse[i][j] = desiredResponse[i][j];
      else estimationResponse[i][j] = desiredResponse[i + 1][j];
    }
  }

  RandomizeTrainingData(estimationSubset, randomizedTrainingData, estimationResponse, randomizedDesiredResponse, estimationSize);

  for(int i = 0; i < estimationSize; i++)
  {
    Application->ProcessMessages();

    //FORWARD COMPUTATION
    for(int j = 0; j < inputSize; j++)
    {
      nodes[0][j].response = randomizedTrainingData[i][j];
    }

    for(int j = 0; j < hiddenLayers; j++)
    {
      for(int k = 0; k < neuronsPerHiddenLayer[j]; k++)
      {
        nodes[j + 1][k].inducedLocalField = 0;

        for(list<TSynapse>::iterator l = nodes[j + 1][k].synapses.begin(); l != nodes[j + 1][k].synapses.end(); l++)
        {
          nodes[j + 1][k].inducedLocalField += ((*l).synapticWeight) * ((*l).inputNode->response);
        }

        nodes[j + 1][k].response = nodes[j + 1][k].activationFunction(nodes[j + 1][k].inducedLocalField);
      }
    }

    for(int j = 0; j < outputSize; j++)
    {
```

```
            nodes[hiddenLayers + 1][j].inducedLocalField = 0;

            for(list<TSynapse>::iterator k = nodes[hiddenLayers + 1][j].synapses.begin(); k != nodes[hiddenLayers + 1][j].synapses.end(); k++)
            {
              nodes[hiddenLayers + 1][j].inducedLocalField += ((*k).synapticWeight) * ((*k).inputNode->response);
            }

            nodes[hiddenLayers + 1][j].response = nodes[hiddenLayers + 1][j].activationFunction(nodes[hiddenLayers + 1][j].inducedLocalField);
            nodes[hiddenLayers + 1][j].error = randomizedDesiredResponse[i][j] - nodes[hiddenLayers + 1][j].response;
          }

          //BACKWARD COMPUTATION

          for(int j = 0; j < neuronsPerHiddenLayer[hiddenLayers - 1]; j++)
          {
            nodes[hiddenLayers][j].SUMDjW = 0;
          }
          for(int j = 0; j < outputSize; j++)
          {
            nodes[hiddenLayers + 1][j].Dj = nodes[hiddenLayers + 1][j].error * nodes[hiddenLayers +
1][j].activationFunctionDerivative(nodes[hiddenLayers + 1][j].response);
            for(list<TSynapse>::iterator k = nodes[hiddenLayers + 1][j].synapses.begin(); k!= nodes[hiddenLayers + 1][j].synapses.end(); k++)
            {
              tempDouble = (*k).synapticWeight;
              (*k).synapticWeight = (*k).synapticWeight + (nodes[hiddenLayers + 1][j].a * ((*k).synapticWeightDif)) + (nodes[hiddenLayers +
1][j].n * nodes[hiddenLayers + 1][j].Dj * ((*k).inputNode->response));
              (*k).synapticWeightDif = (*k).synapticWeight - tempDouble;
              (*k).inputNode->SUMDjW += (*k).synapticWeight * nodes[hiddenLayers + 1][j].Dj;
            }
          }

          for(int j = hiddenLayers; j > 0; j--)
          {
            if(j > 1)
            {
              for(int k = 0; k < neuronsPerHiddenLayer[j - 2]; k++)
              {
                nodes[j - 1][k].SUMDjW = 0;
              }
            }
            else
            {
              for(int k = 0; k < inputSize; k++)
              {
                nodes[j - 1][k].SUMDjW = 0;
              }
            }

            for(int k = 0; k < neuronsPerHiddenLayer[j - 1]; k++)
            {
              nodes[j][k].Dj = nodes[j][k].SUMDjW * nodes[j][k].activationFunctionDerivative(nodes[j][k].response);

              for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l!= nodes[j][k].synapses.end(); l++)
              {
                tempDouble = (*l).synapticWeight;
                (*l).synapticWeight = (*l).synapticWeight + (nodes[j][k].a * ((*l).synapticWeightDif)) + (nodes[j][k].n * nodes[j][k].Dj *
((*l).inputNode->response));
                (*l).synapticWeightDif = (*l).synapticWeight - tempDouble;
                (*l).inputNode->SUMDjW += (*l).synapticWeight * nodes[j][k].Dj;
              }
            }
          }
        }

        //PRESENT VALIDATION SUBSET

        if((numberOfEpochs % period) == 0)
```

```
  {
    for(int i = 0; i < validationSize; i++)
    {
      Application->ProcessMessages();

      //FORWARD COMPUTATION
      for(int j = 0; j < inputSize; j++)
      {
        nodes[0][j].response = validationSubset[i][j];
      }

      for(int j = 0; j < hiddenLayers; j++)
      {
        for(int k = 0; k < neuronsPerHiddenLayer[j]; k++)
        {
          nodes[j + 1][k].inducedLocalField = 0;

          for(list<TSynapse>::iterator l = nodes[j + 1][k].synapses.begin(); l != nodes[j + 1][k].synapses.end(); l++)
          {
            nodes[j + 1][k].inducedLocalField += ((*l).synapticWeight) * ((*l).inputNode->response);
          }

          nodes[j + 1][k].response = nodes[j + 1][k].activationFunction(nodes[j + 1][k].inducedLocalField);
        }
      }

      for(int j = 0; j < outputSize; j++)
      {
        nodes[hiddenLayers + 1][j].inducedLocalField = 0;

        for(list<TSynapse>::iterator k = nodes[hiddenLayers + 1][j].synapses.begin(); k != nodes[hiddenLayers + 1][j].synapses.end(); k++)
        {
          nodes[hiddenLayers + 1][j].inducedLocalField += ((*k).synapticWeight) * ((*k).inputNode->response);
        }

        nodes[hiddenLayers + 1][j].response = nodes[hiddenLayers + 1][j].activationFunction(nodes[hiddenLayers + 1][j].inducedLocalField);
        nodes[hiddenLayers + 1][j].error = validationResponse[i][j] - nodes[hiddenLayers + 1][j].response;
      }

      //CALCULATE MEAN SQUARED ERROR OVER VALIDATION SUBSET

      for(int j = 0; j < outputSize; j++)
      {
        MSE += pow(nodes[hiddenLayers + 1][j].error, double(2))/double(2);
      }
    }
  }
}

if((numberOfEpochs % period) == 0)
{
  MSE = MSE / double(nOfExamples);
}
if(numberOfEpochs == period) pMSE = MSE;

numberOfEpochs++;

if(maxNumberOfEpochs !=0)
{
  if(numberOfEpochs > maxNumberOfEpochs) *stop = true;
}

if(myForm)
{
  myForm->curAction = countAction;
  if(DataModule1->curLanguage == Greek)
  {
```

```
        myForm->StatusBar1->Panels->Items[1]->Text = "Αριθμός Εποχών: " + IntToStr(numberOfEpochs);
      }
      else if(DataModule1->curLanguage == English)
      {
        myForm->StatusBar1->Panels->Items[1]->Text = "Number Of Epochs: " + IntToStr(numberOfEpochs);
      }
      myForm->StatusBar1->Invalidate();
    }
    if(myForm2)
    {
      myForm2->curAction = countAction;
      if(DataModule1->curLanguage == Greek)
      {
        myForm2->StatusBar1->Panels->Items[1]->Text = "Αριθμός Εποχών: " + IntToStr(numberOfEpochs);
      }
      else if(DataModule1->curLanguage == English)
      {
        myForm2->StatusBar1->Panels->Items[1]->Text = "Number Of Epochs: " + IntToStr(numberOfEpochs);
      }
      myForm2->StatusBar1->Invalidate();
    }
  }

  (*MeanSquareError) = MSE;

  if(estimationSubset)
  {
    for(int i = 0; i < estimationSize; i++)
    {
      delete [] estimationSubset[i];
    }
    delete [] estimationSubset;
  }

  if(estimationResponse)
  {
    for(int i = 0; i < estimationSize; i++)
    {
      delete [] estimationResponse[i];
    }
    delete [] estimationResponse;
  }

  if(validationSubset)
  {
    for(int i = 0; i < validationSize; i++)
    {
      delete [] validationSubset[i];
    }
    delete [] validationSubset;
  }

  if(validationResponse)
  {
    for(int i = 0; i < validationSize; i++)
    {
      delete [] validationResponse[i];
    }
    delete [] validationResponse;
  }

  if(randomizedTrainingData)
  {
    for(int i = 0; i < estimationSize; i++)
    {
      delete [] randomizedTrainingData[i];
    }
```

```
        delete [] randomizedTrainingData;
      }

      if(randomizedDesiredResponse)
      {
        for(int i = 0; i < estimationSize; i++)
        {
          delete [] randomizedDesiredResponse[i];
        }
        delete [] randomizedDesiredResponse;
      }

      if(myForm)
      {
        myForm->curAction = trainAction;
        if(DataModule1->curLanguage == Greek)
        {
          myForm->StatusBar1->Panels->Items[1]->Text = "Η εκπαίδευση του δικτύου ολοκληρώθηκε με επιτυχία.  MSE: " + FloatToStrF(MSE,
ffExponent, 4, 3);
        }
        else if(DataModule1->curLanguage == English)
        {
          myForm->StatusBar1->Panels->Items[1]->Text = "Network training completed successfully.  MSE: " + FloatToStrF(MSE, ffExponent, 4, 3);
        }
        myForm->StatusBar1->Invalidate();
        myForm->Timer1->Enabled = true;
      }
      if(myForm2)
      {
        DataModule1->Timer1->Enabled = true;
      }
    }

    //FREE MEMORY

    if(trainingData)
    {
      for(int i = 0; i < nOfExamples; i++)
      {
        delete [] trainingData[i];
      }
      delete [] trainingData;
    }

    if(desiredResponse)
    {
      for(int i = 0; i < nOfExamples; i++)
      {
        delete [] desiredResponse[i];
      }
      delete [] desiredResponse;
    }

    if(meanVector)
    {
      delete [] meanVector;
    }

    return true;
  }
  catch(Exception &e)
  {
    ShowMessage(e.Message);
    ShowMessage("An unexpected error occured on TrainANN function in TANeuralNetwork Unit.");
    return false;
  }
}
```

```
//---------------------------------------------------------------------------
void __fastcall TANeuralNetwork::PruneANN(double **tData, double **dResponse, int nOfExamples, bool *stop)
{
  try
  {
    int totalSynapsesWithBias, counter, counter2, synapseToDelete, totalDeletedSynapses;
    long double saliency, rOptimal, tempDouble, MSE, comparator;
    bool *deletedSynapses, synapseDeleted, cannotDeleteMore, foundSynapseToDelete;
    bool firstIteration, finishedDeleting, deleteNodeSynapses;
    int estimationSize, totalSynapses, tempInt;
    long double **trainingData, **desiredResponse, **partialDerivativeVector, **inverseHessianMatrix, **resA, *meanVector;
    long double **tempHessian, **tempVector, **tempVectorT, **vectorT;

    //CALCULATE TOTAL SYNAPSES OF NETWORK

    estimationSize = nOfExamples;
    totalSynapsesWithBias = 0;
    totalSynapses = 0;
    totalDeletedSynapses = 0;
    MSE = 0;

    ANNForm->Timer1->Enabled = false;
    ANNForm->curAction = buildAction;
    if(DataModule1->curLanguage == Greek)
    {
      ANNForm->StatusBar1->Panels->Items[1]->Text = "Έναρξη διαδικασίας περικοπής συνάψεων.";
    }
    else if(DataModule1->curLanguage == English)
    {
      ANNForm->StatusBar1->Panels->Items[1]->Text = "Network pruning initiation.";
    }
    ANNForm->StatusBar1->Invalidate();

    for(int i = 0; i < hiddenLayers + 1; i++)
    {
      Application->ProcessMessages();
      if(i != hiddenLayers)
      {
        for(int j = 0; j < neuronsPerHiddenLayer[i]; j++)
        {
          totalSynapsesWithBias += nodes[i + 1][j].synapses.size();
          totalSynapses += nodes[i + 1][j].synapses.size() - 1;
        }
      }
      else
      {
        for(int j = 0; j < outputSize; j++)
        {
          totalSynapsesWithBias += nodes[i + 1][j].synapses.size();
          totalSynapses += nodes[i + 1][j].synapses.size() - 1;
        }
      }
    }

    if(forecastMethod == nonAsymptotic)
    {
      //NONASYMPTOTIC MODE

      rOptimal = ( sqrt(((long double)(2)) * ((long double)(totalSynapsesWithBias)) - ((long double)(1)) ) - ((long double)(1)) ) / ( ((long double)(2)) *
( ((long double)(totalSynapsesWithBias)) - ((long double)(1)) ) );
      estimationSize = int(((long double)(nOfExamples)) * (((long double)(1)) - rOptimal));
      if(estimationSize == nOfExamples) estimationSize = nOfExamples - 1;
      if(estimationSize == 0) estimationSize = 1;
    }

    if(forecastMethod == leaveOneOut)
    {
```

```
    comparator = 1000000;
  }
  else
  {
    comparator = 1000;
  }

  if(*stop)
  {
    return;
  }

  trainingData = new long double*[nOfExamples];
  for(int i = 0; i < nOfExamples; i++)
  {
    trainingData[i] = new long double[inputSize];
  }

  desiredResponse = new long double*[nOfExamples];
  for(int i = 0; i < nOfExamples; i++)
  {
    desiredResponse[i] = new long double[outputSize];
  }

  partialDerivativeVector = new long double*[totalSynapses];
  for(int i = 0; i < totalSynapses; i++)
  {
    partialDerivativeVector[i] = new long double[1];
  }

  inverseHessianMatrix = new long double*[totalSynapses];
  for(int i = 0; i < totalSynapses; i++)
  {
    inverseHessianMatrix[i] = new long double[totalSynapses];
  }

  resA = new long double*[1];
  resA[0] = new long double[1];

  meanVector = new long double[inputSize];

  for(int i = 0; i < inputSize; i++)
  {
    meanVector[i] = 0;
  }

  for(int i = 0; i < nOfExamples; i++)
  {
    Application->ProcessMessages();
    for(int j = 0; j < inputSize; j++)
    {
      meanVector[j] += tData[i][j];
    }
  }

  for(int i = 0; i < inputSize; i++)
  {
    meanVector[i] = meanVector[i]/((long double)(nOfExamples));
  }

  for(int i = 0; i < nOfExamples; i++)
  {
    Application->ProcessMessages();
    for(int j = 0; j < inputSize; j++)
    {
      trainingData[i][j] = tData[i][j] - meanVector[j];
    }
```

```
}

for(int i = 0; i < inputSize; i++)
{
  minInput[i] = trainingData[0][i];
  maxInput[i] = trainingData[0][i];
}

for(int i = 1; i < nOfExamples; i++)
{
  for(int j = 0; j < inputSize; j++)
  {
    if(trainingData[i][j] > maxInput[j]) maxInput[j] = trainingData[i][j];
    if(trainingData[i][j] < minInput[j]) minInput[j] = trainingData[i][j];
  }
}

for(int i = 0; i < nOfExamples; i++)
{
  for(int j = 0; j < inputSize; j++)
  {
    trainingData[i][j] = double(-1) + double(2)*((trainingData[i][j] - minInput[j])/(maxInput[j] - minInput[j]));
  }
}

for(int i = 0; i < outputSize; i++)
{
  minResponse[i] = dResponse[0][i];
  maxResponse[i] = dResponse[0][i];
}

for(int i = 1; i < nOfExamples; i++)
{
  for(int j = 0; j < outputSize; j++)
  {
    if(dResponse[i][j] > maxResponse[j]) maxResponse[j] = dResponse[i][j];
    if(dResponse[i][j] < minResponse[j]) minResponse[j] = dResponse[i][j];
  }
}

for(int i = 0; i < nOfExamples; i++)
{
  for(int j = 0; j < outputSize; j++)
  {
    desiredResponse[i][j] = ((long double)(-1)) + ((long double)(2))*((dResponse[i][j] - minResponse[j])/(maxResponse[j] - minResponse[j]));
  }
}

for(int i = 0; i < totalSynapses; i++)
{
  partialDerivativeVector[i][0] = 0;
  for(int j = 0; j < totalSynapses; j++)
  {
    if(j != i) inverseHessianMatrix[i][j] = 0;
    else inverseHessianMatrix[i][j] = ((long double)(1))/((long double)(10E-200));
  }
}

if(*stop)
{
  for(int i = 0; i < nOfExamples; i++)
  {
    delete [] trainingData[i];
  }
  delete [] trainingData;

  for(int i = 0; i < nOfExamples; i++)
```

```
    {
      delete [] desiredResponse[i];
    }
    delete [] desiredResponse;

    for(int i = 0; i < totalSynapses; i++)
    {
      delete [] partialDerivativeVector[i];
    }
    delete [] partialDerivativeVector;

    for(int i = 0; i < totalSynapses; i++)
    {
      delete [] inverseHessianMatrix[i];
    }
    delete [] inverseHessianMatrix;

    delete [] resA[0];
    delete [] resA;

    delete [] meanVector;

    return;
  }

  //START OPTIMAL BRAIN SURGEON ALGORITHM

  foundSynapseToDelete = true;
  firstIteration = true;
  finishedDeleting = false;

  while((foundSynapseToDelete) && (!finishedDeleting))
  {
    ANNForm->curAction = buildAction;
    if(DataModule1->curLanguage == Greek)
    {
      ANNForm->StatusBar1->Panels->Items[1]->Text = "Υπολογισμός μερικών παραγώγων ΔF/ΔW.";
    }
    else if(DataModule1->curLanguage == English)
    {
      ANNForm->StatusBar1->Panels->Items[1]->Text = "Calculation of Partial Derivatives ΔF/ΔW.";
    }
    ANNForm->StatusBar1->Invalidate();

    ANNForm->ProgressBar1->Position = 0;
    ANNForm->Shape1->Brush->Color = clGreen;
    Application->ProcessMessages();

    tempHessian = new long double*[totalSynapses];
    for(int j = 0; j < totalSynapses; j++)
    {
      tempHessian[j] = new long double[totalSynapses];
    }

    vectorT = new long double*[1];
    vectorT[0] = new long double[totalSynapses];

    tempVector = new long double*[totalSynapses];
    for(int j = 0; j < totalSynapses; j++)
    {
      tempVector[j] = new long double[1];
    }

    tempVectorT = new long double*[1];
    tempVectorT[0] = new long double[totalSynapses];

    tempInt = totalSynapses;
```

```
//PRESENT ALL EXAMPLES

for(int i = 0; i < estimationSize; i++)
{
  Application->ProcessMessages();

  //FORWARD COMPUTATION
  for(int j = 0; j < inputSize; j++)
  {
    nodes[0][j].response = trainingData[i][j];
  }

  for(int j = 0; j < hiddenLayers; j++)
  {
    for(int k = 0; k < neuronsPerHiddenLayer[j]; k++)
    {
      nodes[j + 1][k].inducedLocalField = 0;

      for(list<TSynapse>::iterator l = nodes[j + 1][k].synapses.begin(); l != nodes[j + 1][k].synapses.end(); l++)
      {
        nodes[j + 1][k].inducedLocalField += ((*l).synapticWeight) * ((*l).inputNode->response);
      }

      nodes[j + 1][k].response = nodes[j + 1][k].activationFunction(nodes[j + 1][k].inducedLocalField);
    }
  }

  for(int j = 0; j < outputSize; j++)
  {
    nodes[hiddenLayers + 1][j].inducedLocalField = 0;

    for(list<TSynapse>::iterator k = nodes[hiddenLayers + 1][j].synapses.begin(); k != nodes[hiddenLayers + 1][j].synapses.end(); k++)
    {
      nodes[hiddenLayers + 1][j].inducedLocalField += ((*k).synapticWeight) * ((*k).inputNode->response);
    }

    nodes[hiddenLayers + 1][j].response = nodes[hiddenLayers + 1][j].activationFunction(nodes[hiddenLayers + 1][j].inducedLocalField);
    nodes[hiddenLayers + 1][j].error = desiredResponse[i][j] - nodes[hiddenLayers + 1][j].response;
  }

  if(firstIteration)
  {
    for(int j = 0; j < outputSize; j++)
    {
      MSE += pow(nodes[hiddenLayers + 1][j].error, ((long double)(2)))/((long double)(2));
    }
  }

  //COMPUTE partialDerivativeVector

  counter = 0;
  for(int j = 1; j < hiddenLayers + 2; j++)
  {
    if(*stop)
    {
      for(int i = 0; i < nOfExamples; i++)
      {
        delete [] trainingData[i];
      }
      delete [] trainingData;

      for(int i = 0; i < nOfExamples; i++)
      {
        delete [] desiredResponse[i];
      }
      delete [] desiredResponse;
```

```
            for(int i = 0; i < totalSynapses; i++)
            {
              delete [] partialDerivativeVector[i];
            }
            delete [] partialDerivativeVector;

            for(int i = 0; i < totalSynapses; i++)
            {
              delete [] inverseHessianMatrix[i];
            }
            delete [] inverseHessianMatrix;

            delete [] resA[0];
            delete [] resA;

            delete [] meanVector;

            for(int j = 0; j < totalSynapses; j++)
            {
              delete [] tempHessian[j];
            }
            delete [] tempHessian;
            delete [] tempVectorT[0];
            delete [] tempVectorT;
            delete [] vectorT[0];
            delete [] vectorT;

            for(int j = 0; j < tempInt; j++)
            {
              delete [] tempVector[j];
            }
            delete [] tempVector;

            return;
          }

          if(j == 1)
          {
            for(int k = 0; k < neuronsPerHiddenLayer[j - 1]; k++)
            {
              Application->ProcessMessages();
              for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l != nodes[j][k].synapses.end(); l++)
              {
                if((*l).inputNode != &(fixedInput))
                {
                  if(counter < totalSynapses)
                  {
                    partialDerivativeVector[counter][0] = (((long double)(1)) / sqrt(estimationSize)) *
HyperbolicTangentDerivative(nodes[hiddenLayers + 1][0].response) * HyperbolicTangentDerivative(nodes[j][k].response) * ((*l).inputNode-
>response) * SumCalculation(hiddenLayers - j, &nodes[hiddenLayers + 1][0], (*l).outputNode);
                    counter++;
                    ANNForm->ProgressBar1->Position = (500 * ( i * totalSynapses + counter)) / (estimationSize * totalSynapses);
                  }
                  else
                  {
                    ShowMessage("An unexpected error occured on PruneANN function in TANeuralNetwork Unit.\nTotal Synapses not synchronized
with count variable.");
                  }
                }
              }
            }
          }
          else if(j == (hiddenLayers + 1))
          {
            Application->ProcessMessages();
```

```
            for(int k = 0; k < outputSize; k++)
            {
              for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l != nodes[j][k].synapses.end(); l++)
              {
                if((*l).inputNode != &(fixedInput))
                {
                  if(counter < totalSynapses)
                  {
                    partialDerivativeVector[counter][0] = (((long double)(1)) / sqrt(estimationSize)) *
HyperbolicTangentDerivative(nodes[hiddenLayers + 1][0].response) * HyperbolicTangent((*l).inputNode->inducedLocalField);
                    counter++;
                    ANNForm->ProgressBar1->Position = (500 * ( i * totalSynapses + counter)) / (estimationSize * totalSynapses);
                  }
                  else
                  {
                    ShowMessage("An unexpected error occured on PruneANN function in TANeuralNetwork Unit.\nTotal Synapses not synchronized
with count variable.");
                  }
                }
              }
            }
          }
          else
          {
            for(int k = 0; k < neuronsPerHiddenLayer[j - 1]; k++)
            {
              Application->ProcessMessages();

              for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l != nodes[j][k].synapses.end(); l++)
              {
                if((*l).inputNode != &(fixedInput))
                {
                  if(counter < totalSynapses)
                  {
                    partialDerivativeVector[counter][0] = (((long double)(1)) / sqrt(estimationSize)) *
HyperbolicTangentDerivative(nodes[hiddenLayers + 1][0].response) * HyperbolicTangentDerivative(nodes[j][k].response) *
HyperbolicTangent((*l).inputNode->inducedLocalField) * SumCalculation(hiddenLayers - j, &nodes[hiddenLayers + 1][0], (*l).outputNode);
                    counter++;
                    ANNForm->ProgressBar1->Position = (500 * ( i * totalSynapses + counter)) / (estimationSize * totalSynapses);
                  }
                  else
                  {
                    ShowMessage("An unexpected error occured on PruneANN function in TANeuralNetwork Unit.\nTotal Synapses not synchronized
with count variable.");
                  }
                }
              }
            }
          }
        }

        //COMPUTE THE INVERSE HESSIAN MATRIX

        MatrixTranspoze(partialDerivativeVector, totalSynapses, 1, vectorT, 1, totalSynapses);
        MatrixMultiply(vectorT, 1, totalSynapses, inverseHessianMatrix, totalSynapses, totalSynapses, tempVectorT, 1, totalSynapses);
        MatrixMultiply(tempVectorT, 1, totalSynapses, partialDerivativeVector, totalSynapses, 1, resA, 1, 1);

        resA[0][0] += 1;
        if(resA[0][0] == 0)  resA[0][0] = 0.000000000000000001;
        for(int j = 0; j < totalSynapses; j++)
        {
          partialDerivativeVector[j][0] = partialDerivativeVector[j][0]/resA[0][0];
        }

        MatrixMultiply(vectorT, 1, totalSynapses, inverseHessianMatrix, totalSynapses, totalSynapses, tempVectorT, 1, totalSynapses);
        MatrixMultiply(inverseHessianMatrix, totalSynapses, totalSynapses, partialDerivativeVector, totalSynapses, 1, tempVector, totalSynapses, 1);
        MatrixMultiply(tempVector, totalSynapses, 1, tempVectorT, 1, totalSynapses, tempHessian, totalSynapses, totalSynapses);
```

```
      MatrixSubtract(inverseHessianMatrix, totalSynapses, totalSynapses, tempHessian, totalSynapses, totalSynapses, inverseHessianMatrix,
totalSynapses, totalSynapses);
    }

    ANNForm->Shape1->Brush->Color = clRed;

    //CALCULATE MEAN SQUARE ERROR

    if(firstIteration)
    {
      MSE = MSE / ((long double)(estimationSize));

      firstIteration = false;
    }

    Application->ProcessMessages();
    for(int j = 0; j < totalSynapses; j++)
    {
      delete [] tempHessian[j];
    }
    delete [] tempHessian;
    delete [] tempVectorT[0];
    delete [] tempVectorT;
    delete [] vectorT[0];
    delete [] vectorT;

    //STORE ALL SYNAPTIC WEIGHTS

    counter = 0;

    deletedSynapses = new bool[totalSynapses];

    for(int j = 1; j < hiddenLayers + 2; j++)
    {
      Application->ProcessMessages();

      if(*stop)
      {
        for(int i = 0; i < nOfExamples; i++)
        {
          delete [] trainingData[i];
        }
        delete [] trainingData;

        for(int i = 0; i < nOfExamples; i++)
        {
          delete [] desiredResponse[i];
        }
        delete [] desiredResponse;

        for(int i = 0; i < totalSynapses; i++)
        {
          delete [] partialDerivativeVector[i];
        }
        delete [] partialDerivativeVector;

        for(int i = 0; i < totalSynapses; i++)
        {
          delete [] inverseHessianMatrix[i];
        }
        delete [] inverseHessianMatrix;

        delete [] resA[0];
        delete [] resA;

        delete [] meanVector;
```

```
    for(int j = 0; j < tempInt; j++)
    {
      delete [] tempVector[j];
    }
    delete [] tempVector;

    delete [] deletedSynapses;

    return;
  }

  if(j != (hiddenLayers + 1))
  {
    for(int k = 0; k < neuronsPerHiddenLayer[j - 1]; k++)
    {
      for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l != nodes[j][k].synapses.end(); l++)
      {
        if((*l).inputNode != &(fixedInput))
        {
          tempVector[counter][0] = ((*l).synapticWeight);
          deletedSynapses[counter] = false;
          counter++;
        }
      }
    }
  }
  else
  {
    for(int k = 0; k < outputSize; k++)
    {
      for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l != nodes[j][k].synapses.end(); l++)
      {
        if((*l).inputNode != &(fixedInput))
        {
          tempVector[counter][0] = ((*l).synapticWeight);
          deletedSynapses[counter] = false;
          counter++;
        }
      }
    }
  }
}

//FIND THE I THAT CORRESPONDS TO THE SMALLEST SALIENCY AND DELETE RELATED SYNAPTIC WEIGHT

foundSynapseToDelete = false;

do
{
  counter = 0;
  synapseToDelete = -1;
  saliency = MSE;

  for(int j = 1; j < hiddenLayers + 2; j++)
  {
    Application->ProcessMessages();

    if(*stop)
    {
      for(int i = 0; i < nOfExamples; i++)
      {
        delete [] trainingData[i];
      }
      delete [] trainingData;

      for(int i = 0; i < nOfExamples; i++)
      {
```

```
    delete [] desiredResponse[i];
  }
  delete [] desiredResponse;

  for(int i = 0; i < totalSynapses; i++)
  {
    delete [] partialDerivativeVector[i];
  }
  delete [] partialDerivativeVector;

  for(int i = 0; i < totalSynapses; i++)
  {
    delete [] inverseHessianMatrix[i];
  }
  delete [] inverseHessianMatrix;

  delete [] resA[0];
  delete [] resA;

  delete [] meanVector;

  for(int j = 0; j < tempInt; j++)
  {
    delete [] tempVector[j];
  }
  delete [] tempVector;

  delete [] deletedSynapses;

  return;
}

if(j != (hiddenLayers + 1))
{
  for(int k = 0; k < neuronsPerHiddenLayer[j - 1]; k++)
  {
    for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l != nodes[j][k].synapses.end(); l++)
    {
      if((*l).inputNode != &(fixedInput))
      {
        if(inverseHessianMatrix[counter][counter] == 0)
        {
          counter++;
          continue;
        }
        tempDouble = pow((*l).synapticWeight, ((long double)(2))) / (((long double)(2)) * inverseHessianMatrix[counter][counter]);
        if(saliency > tempDouble)
        {
          if(tempDouble > 0)
          {
            saliency = tempDouble;
            synapseToDelete = counter;
          }
          else
          {
            finishedDeleting = true;
          }
        }
        counter++;
      }
    }
  }
}
else
{
  for(int k = 0; k < outputSize; k++)
  {
```

```
        for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l != nodes[j][k].synapses.end(); l++)
        {
          if((*l).inputNode != &(fixedInput))
          {
            if(inverseHessianMatrix[counter][counter] == 0)
            {
              counter++;
              continue;
            }
            tempDouble = pow((*l).synapticWeight, ((long double)(2))) / (((long double)(2)) * inverseHessianMatrix[counter][counter]);
            if(saliency > tempDouble)
            {
              if(tempDouble > 0)
              {
                saliency = tempDouble;
                synapseToDelete = counter;
              }
              else
              {
                finishedDeleting = true;
              }
            }
            counter++;
          }
        }
      }
    }
  }

  counter = 0;
  counter2 = 0;
  synapseDeleted = false;
  cannotDeleteMore = false;

  if((saliency < (MSE / ((long double)(comparator)))) && (!finishedDeleting))
  {
    for(int j = 1; j < hiddenLayers + 2; j++)
    {
      Application->ProcessMessages();

      if(*stop)
      {
        for(int i = 0; i < nOfExamples; i++)
        {
          delete [] trainingData[i];
        }
        delete [] trainingData;

        for(int i = 0; i < nOfExamples; i++)
        {
          delete [] desiredResponse[i];
        }
        delete [] desiredResponse;

        for(int i = 0; i < totalSynapses; i++)
        {
          delete [] partialDerivativeVector[i];
        }
        delete [] partialDerivativeVector;

        for(int i = 0; i < totalSynapses; i++)
        {
          delete [] inverseHessianMatrix[i];
        }
        delete [] inverseHessianMatrix;

        delete [] resA[0];
```

```
            delete [] resA;

            delete [] meanVector;

            for(int j = 0; j < tempInt; j++)
            {
              delete [] tempVector[j];
            }
            delete [] tempVector;

            delete [] deletedSynapses;

            return;
          }

          if(j != (hiddenLayers + 1))
          {
            for(int k = 0; k < neuronsPerHiddenLayer[j - 1]; k++)
            {
              for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l != nodes[j][k].synapses.end(); l++)
              {
                if((*l).inputNode != &(fixedInput))
                {
                  while(deletedSynapses[counter2]) counter2++;
                  if(counter != synapseToDelete)
                  {
                    counter++;
                    counter2++;
                  }
                  else
                  {
                    if(nodes[j][k].synapses.size() > 2)
                    {
                      nodes[j][k].synapses.erase(l);
                      deletedSynapses[counter2] = true;
                      synapseDeleted = true;
                      foundSynapseToDelete = true;
                      totalDeletedSynapses++;

                      ANNForm->curAction = buildAction;
                      if(DataModule1->curLanguage == Greek)
                      {
                        ANNForm->StatusBar1->Panels->Items[1]->Text = "Συνολικός αριθμός συνάψεων που περικόπηκαν : " +
IntToStr(totalDeletedSynapses);
                      }
                      else if(DataModule1->curLanguage == English)
                      {
                        ANNForm->StatusBar1->Panels->Items[1]->Text = "Total number of pruned synapses : " + IntToStr(totalDeletedSynapses);
                      }
                      ANNForm->StatusBar1->Invalidate();
                    }
                    else
                    {
                      cannotDeleteMore = true;
                    }
                  }
                }

                if(synapseDeleted) break;
              }

              if(synapseDeleted) break;
            }
          }
          else
          {
            for(int k = 0; k < outputSize; k++)
```

```
            {
              for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l != nodes[j][k].synapses.end(); l++)
              {
                if((*l).inputNode != &(fixedInput))
                {
                  while(deletedSynapses[counter2]) counter2++;
                  if(counter != synapseToDelete)
                  {
                    counter++;
                    counter2++;
                  }
                  else
                  {
                    if(nodes[j][k].synapses.size() > 2)
                    {
                      nodes[j][k].synapses.erase(l);
                      deletedSynapses[counter2] = true;
                      synapseDeleted = true;
                      foundSynapseToDelete = true;
                      totalDeletedSynapses++;

                      ANNForm->curAction = buildAction;
                      if(DataModule1->curLanguage == Greek)
                      {
                        ANNForm->StatusBar1->Panels->Items[1]->Text = "Συνολικός αριθμός συνάψεων που περικόπηκαν : " +
IntToStr(totalDeletedSynapses);
                      }
                      else if(DataModule1->curLanguage == English)
                      {
                        ANNForm->StatusBar1->Panels->Items[1]->Text = "Total number of pruned synapses : " + IntToStr(totalDeletedSynapses);
                      }
                      ANNForm->StatusBar1->Invalidate();
                    }
                    else
                    {
                      cannotDeleteMore = true;
                    }
                  }
                }

                if(synapseDeleted) break;
              }

              if(synapseDeleted) break;
            }
          }

          if(synapseDeleted) break;
        }

        //UPDATE ALL SYNAPTIC WEIGHTS TEMPORARY

        counter = 0;
        counter2 = 0;

        for(int j = 1; j < hiddenLayers + 2; j++)
        {
          Application->ProcessMessages();

          if(*stop)
          {
            for(int i = 0; i < nOfExamples; i++)
            {
              delete [] trainingData[i];
            }
            delete [] trainingData;
```

```
            for(int i = 0; i < nOfExamples; i++)
            {
              delete [] desiredResponse[i];
            }
            delete [] desiredResponse;

            for(int i = 0; i < totalSynapses; i++)
            {
              delete [] partialDerivativeVector[i];
            }
            delete [] partialDerivativeVector;

            for(int i = 0; i < totalSynapses; i++)
            {
              delete [] inverseHessianMatrix[i];
            }
            delete [] inverseHessianMatrix;

            delete [] resA[0];
            delete [] resA;

            delete [] meanVector;

            for(int j = 0; j < tempInt; j++)
            {
              delete [] tempVector[j];
            }
            delete [] tempVector;

            delete [] deletedSynapses;

            return;
          }

          if(j != (hiddenLayers + 1))
          {
            for(int k = 0; k < neuronsPerHiddenLayer[j - 1]; k++)
            {
              for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l != nodes[j][k].synapses.end(); l++)
              {
                if((*l).inputNode != &(fixedInput))
                {
                  while(deletedSynapses[counter2]) counter2++;
                  if(inverseHessianMatrix[counter][counter] == 0) inverseHessianMatrix[counter][counter] = 0.00000000000000000001;
                  tempVector[counter2][0] = ((long double)(-1)) * (inverseHessianMatrix[counter][synapseToDelete] /
inverseHessianMatrix[counter][counter]) * ((*l).synapticWeight);
                  counter++;
                  counter2++;
                }
              }
            }
          }
          else
          {
            for(int k = 0; k < outputSize; k++)
            {
              for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l != nodes[j][k].synapses.end(); l++)
              {
                if((*l).inputNode != &(fixedInput))
                {
                  while(deletedSynapses[counter2]) counter2++;
                  if(inverseHessianMatrix[counter][counter] == 0) inverseHessianMatrix[counter][counter] = 0.00000000000000000001;
                  tempVector[counter2][0] = ((long double)(-1)) * (inverseHessianMatrix[counter][synapseToDelete] /
inverseHessianMatrix[counter][counter]) * ((*l).synapticWeight);
                  counter++;
                  counter2++;
                }
```

```
        }
      }
    }
  }

  //UPDATE HESSIAN MATRIX

  if(synapseDeleted)
  {
    tempHessian = new long double*[totalSynapses];
    for(int j = 0; j < totalSynapses; j++)
    {
      tempHessian[j] = new long double[totalSynapses];
    }

    for(int j = 0; j < totalSynapses; j++)
    {
      for(int k = 0; k < totalSynapses; k++)
      {
        tempHessian[j][k] = inverseHessianMatrix[j][k];
      }
    }

    for(int j = 0; j < totalSynapses; j++)
    {
      delete [] inverseHessianMatrix[j];
      delete [] partialDerivativeVector[j];
    }
    delete [] inverseHessianMatrix;
    delete [] partialDerivativeVector;

    totalSynapses--;

    inverseHessianMatrix = new long double*[totalSynapses];
    partialDerivativeVector = new long double*[totalSynapses];
    for(int j = 0; j < totalSynapses; j++)
    {
      inverseHessianMatrix[j] = new long double[totalSynapses];
      partialDerivativeVector[j] = new long double[1];
    }

    for(int j = 0; j < totalSynapses + 1; j++)
    {
      if(j < synapseToDelete)
      {
        for(int k = 0; k < totalSynapses + 1; k++)
        {
          if(k < synapseToDelete)
          {
            inverseHessianMatrix[j][k] = tempHessian[j][k];
          }
          else if(k > synapseToDelete)
          {
            inverseHessianMatrix[j][k - 1] = tempHessian[j][k];
          }
        }
      }
      else if(j > synapseToDelete)
      {
        for(int k = 0; k < totalSynapses + 1; k++)
        {
          if(k < synapseToDelete)
          {
            inverseHessianMatrix[j - 1][k] = tempHessian[j][k];
          }
          else if(k > synapseToDelete)
          {
```

```
                inverseHessianMatrix[j - 1][k - 1] = tempHessian[j][k];
              }
            }
          }
        }

        for(int j = 0; j < totalSynapses + 1; j++)
        {
          delete [] tempHessian[j];
        }
        delete [] tempHessian;
      }
    }
  } while((saliency < (MSE / ((long double)(comparator)))) && (!cannotDeleteMore) && (!finishedDeleting));

  //UPDATE ALL SYNAPTIC WEIGHTS IN THE NETWORK

  counter = 0;

  for(int j = 1; j < hiddenLayers + 2; j++)
  {
    Application->ProcessMessages();

    if(*stop)
    {
      for(int i = 0; i < nOfExamples; i++)
      {
        delete [] trainingData[i];
      }
      delete [] trainingData;

      for(int i = 0; i < nOfExamples; i++)
      {
        delete [] desiredResponse[i];
      }
      delete [] desiredResponse;

      for(int i = 0; i < totalSynapses; i++)
      {
        delete [] partialDerivativeVector[i];
      }
      delete [] partialDerivativeVector;

      for(int i = 0; i < totalSynapses; i++)
      {
        delete [] inverseHessianMatrix[i];
      }
      delete [] inverseHessianMatrix;

      delete [] resA[0];
      delete [] resA;

      delete [] meanVector;

      for(int j = 0; j < tempInt; j++)
      {
        delete [] tempVector[j];
      }
      delete [] tempVector;

      delete [] deletedSynapses;

      return;
    }

    if(j != (hiddenLayers + 1))
    {
```

```
        for(int k = 0; k < neuronsPerHiddenLayer[j - 1]; k++)
        {
          for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l != nodes[j][k].synapses.end(); l++)
          {
            if((*l).inputNode != &(fixedInput))
            {
              while(deletedSynapses[counter]) counter++;
              (*l).synapticWeight = tempVector[counter][0];
              counter++;
            }
          }
        }
      }
      else
      {
        for(int k = 0; k < outputSize; k++)
        {
          for(list<TSynapse>::iterator l = nodes[j][k].synapses.begin(); l != nodes[j][k].synapses.end(); l++)
          {
            if((*l).inputNode != &(fixedInput))
            {
              while(deletedSynapses[counter]) counter++;
              (*l).synapticWeight = tempVector[counter][0];
              counter++;
            }
          }
        }
      }
    }

    //END OF MAIN ITERATION

    for(int j = 0; j < tempInt; j++)
    {
      delete [] tempVector[j];
    }
    delete [] tempVector;
    delete [] deletedSynapses;
  }

  //DELETE SYNAPSES THAT CANNOT REACH OUTPUTLAYER

  for(int i = 0; i < outputSize; i++)
  {
    if(nodes[hiddenLayers + 1][i].synapses.size() == 1)
    {
      totalDeletedSynapses++;
      nodes[hiddenLayers + 1][i].synapses.clear();
    }
  }

  for(int i = 0; i < neuronsPerHiddenLayer[hiddenLayers - 1]; i++)
  {
    deleteNodeSynapses = true;

    for(int j = 0; j < outputSize; j++)
    {
      for(list<TSynapse>::iterator k = nodes[hiddenLayers + 1][j].synapses.begin(); k != nodes[hiddenLayers + 1][j].synapses.end(); k++)
      {
        if((*k).inputNode == &nodes[hiddenLayers][i])
        {
          deleteNodeSynapses = false;
          break;
        }
      }

      if(!deleteNodeSynapses) break;
```

```
    }

    if(deleteNodeSynapses)
    {
      totalDeletedSynapses += nodes[hiddenLayers][i].synapses.size();
      nodes[hiddenLayers][i].synapses.clear();
    }
  }

  for(int i = hiddenLayers - 1; i > 0; i--)
  {
    for(int j = 0; j < neuronsPerHiddenLayer[i - 1]; j++)
    {
      deleteNodeSynapses = true;

      for(int k = 0; k < neuronsPerHiddenLayer[i]; k++)
      {
        for(list<TSynapse>::iterator l = nodes[i + 1][k].synapses.begin(); l != nodes[i + 1][k].synapses.end(); l++)
        {
          if((*l).inputNode == &nodes[i][j])
          {
            deleteNodeSynapses = false;
            break;
          }
        }

        if(!deleteNodeSynapses) break;
      }

      if(deleteNodeSynapses)
      {
        totalDeletedSynapses += nodes[i][j].synapses.size();
        nodes[i][j].synapses.clear();
      }
    }
  }

  ANNForm->curAction = buildAction;
  if(DataModule1->curLanguage == Greek)
  {
    ANNForm->StatusBar1->Panels->Items[1]->Text = "Συνολικά διαγράφτηκαν " + IntToStr(totalDeletedSynapses) + " σύνδεσμοι.";
  }
  else if(DataModule1->curLanguage == English)
  {
    ANNForm->StatusBar1->Panels->Items[1]->Text = "Totaly pruned synapses : " + IntToStr(totalDeletedSynapses);
  }
  ANNForm->StatusBar1->Invalidate();
  ANNForm->Timer1->Enabled = true;

  for(int i = 0; i < nOfExamples; i++)
  {
    delete [] trainingData[i];
  }
  delete [] trainingData;

  for(int i = 0; i < nOfExamples; i++)
  {
    delete [] desiredResponse[i];
  }
  delete [] desiredResponse;

  for(int i = 0; i < totalSynapses; i++)
  {
    delete [] partialDerivativeVector[i];
  }
  delete [] partialDerivativeVector;
```

```
    for(int i = 0; i < totalSynapses; i++)
    {
      delete [] inverseHessianMatrix[i];
    }
    delete [] inverseHessianMatrix;

    delete [] resA[0];
    delete [] resA;

    delete [] meanVector;
  }
  catch(Exception &e)
  {
    ShowMessage(e.Message);
    ShowMessage("An unexpected error occured on PruneANN function in TANeuralNetwork Unit.");
  }
}
//---------------------------------------------------------------------------
double __fastcall TANeuralNetwork::SumCalculation(int layersLeft, TNode *currentNode, TNode *endNode)
{
  double temp = 0;

  if(layersLeft != 0)
  {
    for(list<TSynapse>::iterator i = currentNode->synapses.begin(); i != currentNode->synapses.end(); i++)
    {
      if((*i).inputNode != &(fixedInput))
      {
        temp += ((*i).synapticWeight) * HyperbolicTangentDerivative((*i).inputNode->response) * SumCalculation(layersLeft - 1, (*i).inputNode,
endNode);
      }
    }

    return temp;
  }
  else if (layersLeft == 0)
  {
    for(list<TSynapse>::iterator i = currentNode->synapses.begin(); i != currentNode->synapses.end(); i++)
    {
      if((*i).inputNode == endNode)
      {
        return (*i).synapticWeight;
      }
    }
  }

  return 0;
}
//---------------------------------------------------------------------------
void __fastcall TANeuralNetwork::Forecast(double **tData, int nOfExamples, double *inputVector, double *outputVector)
{
  try
  {
    double *input, *ratioTranformation, *meanVector;

    //TRANSFORM INPUT VECTOR

    meanVector = new double[inputSize];
    ratioTranformation = new double[outputSize];
    input = new double[inputSize];

    for(int i = 0; i < inputSize; i++)
    {
      meanVector[i] = 0;
    }

    for(int i = 0; i < nOfExamples; i++)
```

```
{
  for(int j = 0; j < inputSize; j++)
  {
    meanVector[j] += tData[i][j];
  }
}

for(int i = 0; i < inputSize; i++)
{
  meanVector[i] = meanVector[i]/double(nOfExamples);
}

for(int i = 0; i < nOfExamples; i++)
{
  Application->ProcessMessages();
  for(int j = 0; j < inputSize; j++)
  {
    input[j] = inputVector[j] - meanVector[j];
    input[j] = double(-1) + double(2)*((input[j] - minInput[j])/(maxInput[j] - minInput[j]));
  }
}

//FORWARD COMPUTATION
for(int i = 0; i < inputSize; i++)
{
  nodes[0][i].response = input[i];
}

for(int i = 0; i < hiddenLayers; i++)
{
  for(int j = 0; j < neuronsPerHiddenLayer[i]; j++)
  {
    nodes[i + 1][j].inducedLocalField = 0;

    for(list<TSynapse>::iterator k = nodes[i + 1][j].synapses.begin(); k != nodes[i + 1][j].synapses.end(); k++)
    {
      nodes[i + 1][j].inducedLocalField += ((*k).synapticWeight) * ((*k).inputNode->response);
    }

    nodes[i + 1][j].response = nodes[i + 1][j].activationFunction(nodes[i + 1][j].inducedLocalField);
  }
}

for(int i = 0; i < outputSize; i++)
{
  nodes[hiddenLayers + 1][i].inducedLocalField = 0;

  for(list<TSynapse>::iterator j = nodes[hiddenLayers + 1][i].synapses.begin(); j != nodes[hiddenLayers + 1][i].synapses.end(); j++)
  {
    nodes[hiddenLayers + 1][i].inducedLocalField += ((*j).synapticWeight) * ((*j).inputNode->response);
  }

  nodes[hiddenLayers + 1][i].response = nodes[hiddenLayers + 1][i].activationFunction(nodes[hiddenLayers + 1][i].inducedLocalField);
  outputVector[i] = nodes[hiddenLayers + 1][i].response;
}

//TRANSFORM OUTPUT RESULTS

for(int i = 0; i < outputSize; i++)
{
  ratioTranformation[i] = (maxResponse[i] - minResponse[i])/double(2);
}

for(int i = 0; i < outputSize; i++)
{
  if(outputVector[i] >= 1)
  {
```

```
          outputVector[i] = maxResponse[i] + ratioTranformation[i] * (outputVector[i] - double(1));
        }
        else if(outputVector[i] <= -1)
        {
          outputVector[i] = minResponse[i] + ratioTranformation[i] * (outputVector[i] - double(-1));
        }
        else
        {
          outputVector[i] = minResponse[i] + ratioTranformation[i] * fabs((outputVector[i] - double(-1)));
        }
      }

      delete [] meanVector;
      delete [] ratioTranformation;
      delete [] input;
    }
    catch(Exception &e)
    {
      ShowMessage(e.Message);
      ShowMessage("An unexpected error occured on Forecast function in TANeuralNetwork Unit.");
    }
}
//---------------------------------------------------------------------------
void __fastcall TANeuralNetwork::RandomizeTrainingData(double **trainingData, double **randomizedTrainingData,
                                       double **desiredResponse, double **randomizedDesiredResponse, int nOfExamples)
{
    try
    {
      int rVar;
      deque<int> trainingDataSequence;
      deque<int>::iterator tdsIterator;

      RandSeed = rand();

      for(int i = 0; i < nOfExamples; i++)
      {
        trainingDataSequence.push_back(i);
      }

      for(int i = 0; i < nOfExamples; i++)
      {
        rVar = rand() % (nOfExamples - i);
        tdsIterator = trainingDataSequence.begin();
        tdsIterator += rVar;

        for(int j = 0; j < inputSize; j++)
        {
          randomizedTrainingData[i][j] = trainingData[(*tdsIterator)][j];
        }
        for(int j = 0; j < outputSize; j++)
        {
          randomizedDesiredResponse[i][j] = desiredResponse[(*tdsIterator)][j];
        }

        trainingDataSequence.erase(tdsIterator);
      }
    }
    catch(Exception &e)
    {
      ShowMessage(e.Message);
      ShowMessage("An unexpected error occured on RandomizeTrainingData function in TANeuralNetwork Unit.");
    }
}
//---------------------------------------------------------------------------
bool MatrixTranspoze(long double **a, int aRows, int aCols,
                     long double **b, int bRows, int bCols)
{
```

```
  if((aRows != bCols) || (aCols != bRows)) return false;

  for(int i = 0; i < aRows; i++)
  {
    for(int j = 0; j < aCols; j++)
    {
      b[j][i] = a[i][j];
    }
  }

  return true;
}
//---------------------------------------------------------------------------
bool MatrixSubtract(long double **a, int aRows, int aCols,
                    long double **b, int bRows, int bCols,
                    long double **c, int cRows, int cCols)
{
  if((aRows != bRows) || (aCols != bCols) || (aRows != cRows) || (aCols != cCols)) return false;

  for(int i = 0; i < aRows; i++)
  {
    for(int j = 0; j < aCols; j++)
    {
      c[i][j] = a[i][j] - b[i][j];
    }
  }

  return true;
}
//---------------------------------------------------------------------------
bool MatrixMultiply(long double **a, int aRows, int aCols,
                    long double **b, int bRows, int bCols,
                    long double **c, int cRows, int cCols)
{
  if(aCols != bRows) return false;

  for(int i = 0; i < aRows; i++)
  {
    for(int j = 0; j < bCols; j++)
    {
      double tempResult = 0;
      for(int k = 0; k < aCols; k++)
      {
        tempResult += a[i][k]*b[k][j];
      }
      c[i][j] = tempResult;
    }
  }

  return true;
}
//---------------------------------------------------------------------------
double NDistribution(double X, double s, double m)
{
  try
  {
    double retValue, c;
    c = double(1)/(s*sqrt(double(2)*M_PI));
    retValue = c * exp(-pow((X-m), 2) / (double(2)*pow(s, 2)));
    return retValue;
  }
  catch(Exception &e)
  {
    ShowMessage(e.Message);
    ShowMessage("An unexpected error occured on NDistribution function in TANeuralNetwork Unit.");
  }
  return 0;
```

```
}
//---------------------------------------------------------------------------
double HyperbolicTangent(double U)
{
  try
  {
    double a, b, result;

    a = 1.7159;
    b = double(2)/double(3);
    result = a*tanh(b*U);
    return result;
  }
  catch(Exception &e)
  {
    ShowMessage(e.Message);
    ShowMessage("An unexpected error occured on HyperbolicTangent function in TANeuralNetwork Unit.");
  }
  return 0;
}
//---------------------------------------------------------------------------
double HyperbolicTangentDerivative(double Y)
{
  try
  {
    double a, b, result;

    a = 1.7159;
    b = double(2)/double(3);
    result = (b/a)*(a - Y)*(a + Y);
    return result;
  }
  catch(Exception &e)
  {
    ShowMessage(e.Message);
    ShowMessage("An unexpected error occured on HyperbolicTangentDerivative function in TANeuralNetwork Unit.");
  }
  return 0;
}
//---------------------------------------------------------------------------
```

# Αλγόριθμος Απλής Παλινδρόμησης

```
void __fastcall TDataModule1::SRFMClick(TObject *Sender)
{
  try
  {
    int counter, counter2, xRows, xCols, maxSoloAppearance, paramPosition, index;
    double **X, **Y, *xValues, *yValues;
    double meanX, meanY, sumXY, sumX2, a, b, impact;
    bool foundFutureEvent, finished, complexedParameters, allFutureParamsCalculated, forecastOnlyOne;
    AnsiString paramCode;
    map<AnsiString, int> parameters;
    map<AnsiString, TSimpleRegressionParamsCoefficients> parametersCoefficients;
    TExpForecast *myForm;
    TMenuItem *myMenuItem;

    forecastOnlyOne = false;
    myMenuItem = dynamic_cast<TMenuItem *>(Sender);
    if(myMenuItem)
    {
      if(myMenuItem->Name == "SRFM2") forecastOnlyOne = true;
    }

    if(MainForm->MDIChildCount > 0)
    {
      myForm = dynamic_cast<TExpForecast *>(MainForm->MDIChildren[0]);
    }
    else myForm = NULL;
    if(myForm)
    {

      counter = 0;
      counter2 = 0;
      foundFutureEvent = false;

      for(map<TDate, TJudgmentalEvent>::iterator i = myForm->judgmentalEventBase.judgmentalEvents.begin(); i != myForm-
>judgmentalEventBase.judgmentalEvents.end(); i++)
      {
        if(!(*i).second.futureEvent)
        {
          counter2++;
        }
        else foundFutureEvent = true;

        for(map<AnsiString, TFactor>::iterator j = (*i).second.factors.begin(); j != (*i).second.factors.end(); j++)
        {
          for(map<AnsiString, TParameter>::iterator k = (*j).second.parameters.begin(); k != (*j).second.parameters.end(); k++)
          {
            if(parameters.find(AnsiString((*j).second.name + "." + (*k).second.name)) == parameters.end())
            {
              parameters[AnsiString((*j).second.name + "." + (*k).second.name)] = counter;
              parametersCoefficients[AnsiString((*j).second.name + "." + (*k).second.name)].b0 = 0;
              parametersCoefficients[AnsiString((*j).second.name + "." + (*k).second.name)].b1 = 0;
              parametersCoefficients[AnsiString((*j).second.name + "." + (*k).second.name)].coefficientsCalculated = false;
              counter++;
            }
          }
        }
      }

      xCols = counter;
      xRows = counter2;

      if(xCols == 0)
      {
```

```
    Timer1->Enabled = false;
    MainForm->curAction = error;
    if(curLanguage == Greek)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Αδύνατη πρόβλεψη. Κανένα υποκειμενικό γεγονός δεν έχει εισαχθεί στη ΒΔ.";
    }
    else if(curLanguage == English)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Impossible forecast. None Judgmental event has been inserted into Database.";
    }
    MainForm->StatusBar1->Invalidate();
    Timer1->Enabled = true;

    return;
  }
  if(xRows == 0)
  {
    Timer1->Enabled = false;
    MainForm->curAction = error;
    if(curLanguage == Greek)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Αδύνατη πρόβλεψη. Υπάρχουν ελλειπή ιστορικά δεδομένα.";
    }
    else if(curLanguage == English)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Forecast impossible. Not enough history data.";
    }
    MainForm->StatusBar1->Invalidate();
    Timer1->Enabled = true;

    return;
  }

  if(!foundFutureEvent)
  {
    Timer1->Enabled = false;
    MainForm->curAction = error;
    if(curLanguage == Greek)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Αδύνατη πρόβλεψη. Δεν έχει δηλωθεί κανένα μελλοντικό υποκειμενικό γεγονός.";
    }
    else if(curLanguage == English)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Forecast impossible. No future event has been declared.";
    }
    MainForm->StatusBar1->Invalidate();
    Timer1->Enabled = true;

    return;
  }

  X = new double*[xRows];
  for(int i = 0; i < xRows; i++)
  {
    X[i] = new double[xCols];
  }

  Y = new double*[xRows];
  for(int i = 0; i < xRows; i++)
  {
    Y[i] = new double[1];
  }

  for(int i = 0; i < xRows; i++)
  {
    for(int j = 0; j < xCols; j++)
    {
```

```
            X[i][j] = 0;
          }
        }

        for(int i = 0; i < xRows; i++)
        {
          Y[i][0] = 0;
        }

        counter = 0;
        for(map<TDate, TJudgmentalEvent>::iterator i = myForm->judgmentalEventBase.judgmentalEvents.begin(); i != myForm-
>judgmentalEventBase.judgmentalEvents.end(); i++)
        {
          if(!(*i).second.futureEvent)
          {
            for(map<AnsiString, TFactor>::iterator j = (*i).second.factors.begin(); j != (*i).second.factors.end(); j++)
            {
              for(map<AnsiString, TParameter>::iterator k = (*j).second.parameters.begin(); k != (*j).second.parameters.end(); k++)
              {
                if(parameters.find(AnsiString((*j).second.name + "." + (*k).second.name)) != parameters.end())
                {
                  X[counter][parameters[AnsiString((*j).second.name + "." + (*k).second.name)]] = (*k).second.value;
                }
              }
            }
            Y[counter][0] = (*i).second.impact;
            counter++;
          }
        }

        finished = false;
        while(!finished)
        {
          //FIND PARAMETER THAT APPEARS ALONE MORE TIMES THAN ALL OTHERS IN JUDGMENTAL EVENT BASE

          maxSoloAppearance = 1;
          paramPosition = -1;

          for(int i = 0; i < xCols; i++)
          {
            counter = 0;
            for(int j = 0; j < xRows; j++)
            {
              complexedParameters = false;
              if(X[j][i] != 0)
              {
                for(int k = 0; k < xCols; k++)
                {
                  if((k != i) && (X[j][k] != 0))
                  {
                    complexedParameters = true;
                    break;
                  }
                }
                if(!complexedParameters)
                {
                  counter++;
                }
              }
            }

            if(counter > maxSoloAppearance)
            {
              maxSoloAppearance = counter;
              paramPosition = i;

              for(map<AnsiString, int>::iterator j = parameters.begin(); j != parameters.end(); j++)
```

```
      {
        if((*j).second == paramPosition) paramCode = (*j).first;
      }
    }
  }

  if(paramPosition != -1)
  {
    //CALCULATE SIMPLE REGRESSION PARAMETERS

    yValues = new double[maxSoloAppearance];
    xValues = new double[maxSoloAppearance];

    counter = 0;
    for(int i = 0; i < xRows; i++)
    {
      complexedParameters = false;
      if(X[i][paramPosition] != 0)
      {
        for(int j = 0; j < xCols; j++)
        {
          if((j != paramPosition) && (X[i][j] != 0))
          {
            complexedParameters = true;
            break;
          }
        }
        if(!complexedParameters)
        {
          xValues[counter] = X[i][paramPosition];
          yValues[counter] = Y[i][0];
          counter++;
        }
      }
    }

    meanX = 0;
    meanY = 0;
    sumXY = 0;
    sumX2 = 0;
    for(int i = 0; i < maxSoloAppearance; i++)
    {
      meanX += xValues[i];
      meanY += yValues[i];
      sumXY += xValues[i]*yValues[i];
      sumX2 += pow(xValues[i], double(2));
    }
    meanX = meanX / double(maxSoloAppearance);
    meanY = meanY / double(maxSoloAppearance);

    if((sumX2/double(maxSoloAppearance) - pow(meanX, double(2))) == 0)
    {
      Timer1->Enabled = false;
      MainForm->curAction = error;
      if(curLanguage == Greek)
      {
        MainForm->StatusBar1->Panels->Items[1]->Text = "Αδύνατη πρόβλεψη. Το σύστημα είναι μη επιλύσιμο.";
      }
      else if(curLanguage == English)
      {
        MainForm->StatusBar1->Panels->Items[1]->Text = "Impossible forecast. Cannot solve linear system.";
      }
      MainForm->StatusBar1->Invalidate();
      Timer1->Enabled = true;

      delete [] xValues;
      delete [] yValues;
```

```
            for(int i = 0; i < xRows; i++)
            {
              delete [] X[i];
            }
            delete [] X;

            for(int i = 0; i < xRows; i++)
            {
              delete [] Y[i];
            }
            delete [] Y;

            return;
          }
          else
          {
            b = (sumXY/double(maxSoloAppearance) - meanX*meanY)/(sumX2/double(maxSoloAppearance) - pow(meanX, double(2)));
            a = meanY - b*meanX;
          }

          //STORE SIMPLE REGRESSION PARAMETERS

          if(parametersCoefficients.find(paramCode) != parametersCoefficients.end())
          {
            parametersCoefficients[paramCode].b0 = a;
            parametersCoefficients[paramCode].b1 = b;
            parametersCoefficients[paramCode].coefficientsCalculated = true;
          }
          else
          {
            if(curLanguage == Greek)
            {
              ShowMessage("Κάποιο λάθος παρουσιάστηκε κατά τη διαδικασία υπολογισμού των παραμέτρων της απλής παλινδρόμησης.");
            }
            else if(curLanguage == English)
            {
              ShowMessage("An unexpected error occured during simple regression constants calculation.");
            }
            throw(-1);
          }

          for(int i = 0; i < xRows; i++)
          {
            if(X[i][paramPosition] != 0)
            {
              impact = a + b * X[i][paramPosition];
              X[i][paramPosition] = 0;
              Y[i][0] -= impact;
            }
          }

          delete [] xValues;
          delete [] yValues;
        }
        else finished = true;
      }

      allFutureParamsCalculated = true;
      for(map<TDate, TJudgmentalEvent>::iterator i = myForm->judgmentalEventBase.judgmentalEvents.begin(); i != myForm-
>judgmentalEventBase.judgmentalEvents.end(); i++)
      {
        if((*i).second.futureEvent)
        {
          for(map<AnsiString, TFactor>::iterator j = (*i).second.factors.begin(); j != (*i).second.factors.end(); j++)
          {
            for(map<AnsiString, TParameter>::iterator k = (*j).second.parameters.begin(); k != (*j).second.parameters.end(); k++)
```

```
            {
              if(parametersCoefficients.find(AnsiString((*j).second.name + "." + (*k).second.name)) != parametersCoefficients.end())
              {
                if(!parametersCoefficients[AnsiString((*j).second.name + "." + (*k).second.name)].coefficientsCalculated)
                {
                  allFutureParamsCalculated = false;
                }
              }
            }
          }
        }
      }

      if(allFutureParamsCalculated)
      {
        for(map<TDate, TJudgmentalEvent>::iterator i = myForm->judgmentalEventBase.judgmentalEvents.begin(); i != myForm-
>judgmentalEventBase.judgmentalEvents.end(); i++)
        {
          if(((!forecastOnlyOne) && ((*i).second.futureEvent)) || (forecastOnlyOne && (i == myForm->selectedEvent) &&
((*i).second.futureEvent)))
          {
            impact = 0;

            for(map<AnsiString, TFactor>::iterator j = (*i).second.factors.begin(); j != (*i).second.factors.end(); j++)
            {
              for(map<AnsiString, TParameter>::iterator k = (*j).second.parameters.begin(); k != (*j).second.parameters.end(); k++)
              {
                if(parametersCoefficients.find(AnsiString((*j).second.name + "." + (*k).second.name)) != parametersCoefficients.end())
                {
                  paramCode = AnsiString((*j).second.name + "." + (*k).second.name);
                  impact += parametersCoefficients[paramCode].b0 + parametersCoefficients[paramCode].b1 * (*k).second.value;
                }
              }
            }

            (*i).second.impact = impact;
            index = myForm->Series2->XValues->Locate((*i).second.eventDate);
            if(index != -1)
            {
              myForm->Series2->YValues->Value[index] = impact;
            }
            else
            {
              index = myForm->Series3->XValues->Locate((*i).second.eventDate);
              if(index != -1)
              {
                if(myForm->Series3->ValueColor[index] == clAqua)
                {
                  myForm->Series2->AddXY((*i).second.eventDate, impact, (*i).second.eventDate.DateString(), clAqua);
                  if(int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count())) > 100)
                  {
                    myForm->Series2->BarWidthPercent = 100;
                  }
                  else
                  {
                    myForm->Series2->BarWidthPercent = int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count()));
                  }
                }
                else
                {
                  myForm->Series2->AddXY((*i).second.eventDate, impact, (*i).second.eventDate.DateString(), clYellow);
                  if(int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count())) > 100)
                  {
                    myForm->Series2->BarWidthPercent = 100;
                  }
                  else
                  {
```

```
                myForm->Series2->BarWidthPercent = int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count()));
              }
            }
          }
          else
          {
            if((*i).second.eventDate.DateString() == myForm->DateBox->Text)
            {
              myForm->Series2->AddXY((*i).second.eventDate, impact, (*i).second.eventDate.DateString(), clAqua);
              if(int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count())) > 100)
              {
                myForm->Series2->BarWidthPercent = 100;
              }
              else
              {
                myForm->Series2->BarWidthPercent = int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count()));
              }
            }
            else
            {
              myForm->Series2->AddXY((*i).second.eventDate, impact, (*i).second.eventDate.DateString(), clYellow);
              if(int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count())) > 100)
              {
                myForm->Series2->BarWidthPercent = 100;
              }
              else
              {
                myForm->Series2->BarWidthPercent = int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count()));
              }
            }
          }
        }
        myForm->Series2->Repaint();
      }
    }
  }
  else
  {
    for(int i = 0; i < xRows; i++)
    {
      delete [] X[i];
    }
    delete [] X;

    for(int i = 0; i < xRows; i++)
    {
      delete [] Y[i];
    }
    delete [] Y;

    Timer1->Enabled = false;
    MainForm->curAction = error;
    if(curLanguage == Greek)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Αδύνατη πρόβλεψη. Το σύστημα είμαι μη επιλύσιμο.";
    }
    else if(curLanguage == English)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Forecast impossible. Can't find Regression constants.";
    }
    MainForm->StatusBar1->Invalidate();
    Timer1->Enabled = true;

    return;
  }

  for(int i = 0; i < xRows; i++)
```

```
        {
          delete [] X[i];
        }
        delete [] X;

        for(int i = 0; i < xRows; i++)
        {
          delete [] Y[i];
        }
        delete [] Y;

        Timer1->Enabled = false;
        MainForm->curAction = forecastAction;
        if(curLanguage == Greek)
        {
          MainForm->StatusBar1->Panels->Items[1]->Text = "Η μέθοδος της Απλής Παλινδρόμησης εφαρμόστηκε με επιτυχία.";
        }
        else if(curLanguage == English)
        {
          MainForm->StatusBar1->Panels->Items[1]->Text = "Simple regression method has been applied successfully.";
        }
        MainForm->StatusBar1->Invalidate();
        Timer1->Enabled = true;
      }
      else
      {
        Timer1->Enabled = false;
        MainForm->curAction = error;
        if(curLanguage == Greek)
        {
          MainForm->StatusBar1->Panels->Items[1]->Text = "Επιλέξτε πρώτα τη χρονοσειρά πάνω στην οποία θα εφαρμοστεί η συγκεκριμένη
μέθοδος.";
        }
        else if(curLanguage == English)
        {
          MainForm->StatusBar1->Panels->Items[1]->Text = "Please select a timeseries chart first.";
        }
        MainForm->StatusBar1->Invalidate();
        Timer1->Enabled = true;
      }
    }
    catch(Exception &e)
    {
      ShowMessage(e.Message);
      ShowMessage("An unexpected error occured in SRFMClick event in DataModule unit.");
    }
}
//---------------------------------------------------------------------------
```

# Αλγόριθμος Πολλαπλής Παλινδρόμησης

```
void __fastcall TDataModule1::MRFMClick(TObject *Sender)
{
  try
  {
    TExpForecast *myForm;

    if(MainForm->MDIChildCount > 0)
    {
      myForm = dynamic_cast<TExpForecast *>(MainForm->MDIChildren[0]);
    }
    else myForm = NULL;
    if(myForm)
    {
      double **X, **Y, **b, **X2, **resA, **resB, **forX, **impact;
      int xRows, xCols, counter, counter2, index;
      bool foundFutureEvent, forecastOnlyOne;
      map<AnsiString, int> parameters;
      TMenuItem *myMenuItem;

      forecastOnlyOne = false;
      myMenuItem = dynamic_cast<TMenuItem *>(Sender);
      if(myMenuItem)
      {
        if(myMenuItem->Name == "MRFM2") forecastOnlyOne = true;
      }

      counter = 1;
      counter2 = 0;
      foundFutureEvent = false;

      for(map<TDate, TJudgmentalEvent>::iterator i = myForm->judgmentalEventBase.judgmentalEvents.begin(); i != myForm-
>judgmentalEventBase.judgmentalEvents.end(); i++)
      {
        if(!(*i).second.futureEvent)
        {
          counter2++;
        }
        else foundFutureEvent = true;

        for(map<AnsiString, TFactor>::iterator j = (*i).second.factors.begin(); j != (*i).second.factors.end(); j++)
        {
          for(map<AnsiString, TParameter>::iterator k = (*j).second.parameters.begin(); k != (*j).second.parameters.end(); k++)
          {
            if(parameters.find(AnsiString((*j).second.name + "." + (*k).second.name)) == parameters.end())
            {
              parameters[AnsiString((*j).second.name + "." + (*k).second.name)] = counter;
              counter++;
            }
          }
        }
      }

      xCols = counter;
      xRows = counter2;

      if(xCols == 1)
      {
        Timer1->Enabled = false;
        MainForm->curAction = error;
        if(curLanguage == Greek)
        {
          MainForm->StatusBar1->Panels->Items[1]->Text = "Αδύνατη πρόβλεψη. Κανένα υποκειμενικό γεγονός δεν έχει εισαχθεί στη ΒΔ.";
        }
```

```
    else if(curLanguage == English)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Impossible forecast. None Judgmental event has been inserted into Database.";
    }
    MainForm->StatusBar1->Invalidate();
    Timer1->Enabled = true;

    return;
  }
  if(xRows == 0)
  {
    Timer1->Enabled = false;
    MainForm->curAction = error;
    if(curLanguage == Greek)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Αδύνατη πρόβλεψη. Υπάρχουν ελλειπή ιστορικά δεδομένα.";
    }
    else if(curLanguage == English)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Forecast impossible. Not enough history data.";
    }
    MainForm->StatusBar1->Invalidate();
    Timer1->Enabled = true;

    return;
  }

  if(!foundFutureEvent)
  {
    Timer1->Enabled = false;
    MainForm->curAction = error;
    if(curLanguage == Greek)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Αδύνατη πρόβλεψη. Δεν έχει δηλωθεί κανένα μελλοντικό υποκειμενικό γεγονός.";
    }
    else if(curLanguage == English)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Forecast impossible. No future event has been declared.";
    }
    MainForm->StatusBar1->Invalidate();
    Timer1->Enabled = true;

    return;
  }

  X = new double*[xRows];
  for(int i = 0; i < xRows; i++)
  {
    X[i] = new double[xCols];
  }

  Y = new double*[xRows];
  for(int i = 0; i < xRows; i++)
  {
    Y[i] = new double[1];
  }

  b = new double*[xCols];
  for(int i = 0; i < xCols; i++)
  {
    b[i] = new double[1];
  }

  X2 = new double*[xCols];
  for(int i = 0; i < xCols; i++)
  {
    X2[i] = new double[xRows];
```

```
    }

    resA = new double*[xCols];
    for(int i = 0; i < xCols; i++)
    {
      resA[i] = new double[xCols];
    }

    resB = new double*[xCols];
    for(int i = 0; i < xCols; i++)
    {
      resB[i] = new double[xRows];
    }

    forX = new double*[1];
    for(int i = 0; i < 1; i++)
    {
      forX[i] = new double[xCols];
    }

    impact = new double*[1];
    impact[0] = new double[1];

    for(int i = 0; i < xRows; i++)
    {
      for(int j = 0; j < xCols; j++)
      {
        if(j != 0) X[i][j] = 0;
        else X[i][j] = 1;
      }
    }

    for(int i = 0; i < xRows; i++)
    {
      Y[i][0] = 0;
    }

    for(int i = 0; i < xCols; i++)
    {
      b[i][0] = 0;
    }

    forX[0][0] = 1;
    for(int i = 1; i < xCols; i++)
    {
      forX[0][i] = 0;
    }

    counter = 0;
    for(map<TDate, TJudgmentalEvent>::iterator i = myForm->judgmentalEventBase.judgmentalEvents.begin(); i != myForm-
>judgmentalEventBase.judgmentalEvents.end(); i++)
    {
      if(!(*i).second.futureEvent)
      {
        for(map<AnsiString, TFactor>::iterator j = (*i).second.factors.begin(); j != (*i).second.factors.end(); j++)
        {
          for(map<AnsiString, TParameter>::iterator k = (*j).second.parameters.begin(); k != (*j).second.parameters.end(); k++)
          {
            if(parameters.find(AnsiString((*j).second.name + "." + (*k).second.name)) != parameters.end())
            {
              X[counter][parameters[AnsiString((*j).second.name + "." + (*k).second.name)]] = (*k).second.value;
            }
          }
        }
        Y[counter][0] = (*i).second.impact;
        counter++;
      }
```

```
}

MatrixTranspoze(X, xRows, xCols, X2, xCols, xRows);
MatrixMultiply(X2, xCols, xRows, X, xRows, xCols, resA, xCols, xCols);
if(!Invert(xCols, resA))
{
  for(int i = 0; i < xRows; i++)
  {
    delete [] X[i];
  }
  delete [] X;

  for(int i = 0; i < xRows; i++)
  {
    delete [] Y[i];
  }
  delete [] Y;

  for(int i = 0; i < xCols; i++)
  {
    delete [] b[i];
  }
  delete [] b;

  for(int i = 0; i < xCols; i++)
  {
    delete [] X2[i];
  }
  delete [] X2;

  for(int i = 0; i < xCols; i++)
  {
    delete [] resA[i];
  }
  delete [] resA;

  for(int i = 0; i < xCols; i++)
  {
    delete [] resB[i];
  }
  delete [] resB;

  delete [] forX[0];
  delete [] forX;

  delete [] impact[0];
  delete [] impact;

  Timer1->Enabled = false;
  MainForm->curAction = error;
  if(curLanguage == Greek)
  {
    MainForm->StatusBar1->Panels->Items[1]->Text = "Αδύνατη πρόβλεψη. Το σύστημα που προκύπτει είναι μη επιλύσιμο.";
  }
  else if(curLanguage == English)
  {
    MainForm->StatusBar1->Panels->Items[1]->Text = "Forecast impossible. Can't find Multiple Regression constants.";
  }
  MainForm->StatusBar1->Invalidate();
  Timer1->Enabled = true;

  return;
}
MatrixMultiply(resA, xCols, xCols, X2, xCols, xRows, resB, xCols, xRows);
MatrixMultiply(resB, xCols, xRows, Y, xRows, 1, b, xCols, 1);
```

```
      for(map<TDate, TJudgmentalEvent>::iterator i = myForm->judgmentalEventBase.judgmentalEvents.begin(); i != myForm-
>judgmentalEventBase.judgmentalEvents.end(); i++)
      {
        if(((!forecastOnlyOne) && ((*i).second.futureEvent)) || (forecastOnlyOne && (i == myForm->selectedEvent) && ((*i).second.futureEvent)))
        {
          forX[0][0] = 1;
          for(int j = 1; j < xCols; j++)
          {
            forX[0][j] = 0;
          }

          for(map<AnsiString, TFactor>::iterator j = (*i).second.factors.begin(); j != (*i).second.factors.end(); j++)
          {
            for(map<AnsiString, TParameter>::iterator k = (*j).second.parameters.begin(); k != (*j).second.parameters.end(); k++)
            {
              if(parameters.find(AnsiString((*j).second.name + "." + (*k).second.name)) != parameters.end())
              {
                forX[0][parameters[AnsiString((*j).second.name + "." + (*k).second.name)]] = (*k).second.value;
              }
            }
          }

          MatrixMultiply(forX, 1, xCols, b, xCols, 1, impact, 1, 1);
          (*i).second.impact = impact[0][0];
          index = myForm->Series2->XValues->Locate((*i).second.eventDate);
          if(index != -1)
          {
            myForm->Series2->YValues->Value[index] = impact[0][0];
          }
          else
          {
            index = myForm->Series3->XValues->Locate((*i).second.eventDate);
            if(index != -1)
            {
              if(myForm->Series3->ValueColor[index] == clAqua)
              {
                myForm->Series2->AddXY((*i).second.eventDate, impact[0][0], (*i).second.eventDate.DateString(), clAqua);
                if(int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count())) > 100)
                {
                  myForm->Series2->BarWidthPercent = 100;
                }
                else
                {
                  myForm->Series2->BarWidthPercent = int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count()));
                }
              }
              else
              {
                myForm->Series2->AddXY((*i).second.eventDate, impact[0][0], (*i).second.eventDate.DateString(), clYellow);
                if(int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count())) > 100)
                {
                  myForm->Series2->BarWidthPercent = 100;
                }
                else
                {
                  myForm->Series2->BarWidthPercent = int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count()));
                }
              }
            }
            else
            {
              if((*i).second.eventDate.DateString() == myForm->DateBox->Text)
              {
                myForm->Series2->AddXY((*i).second.eventDate, impact[0][0], (*i).second.eventDate.DateString(), clAqua);
                if(int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count())) > 100)
                {
                  myForm->Series2->BarWidthPercent = 100;
```

```
            }
            else
            {
              myForm->Series2->BarWidthPercent = int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count()));
            }
          }
          else
          {
            myForm->Series2->AddXY((*i).second.eventDate, impact[0][0], (*i).second.eventDate.DateString(), clYellow);
            if(int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count())) > 100)
            {
              myForm->Series2->BarWidthPercent = 100;
            }
            else
            {
              myForm->Series2->BarWidthPercent = int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count()));
            }
          }
        }
      }
      myForm->Series2->Repaint();
    }
  }

  for(int i = 0; i < xRows; i++)
  {
    delete [] X[i];
  }
  delete [] X;

  for(int i = 0; i < xRows; i++)
  {
    delete [] Y[i];
  }
  delete [] Y;

  for(int i = 0; i < xCols; i++)
  {
    delete [] b[i];
  }
  delete [] b;

  for(int i = 0; i < xCols; i++)
  {
    delete [] X2[i];
  }
  delete [] X2;

  for(int i = 0; i < xCols; i++)
  {
    delete [] resA[i];
  }
  delete [] resA;

  for(int i = 0; i < xCols; i++)
  {
    delete [] resB[i];
  }
  delete [] resB;

  delete [] forX[0];
  delete [] forX;

  delete [] impact[0];
  delete [] impact;

  Timer1->Enabled = false;
```

```
        MainForm->curAction = forecastAction;
        if(curLanguage == Greek)
        {
          MainForm->StatusBar1->Panels->Items[1]->Text = "Η μέθοδος της Πολλαπλής Παλινδρόμησης εφαρμόστηκε με επιτυχία.";
        }
        else if(curLanguage == English)
        {
          MainForm->StatusBar1->Panels->Items[1]->Text = "Multiple Regression method has been applied successfully.";
        }
        MainForm->StatusBar1->Invalidate();
        Timer1->Enabled = true;
      }
      else
      {
        Timer1->Enabled = false;
        MainForm->curAction = error;
        if(curLanguage == Greek)
        {
          MainForm->StatusBar1->Panels->Items[1]->Text = "Επιλέξτε πρώτα τη χρονοσειρά πάνω στην οποία θα εφαρμοστεί η συγκεκριμένη
μέθοδος.";
        }
        else if(curLanguage == English)
        {
          MainForm->StatusBar1->Panels->Items[1]->Text = "Please select a timeseries chart first.";
        }
        MainForm->StatusBar1->Invalidate();
        Timer1->Enabled = true;
      }
    }
    catch(Exception &e)
    {
      ShowMessage(e.Message);
      ShowMessage("An unexpected error occured in MRFMClick event in DataModule unit.");
    }
}
//---------------------------------------------------------------------------
bool __fastcall TDataModule1::MatrixTranspoze(double **a, int aRows, int aCols,
                                              double **b, int bRows, int bCols)
{
  if((aRows != bCols) || (aCols != bRows)) return false;

  for(int i = 0; i < aRows; i++)
  {
    for(int j = 0; j < aCols; j++)
    {
      b[j][i] = a[i][j];
    }
  }

  return true;
}
//---------------------------------------------------------------------------
bool __fastcall TDataModule1::MatrixAdd(double **a, int aRows, int aCols,
                                        double **b, int bRows, int bCols,
                                        double **c, int cRows, int cCols)
{
  if((aRows != bRows) || (aCols != bCols) || (aRows != cRows) || (aCols != cCols)) return false;

  for(int i = 0; i < aRows; i++)
  {
    for(int j = 0; j < aCols; j++)
    {
      c[i][j] = a[i][j] + b[i][j];
    }
  }

  return true;
```

```
}
//---------------------------------------------------------------------------
bool __fastcall TDataModule1::MatrixMultiply(double **a, int aRows, int aCols,
                                             double **b, int bRows, int bCols,
                                             double **c, int cRows, int cCols)
{
  if(aCols != bRows) return false;

  for(int i = 0; i < aRows; i++)
  {
    for(int j = 0; j < bCols; j++)
    {
      double tempResult = 0;
      for(int k = 0; k < aCols; k++)
      {
        tempResult += a[i][k]*b[k][j];
      }
      c[i][j] = tempResult;
    }
  }

  return true;
}
//---------------------------------------------------------------------------
bool TDataModule1::Invert(int n, double **a)
{
  int i, j, k, *p;
  double h, q, s, sup, pivot;

  p = new int[n];

  for (k = 0; k < n; k++)
  {
    sup = 0.0;
            p[k] = 0;
            for (i = k; i < n; i++)
            {
      s = 0.0;
      for (j = k; j < n; j++)
      {
        s += fabs(a[i][j]);
      }
      if (s == 0.0)
      {
        delete [] p;
        return false;
      }
      q = fabs(a[i][k]) / s;
      if (sup < q)
      {
        sup = q;
        p[k] = i;
      }
            }
            if (sup == 0.0)
    {
      delete [] p;
      return false;
    }
            if (p[k] != k)
    {
              for (j = 0; j < n; j++)
              {
                      h = a[k][j];
                      a[k][j] = a[p[k]][j];
                      a[p[k]][j] = h;
              }
```

```
    }
                pivot = a[k][k];
    for (j = 0; j < n; j++)
    {
                  if (j != k)
      {
                        a[k][j] = -a[k][j] / pivot;
                        for (i = 0; i < n; i++)
        {
          if (i != k)
          {
                                  a[i][j] += a[i][k] * a[k][j];
          }
        }
                }
    }
              for (i = 0; i < n; i++)
    {
      a[i][k] = a[i][k] / pivot;
    }
            a[k][k] = 1.0 / pivot;
  }
  for (k = n - 1; k >= 0; k--)
  {
              if (p[k] != k)
    {
              for (i = 0; i < n; i++)
              {
                        h = a[i][k];
                        a[i][k] = a[i][p[k]];
                        a[i][p[k]] = h;
              }
    }
  }

  delete [] p;

  return true;
}
//---------------------------------------------------------------------------
```

## Μέθοδος των Γεπόνων

```
void __fastcall TNBForm::BitBtn1Click(TObject *Sender)
{
  try
  {
    int counter, counter2, xRows, xCols, index, neighborsNumber;
    double **X, **Y, impact, tempDouble;
    bool foundFutureEvent;
    TRankStruct *rankings;
    map<AnsiString, int> parameters;
    TExpForecast *myForm;

    myForm = dynamic_cast<TExpForecast *>(this->Owner);

    if(myForm)
    {
      counter = 0;
      counter2 = 0;
      foundFutureEvent = false;

      for(map<TDate, TJudgmentalEvent>::iterator i = myForm->judgmentalEventBase.judgmentalEvents.begin(); i != myForm-
>judgmentalEventBase.judgmentalEvents.end(); i++)
      {
        if(!(*i).second.futureEvent)
        {
          counter2++;
        }
        else foundFutureEvent = true;

        for(map<AnsiString, TFactor>::iterator j = (*i).second.factors.begin(); j != (*i).second.factors.end(); j++)
        {
          for(map<AnsiString, TParameter>::iterator k = (*j).second.parameters.begin(); k != (*j).second.parameters.end(); k++)
          {
            if(parameters.find(AnsiString((*j).second.name + "." + (*k).second.name)) == parameters.end())
            {
              parameters[AnsiString((*j).second.name + "." + (*k).second.name)] = counter;
              counter++;
            }
          }
        }
      }

      xCols = counter;
      xRows = counter2;

      if(xCols == 0)
      {
        DataModule1->Timer1->Enabled = false;
        MainForm->curAction = error;
        if(DataModule1->curLanguage == Greek)
        {
          MainForm->StatusBar1->Panels->Items[1]->Text = "Αδύνατη πρόβλεψη. Κανένα υποκειμενικό γεγονός δεν έχει εισαχθεί στη ΒΔ.";
        }
        else if(DataModule1->curLanguage == English)
        {
          MainForm->StatusBar1->Panels->Items[1]->Text = "Impossible forecast. None Judgmental event has been inserted into Database.";
        }
        MainForm->StatusBar1->Invalidate();
        DataModule1->Timer1->Enabled = true;

        ModalResult = mrNone;
        return;
```

```
  }
  if(xRows == 0)
  {
    DataModule1->Timer1->Enabled = false;
    MainForm->curAction = error;
    if(DataModule1->curLanguage == Greek)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Αδύνατη πρόβλεψη. Υπάρχουν ελλειπή ιστορικά δεδομένα.";
    }
    else if(DataModule1->curLanguage == English)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Forecast impossible. Not enough history data.";
    }
    MainForm->StatusBar1->Invalidate();
    DataModule1->Timer1->Enabled = true;

    ModalResult = mrNone;
    return;
  }

  if(!foundFutureEvent)
  {
    DataModule1->Timer1->Enabled = false;
    MainForm->curAction = error;
    if(DataModule1->curLanguage == Greek)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Αδύνατη πρόβλεψη. Δεν έχει δηλωθεί κανένα μελλοντικό υποκειμενικό γεγονός.";
    }
    else if(DataModule1->curLanguage == English)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Forecast impossible. No future event has been declared.";
    }
    MainForm->StatusBar1->Invalidate();
    DataModule1->Timer1->Enabled = true;

    ModalResult = mrNone;
    return;
  }

  if((ComboBox1->ItemIndex == 2) && (Edit1->Text.Trim().IsEmpty()))
  {
    DataModule1->Timer1->Enabled = false;
    MainForm->curAction = error;
    if(DataModule1->curLanguage == Greek)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Η επιλεγμένη μέθοδος απαιτεί να συμπληρώσετε τον αριθμό των γειτόνων.";
    }
    else if(DataModule1->curLanguage == English)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Selected method requires neighbors number to be defined.";
    }
    MainForm->StatusBar1->Invalidate();
    DataModule1->Timer1->Enabled = true;

    ModalResult = mrNone;
    return;
  }

  if(ComboBox1->ItemIndex == 2)
  {
    try
    {
      neighborsNumber = StrToInt(Edit1->Text);
    }
    catch(EConvertError &e)
    {
      DataModule1->Timer1->Enabled = false;
```

```
      MainForm->curAction = error;
      if(DataModule1->curLanguage == Greek)
      {
        MainForm->StatusBar1->Panels->Items[1]->Text = "Παρακαλώ συμπληρώστε έναν αριθμό σε σωστή μορφή.";
      }
      else if(DataModule1->curLanguage == English)
      {
        MainForm->StatusBar1->Panels->Items[1]->Text = "Please fill in a valid number of neighbors.";
      }
      MainForm->StatusBar1->Invalidate();
      DataModule1->Timer1->Enabled = true;

      ModalResult = mrNone;
      return;
    }
  }

  if((ComboBox1->ItemIndex == 1) && (xRows < 3))
  {
    DataModule1->Timer1->Enabled = false;
    MainForm->curAction = error;
    if(DataModule1->curLanguage == Greek)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Δεν υπάρχουν αρκετά ιστορικά δεδομένα για την εφαρμογή της επιλεγμένης μεθόδου.";
    }
    else if(DataModule1->curLanguage == English)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "There are not enough judgmental events for the selected method to be applied.";
    }
    MainForm->StatusBar1->Invalidate();
    DataModule1->Timer1->Enabled = true;

    ModalResult = mrNone;
    return;
  }
  else if((ComboBox1->ItemIndex == 2) && (xRows < neighborsNumber))
  {
    DataModule1->Timer1->Enabled = false;
    MainForm->curAction = error;
    if(DataModule1->curLanguage == Greek)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Δεν υπάρχουν αρκετά ιστορικά δεδομένα για την εφαρμογή της επιλεγμένης μεθόδου.";
    }
    else if(DataModule1->curLanguage == English)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "There are not enough judgmental events for the selected method to be applied.";
    }
    MainForm->StatusBar1->Invalidate();
    DataModule1->Timer1->Enabled = true;

    ModalResult = mrNone;
    return;
  }

  X = new double*[xRows];
  for(int i = 0; i < xRows; i++)
  {
    X[i] = new double[xCols];
  }

  Y = new double*[xRows];
  for(int i = 0; i < xRows; i++)
  {
    Y[i] = new double[1];
  }

  rankings = new TRankStruct[xRows];
```

```
      for(int i = 0; i < xRows; i++)
      {
        for(int j = 0; j < xCols; j++)
        {
          X[i][j] = 0;
        }
      }

      for(int i = 0; i < xRows; i++)
      {
        Y[i][0] = 0;
      }

      counter = 0;
      for(map<TDate, TJudgmentalEvent>::iterator i = myForm->judgmentalEventBase.judgmentalEvents.begin(); i != myForm-
>judgmentalEventBase.judgmentalEvents.end(); i++)
      {
        if(!(*i).second.futureEvent)
        {
          for(map<AnsiString, TFactor>::iterator j = (*i).second.factors.begin(); j != (*i).second.factors.end(); j++)
          {
            for(map<AnsiString, TParameter>::iterator k = (*j).second.parameters.begin(); k != (*j).second.parameters.end(); k++)
            {
              if(parameters.find(AnsiString((*j).second.name + "." + (*k).second.name)) != parameters.end())
              {
                X[counter][parameters[AnsiString((*j).second.name + "." + (*k).second.name)]] = (*k).second.value;
              }
            }
          }
          Y[counter][0] = (*i).second.impact;
          counter++;
        }
      }

      for(map<TDate, TJudgmentalEvent>::iterator i = myForm->judgmentalEventBase.judgmentalEvents.begin(); i != myForm-
>judgmentalEventBase.judgmentalEvents.end(); i++)
      {
        if(((!forecastOnlyOne) && ((*i).second.futureEvent)) || (forecastOnlyOne && (i == myForm->selectedEvent) && ((*i).second.futureEvent)))
        {
          impact = 0;

          for(int j = 0; j < xRows; j++)
          {
            rankings[j].penalty = 0;
            rankings[j].eventPointer = j;

            for(map<AnsiString, TFactor>::iterator k = (*i).second.factors.begin(); k != (*i).second.factors.end(); k++)
            {
              for(map<AnsiString, TParameter>::iterator l = (*k).second.parameters.begin(); l != (*k).second.parameters.end(); l++)
              {
                if(parameters.find(AnsiString((*k).second.name + "." + (*l).second.name)) != parameters.end())
                {
                  tempDouble = X[j][parameters[AnsiString((*k).second.name + "." + (*l).second.name)]];

                  if((tempDouble != 0) && ((*l).second.value == 0))
                  {
                    rankings[j].penalty += double(1);
                  }
                  else if((tempDouble == 0) && ((*l).second.value != 0))
                  {
                    rankings[j].penalty += double(1);
                  }
                  else if((tempDouble != 0) && ((*l).second.value != 0))
                  {
                    rankings[j].penalty += fabs(tempDouble - (*l).second.value)/fabs(tempDouble);
                  }
```

```
        }
      }
    }
  }

  BubbleSort(rankings, xRows);

  if(ComboBox1->ItemIndex == 0)
  {
    impact = Y[rankings[0].eventPointer][0];
  }
  else if(ComboBox1->ItemIndex == 1)
  {
    impact = 0.5*Y[rankings[0].eventPointer][0] + 0.25*Y[rankings[1].eventPointer][0] + 0.25*Y[rankings[2].eventPointer][0];
  }
  else if(ComboBox1->ItemIndex == 2)
  {
    impact = 0;
    for(int j = 0; j < neighborsNumber; j++)
    {
      impact += (double(1)/double(neighborsNumber))*Y[rankings[j].eventPointer][0];
    }
  }

  (*i).second.impact = impact;
  index = myForm->Series2->XValues->Locate((*i).second.eventDate);
  if(index != -1)
  {
    myForm->Series2->YValues->Value[index] = impact;
  }
  else
  {
    index = myForm->Series3->XValues->Locate((*i).second.eventDate);
    if(index != -1)
    {
      if(myForm->Series3->ValueColor[index] == clAqua)
      {
        myForm->Series2->AddXY((*i).second.eventDate, impact, (*i).second.eventDate.DateString(), clAqua);
        if(int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count())) > 100)
        {
          myForm->Series2->BarWidthPercent = 100;
        }
        else
        {
          myForm->Series2->BarWidthPercent = int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count()));
        }
      }
      else
      {
        myForm->Series2->AddXY((*i).second.eventDate, impact, (*i).second.eventDate.DateString(), clYellow);
        if(int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count())) > 100)
        {
          myForm->Series2->BarWidthPercent = 100;
        }
        else
        {
          myForm->Series2->BarWidthPercent = int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count()));
        }
      }
    }
    else
    {
      if((*i).second.eventDate.DateString() == myForm->DateBox->Text)
      {
        myForm->Series2->AddXY((*i).second.eventDate, impact, (*i).second.eventDate.DateString(), clAqua);
        if(int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count())) > 100)
        {
```

```
                myForm->Series2->BarWidthPercent = 100;
              }
              else
              {
                myForm->Series2->BarWidthPercent = int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count()));
              }
            }
            else
            {
              myForm->Series2->AddXY((*i).second.eventDate, impact, (*i).second.eventDate.DateString(), clYellow);
              if(int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count())) > 100)
              {
                myForm->Series2->BarWidthPercent = 100;
              }
              else
              {
                myForm->Series2->BarWidthPercent = int(double(10) + double(9)/double(5)*double(myForm->Series2->XValues->Count()));
              }
            }
          }
        }

        myForm->Series2->Repaint();
      }
    }

    for(int i = 0; i < xRows; i++)
    {
      delete [] X[i];
    }
    delete [] X;

    for(int i = 0; i < xRows; i++)
    {
      delete [] Y[i];
    }
    delete [] Y;

    delete [] rankings;

    DataModule1->Timer1->Enabled = false;
    MainForm->curAction = forecastAction;
    if(DataModule1->curLanguage == Greek)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "Η μέθοδος των Γειτόνων εφαρμόστηκε με επιτυχία.";
    }
    else if(DataModule1->curLanguage == English)
    {
      MainForm->StatusBar1->Panels->Items[1]->Text = "The neighbors method has been applied successfully.";
    }
    MainForm->StatusBar1->Invalidate();
    DataModule1->Timer1->Enabled = true;
  }
 }
 catch(Exception &e)
 {
  ShowMessage(e.Message);
  ShowMessage("An unexpected error occured in BitBtn1Click event in TNBForm unit.");
 }
}
//---------------------------------------------------------------------------
void __fastcall TNBForm::Swap(TRankStruct &a, TRankStruct &b)
{
 TRankStruct temp = a;
 a = b;
 b = temp;
}
```

```
//---------------------------------------------------------------------------
void __fastcall TNBForm::BubbleSort(TRankStruct *rankings, int xRows)
{
  for(int i = xRows - 1; i > 0; i--)
  {
    for(int j = 0; j < i; j++)
    {
      if(rankings[j].penalty > rankings[j+1].penalty) swap(rankings[j], rankings[j+1]);
    }
  }
}
//---------------------------------------------------------------------------
```

## Χρήσιμες Δηλώσεις

```
class TParameter
{
public:
  TParameter(){ value = 0; valueName = "";};
  ~TParameter(){};
  double value;
  AnsiString name, valueName;
};
//---------------------------------------------------------------------------
class TFactor
{
public:
  TFactor(){};
  ~TFactor(){};
  AnsiString name;
  map<AnsiString, TParameter> parameters;
};
//---------------------------------------------------------------------------
class TJudgmentalEvent
{
public:
  TJudgmentalEvent(){ impactFunction = ""; };
  ~TJudgmentalEvent(){};
  TDate eventDate;
  double impact;
  bool futureEvent;
  AnsiString impactFunction;
  TImpactType impactType;
  map<AnsiString, TFactor> factors;
};
//---------------------------------------------------------------------------
class TJudgmentalEventBase
{
public:
  TJudgmentalEventBase(){};
  ~TJudgmentalEventBase(){};
  map<TDate, TJudgmentalEvent> judgmentalEvents;
};
//---------------------------------------------------------------------------
```

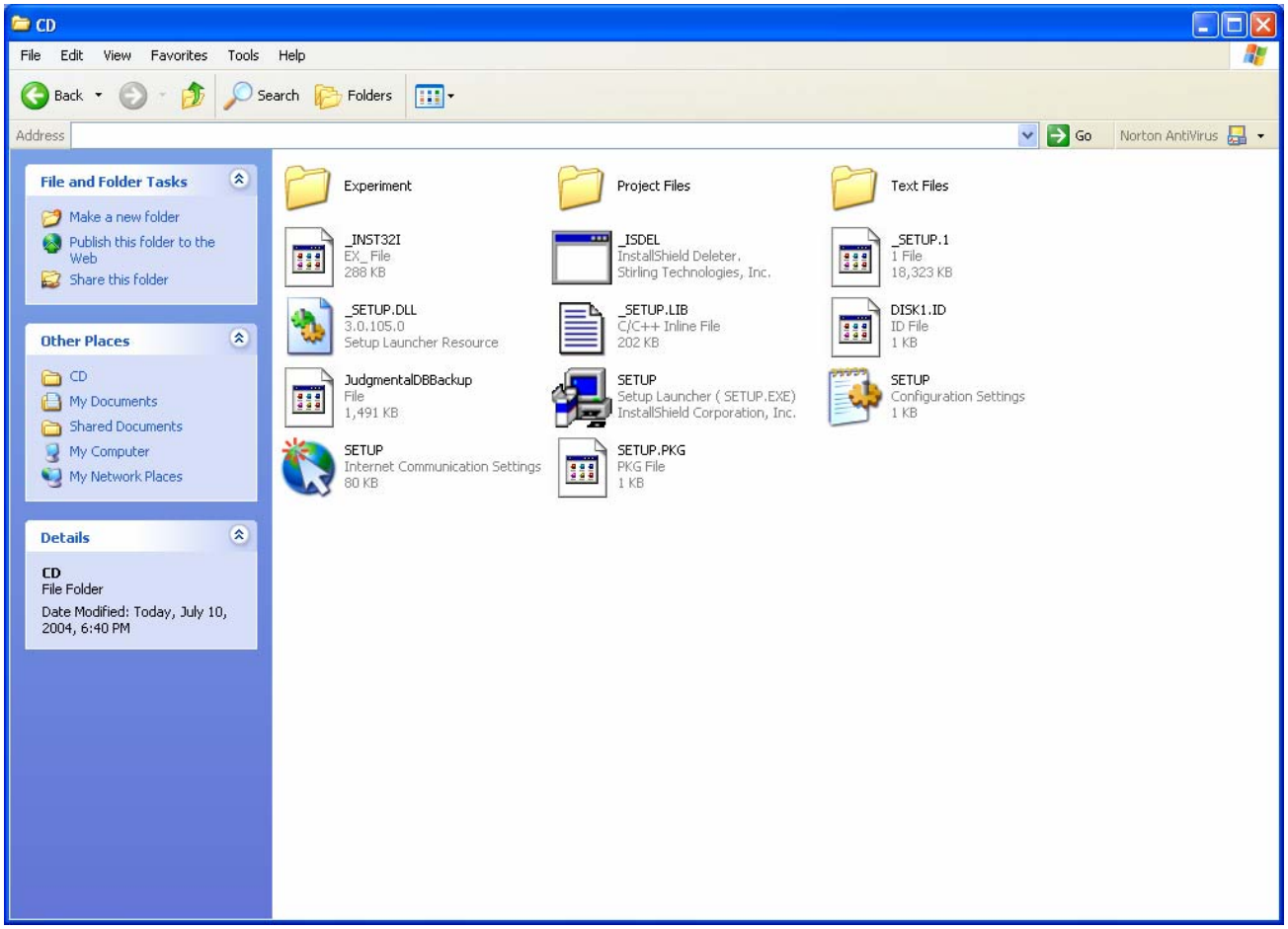
Κεφάλαιο 7

# Οδηγίες Εγκατάστασης

## 7.1 Τα περιεχόμενα του συνοδευτικού CD-ROM

Στο screenshot της επόμενης σελίδας φαίνονται τα αρχεία που περιέχει το συνοδευτικό με τον τόμο CD-ROM. Στον κατάλογο "Project Files" περιέχονται δύο εκδόσεις του πληροφοριακού συστήματος. Η Jforecast και JforecastRelease. Πρόκειται για τα ίδια ακριβώς έργα, με τη μόνη διαφορά ότι το δεύτερο έχει μεταγλωττισθεί με βελτιστοποιήσεις που του δίνουν τη δυνατότητα να εκτελεί τους αλγορίθμους του πληροφοριακού συστήματος γρηγορότερα. Στον κατάλογο με τίτλο Report περιέχεται ο παρόν τόμος σε ηλεκτρονική μορφή ως έγγραφο του Microsoft Word και ως έγγραφο του Adobe Acrobat Reader. Ο κατάλογος με τίτλο Experiment περιέχει δύο αρχεία. Το ένα ονομάζεται data.xls και περιέχει τα δεδομένα (χρονοσειρές) του πειράματος που περιγράφεται στο κεφάλαιο 5 και το δεύτερο ονομάζεται results.xls και περιέχει τα αποτελέσματα του πειράματος.

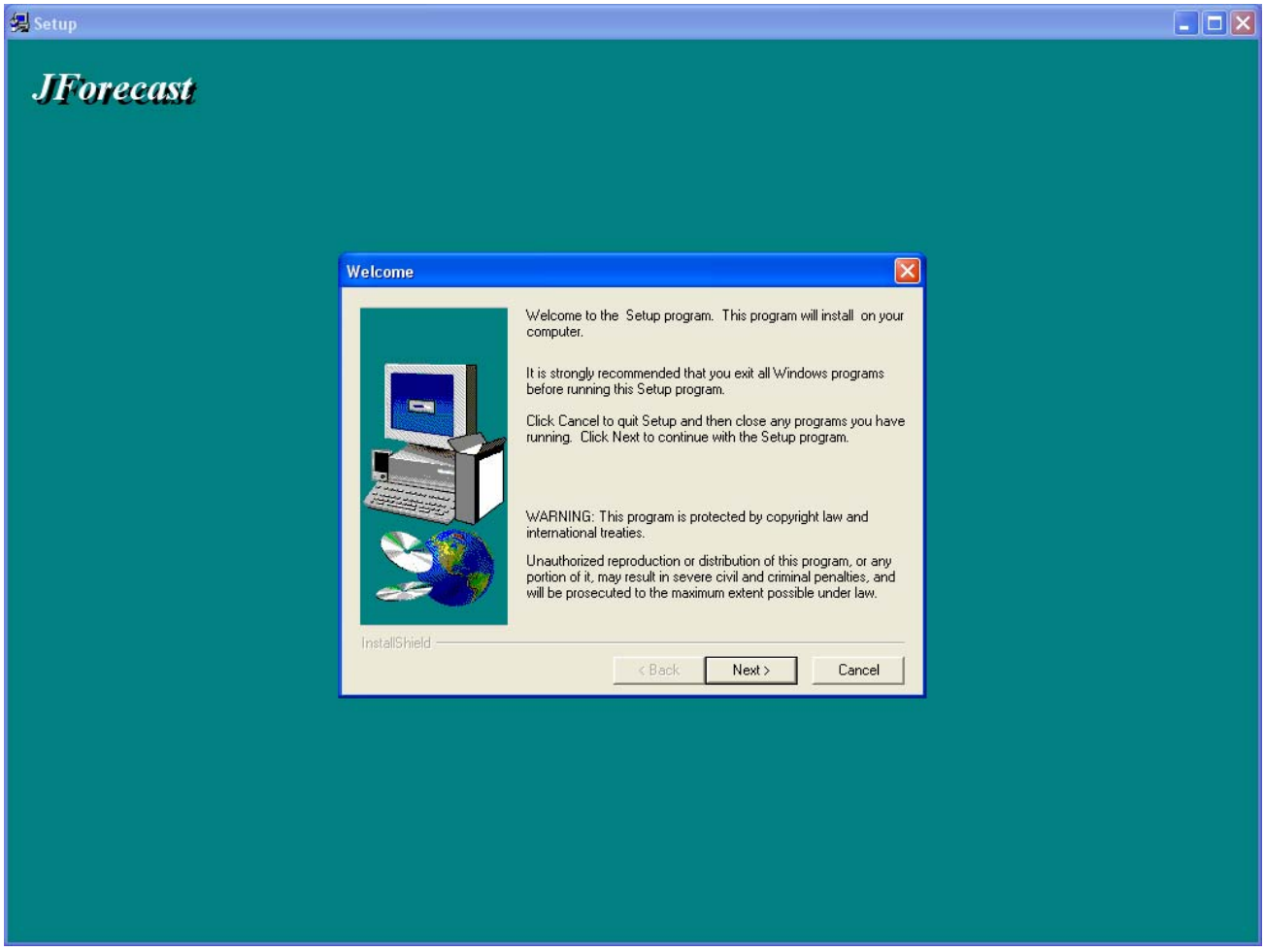
Τέλος στον αρχικό κατάλογο περιέχεται μαζί με άλλα απαραίτητα για την εγκατάσταση αρχεία, το εκτελέσιμο setup.exe με το οποίο γίνεται αυτόματα η εγκατάσταση του πληροφοριακού συστήματος. Στην επόμενη ενότητα θα δούμε πως ακριβώς γίνεται αυτό.


CD
File Edit View Favorites Tools Help
Back Search Folders
Address
Go
Norton AntiVirus
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Share this folder
Other Places
CD
My Documents
Shared Documents
My Computer
My Network Places
Details
CD
File Folder
Date Modified: Today, July 10, 2004, 6:40 PM
Experiment
Project Files
Text Files
_INST32I
EX_ File
288 KB
_ISDEL
InstallShield Deleter.
Stirling Technologies, Inc.
_SETUP.1
1 File
18,323 KB
_SETUP.DLL
3.0.105.0
Setup Launcher Resource
_SETUP.LIB
C/C++ Inline File
202 KB
DISK1.ID
ID File
1 KB
JudgmentalDBBackup
File
1,491 KB
SETUP
Setup Launcher ( SETUP.EXE)
InstallShield Corporation, Inc.
SETUP
Configuration Settings
1 KB
SETUP
Internet Communication Settings
80 KB
SETUP.PKG
PKG File
1 KB

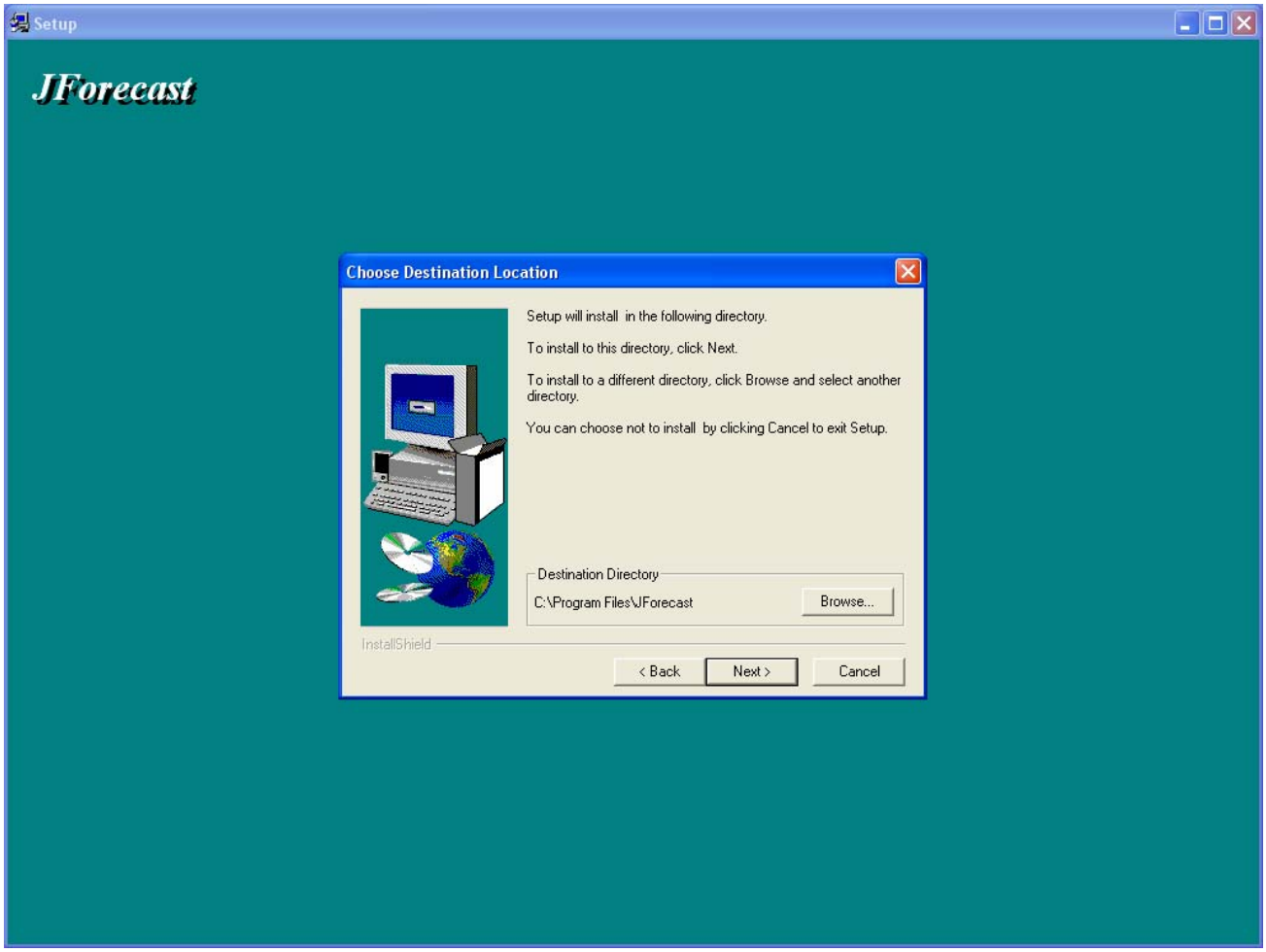
## 7.2 Οδηγίες εγκατάστασης

Τα παρακάτω βήματα περιγράφουν με αναλυτικό τρόπο την εγκατάσταση του πληροφοριακού συστήματος σε οποιοδήποτε λειτουργικό περιβάλλον από τα : Microsoft Windows XP, 2000, 98, 2003 Server

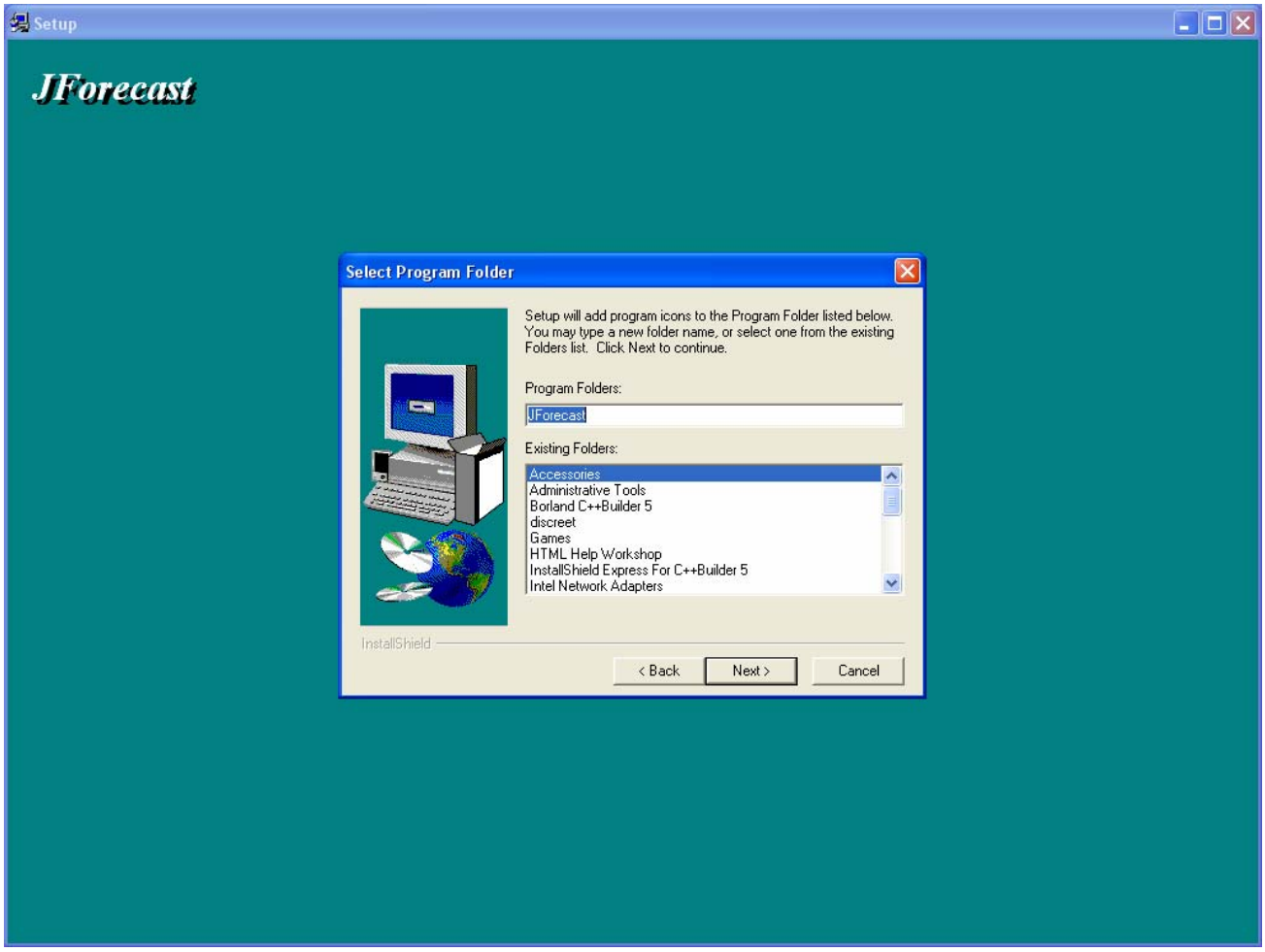
**Βήμα 1ο :** Τοποθετώντας το δίσκο μέσα στο cd-rom περιμένουμε να ξεκινήσει ο οδηγός εγκατάστασης αυτόματα. Σε περίπτωση που δεν ξεκινήσει τον ξεκινάμε τρέχοντας το αρχείο setup.exe που βρίσκεται στο root directory του cd-rom. Στο επόμενο screenshot φαίνεται η πρώτη εικόνα του οδηγού εγκατάστασης.



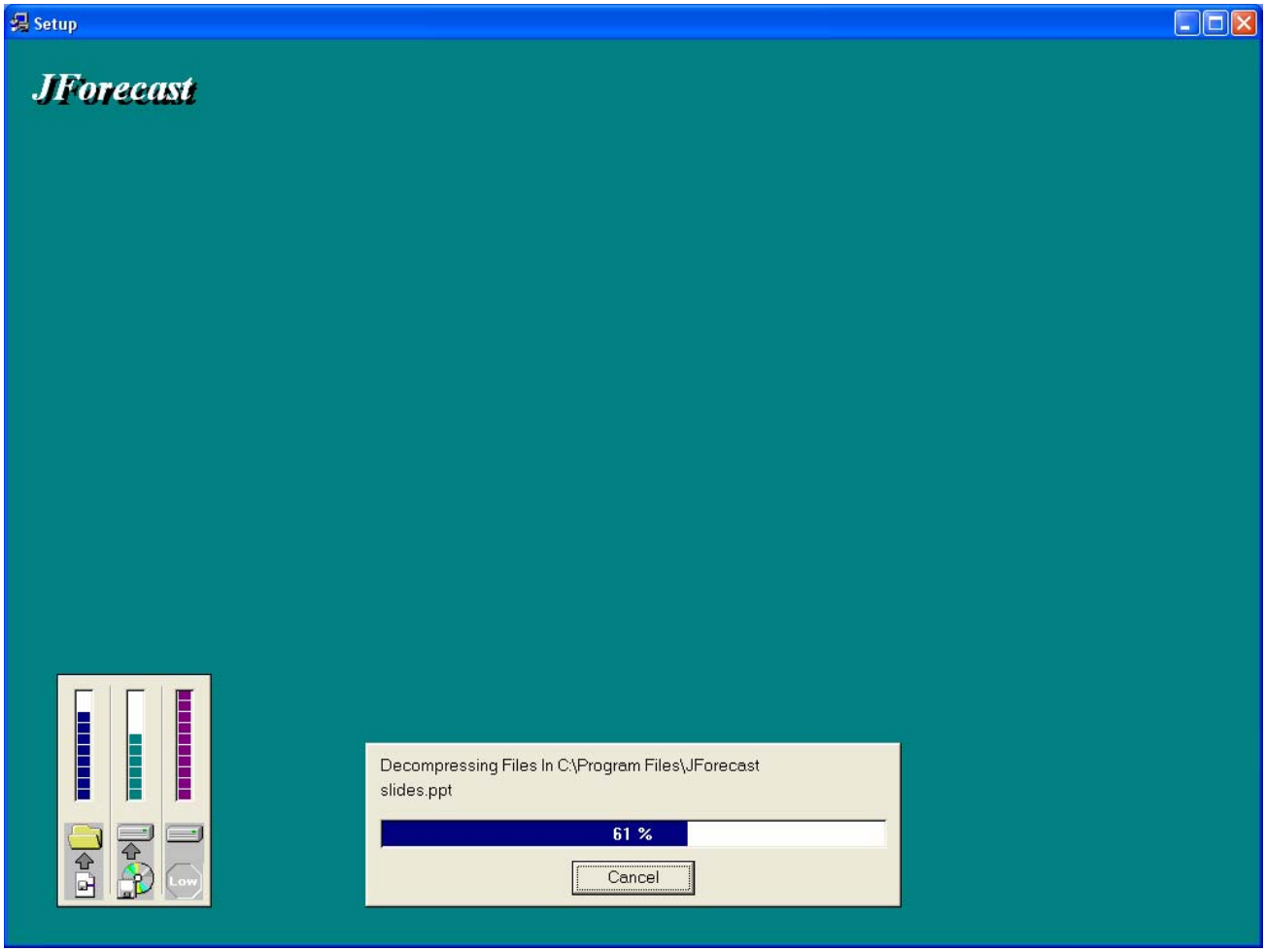
Πατάμε το κουμπί με τίτλο "Next" και εμφανίζεται η επόμενη οθόνη που μας ζητάει να δηλώσουμε την τοποθεσία που θέλουμε να γίνει η εγκατάσταση. Μπορούμε να αφήσουμε και την προεπιλεγμένη τοποθεσία. Στη συνέχεια πατάμε "Next".



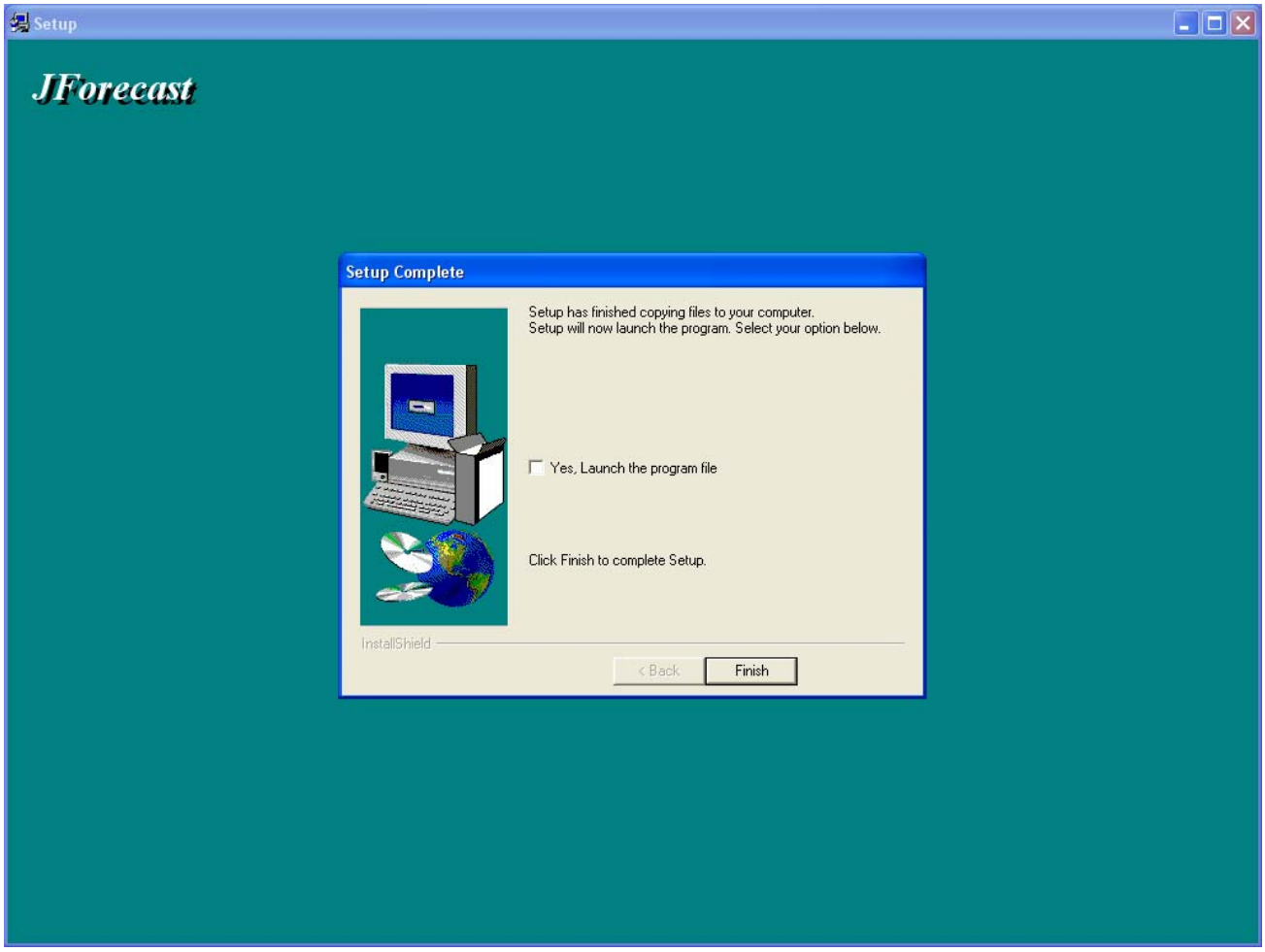

Στην επόμενη οθόνη μας ζητείται να επιλέξουμε το όνομα του καταλόγου μέσα στον οποίο θα δημιουργηθούν οι συντομεύσεις της εφαρμογής στο start menu των windows. Αφού το κάνουμε πατάμε πάλι "Next"




Η επόμενη οθόνη περιέχει μια σύνοψη των πληροφοριών που έχουμε δώσει μέχρι τώρα. Πατάμε και σε αυτήν "Next" και εμφανίζεται η μπάρα προόδου της επόμενης σελίδας.

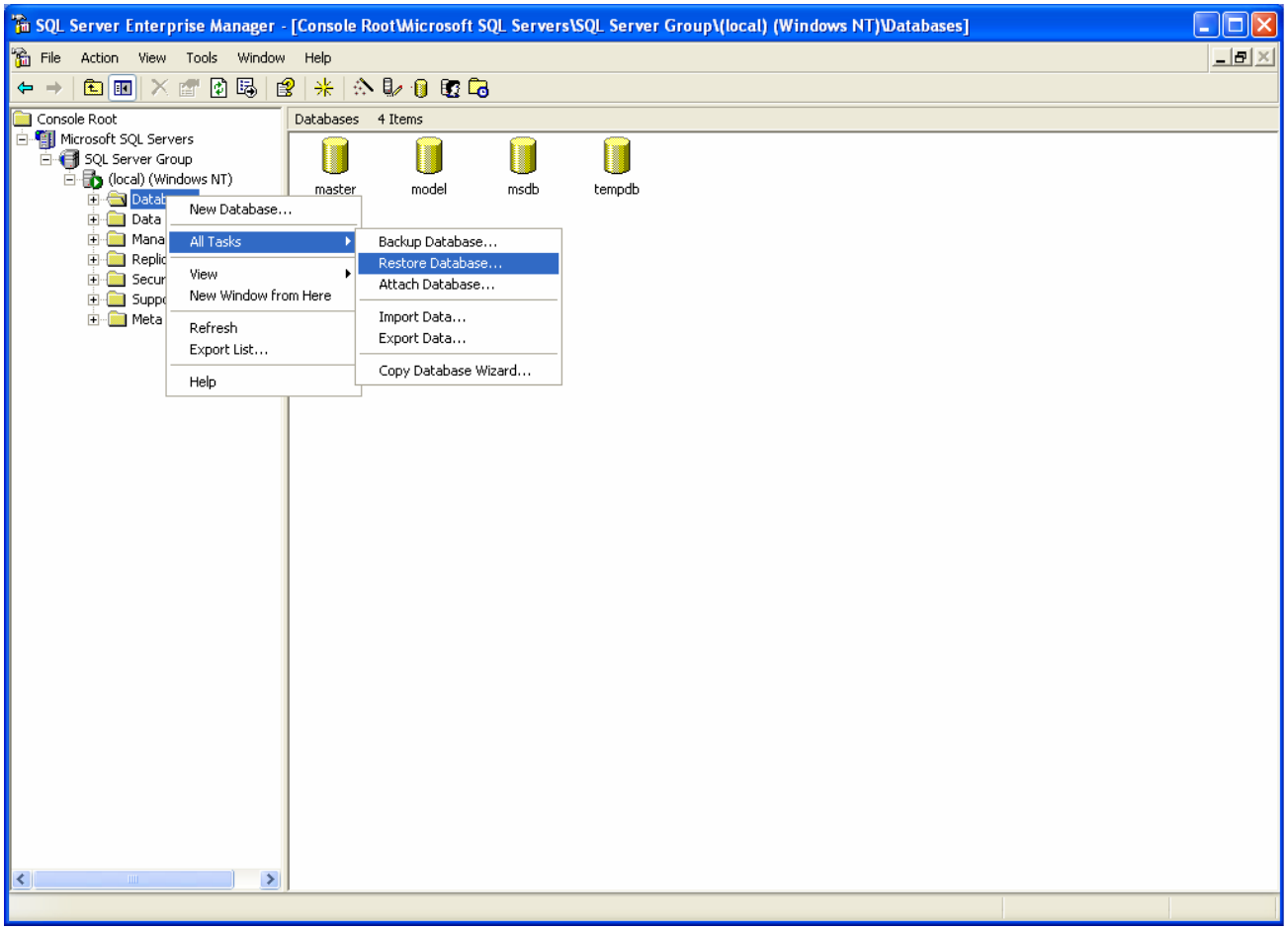



Αφού ολοκληρωθεί η εγκατάσταση εμφανίζεται η οθόνη τερματισμού της επόμενης σελίδας στην οποία αν θέλουμε ενεργοποιούμε την επιλογή να εκτελεστεί η εφαρμογή μετά τον τερματισμό. Καλό είναι όμως να μην την ενεργοποιήσουμε μιας και δε θα λειτουργεί αυτή ακόμα σωστά. Πατάμε «Finish".

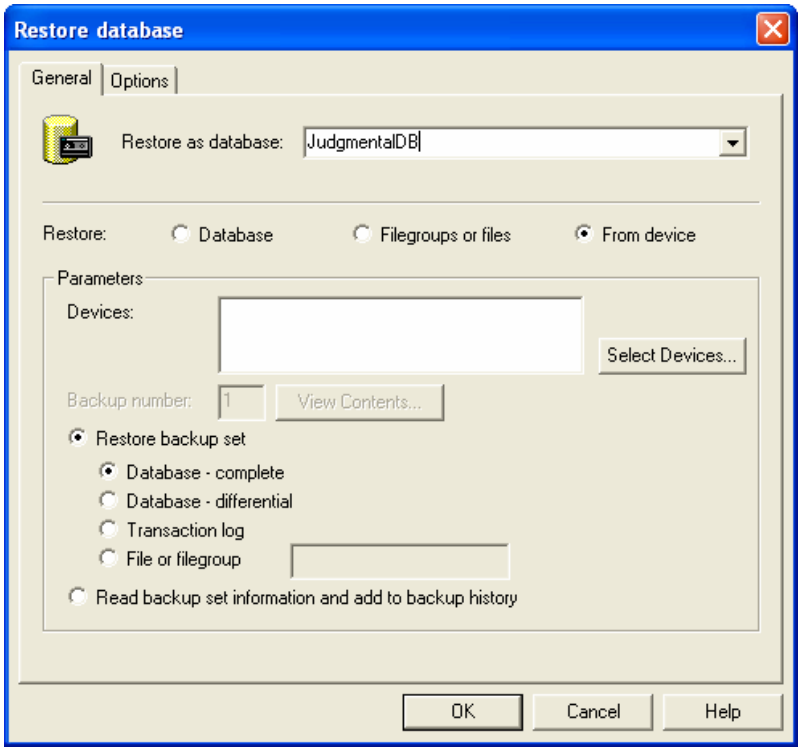



Ένα βασικό μέρος της εγκατάστασης έχει ολοκληρωθεί. Μπορούμε τώρα να δούμε στο start menu τη συντόμευση της εφαρμογής με τον τίτλο Jforecast και τις συντομεύσεις των εγγράφων report.doc , report.pdf , data.xls , results.xls και slides.ppt τα περιεχόμενα των οποίων έχουν αναφερθεί στην εισαγωγή.

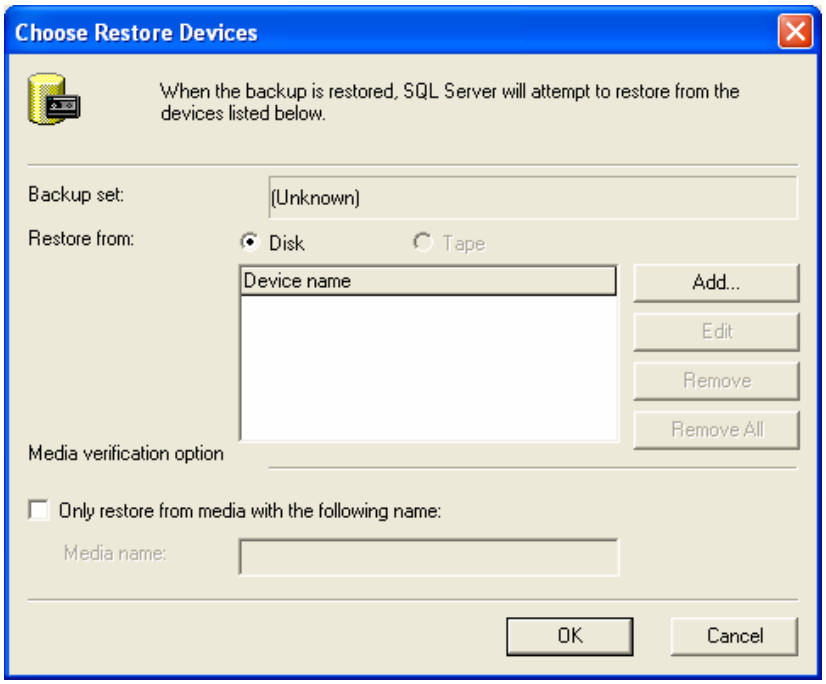
**Βήμα 2ο :** Σε αυτό το βήμα θα προετοιμάσουμε τη βάση δεδομένων. Απαραίτητη προϋπόθεση είναι να έχουμε ήδη εγκατεστημένο τον SQL Server. Αρχικά ανοίγουμε τον Enterprise Manager από το start menu μέσα από τον κατάλογο με τίτλο "Microsoft SQL Server". Στη συνέχεια επιλέγουμε τη λειτουργία Restore Database όπως φαίνεται στο επόμενο screenshot.




Στην επόμενη οθόνη που εμφανίζεται συμπληρώνουμε το πεδίο "Restore as database" με την ονομασία "JudgmentalDB" και ενεργοποιούμε την επιλογή From Device όπως φαίνεται στην επόμενη σελίδα. Τέλος πατάμε το κουμπί Select Devices.

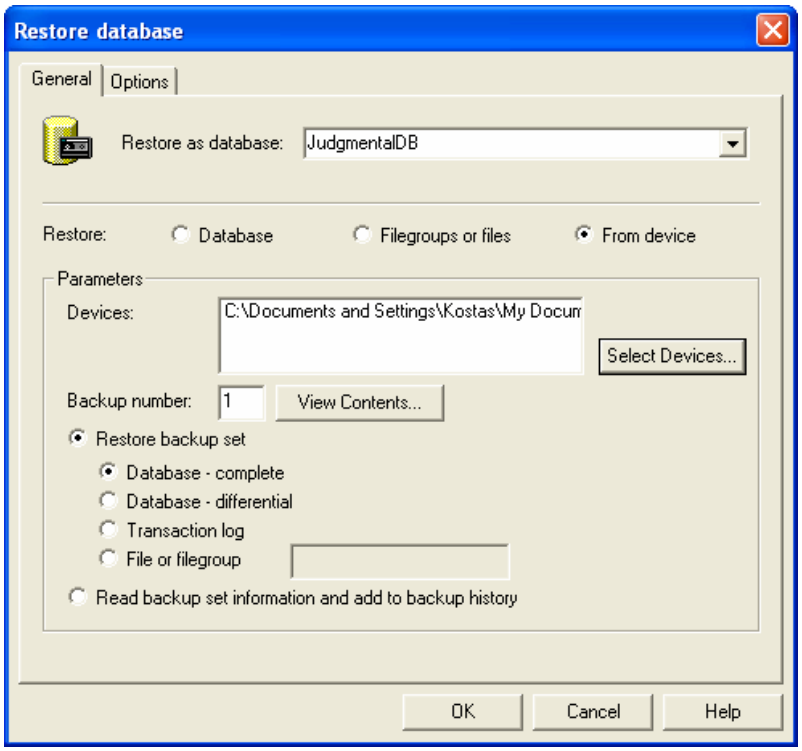



Εμφανίζεται τότε η παρακάτω φόρμα:

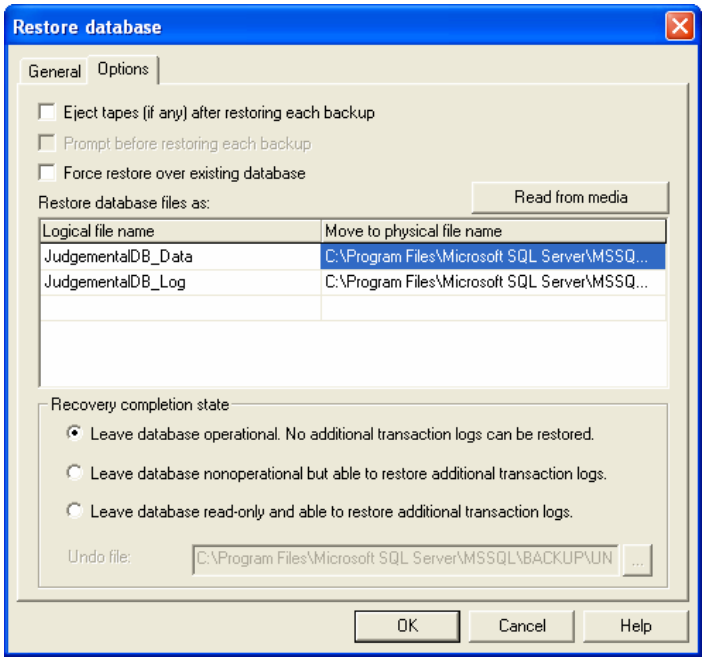


Πατάμε το κουμπί Add… και εμφανίζεται η φόρμα :


Έχοντας ενεργοποιημένη την επιλογή File name πατάμε το κουμπί με τις 3 τελείες και επιλέγουμε το αρχείο με τίτλο JudgmentalDBBackup το οποίο βρίσκεται μέσα στο directory που κάναμε εγκατάσταση την εφαρμογή στο πρώτο βήμα. Στη συνέχεια πατάμε ok μέχρι να βγούμε στην αρχική οθόνη η οποία φαίνεται παρακάτω. Τώρα στο πεδίο με τίτλο Devices γράφεται η τοποθεσία του επιλεγμένου αρχείου.

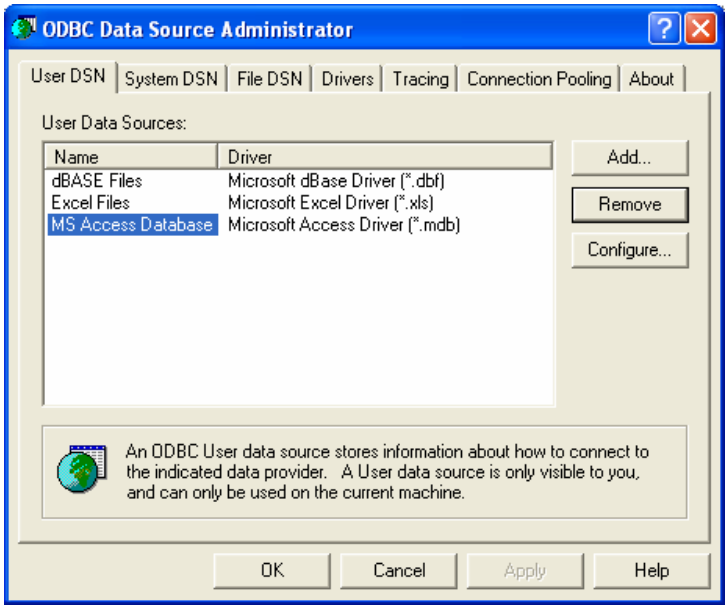


Προσοχή!!! Πατώντας στο tab με τίτλο options βλέπουμε τις τοποθεσίες στις οποίες θα δημιουργηθούνε τα δύο αρχεία της βάσης δεδομένων. Αυτές θα πρέπει να προσέξουμε να είναι ακριβώς ίδιες με αυτές που έχουμε κάνει εγκατάσταση τον SQL Server στο δίσκο μας. Για παράδειγμα αν τον έχουμε κάνει εγκατάσταση στη μονάδα D θα πρέπει να αλλάξουμε την αρχική τοποθεσία από C σε D. Το ίδιο ισχύει και για τον τελικό κατάλογο της εγκατάστασης του SQL Server ο οποίος θα πρέπει να είναι ακριβώς ο ίδιος με αυτόν στο δίσκο μας.

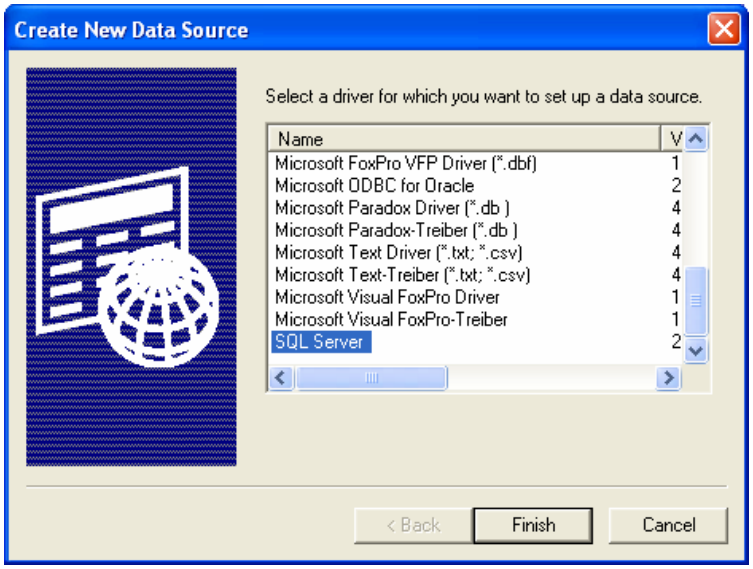



Στη συνέχεια πατάμε ok και δημιουργούμε τη Βάση Δεδομένων. Η λειτουργία έχει ολοκληρωθεί με επιτυχία όταν εμφανίζεται το αντίστοιχο μήνυμα.

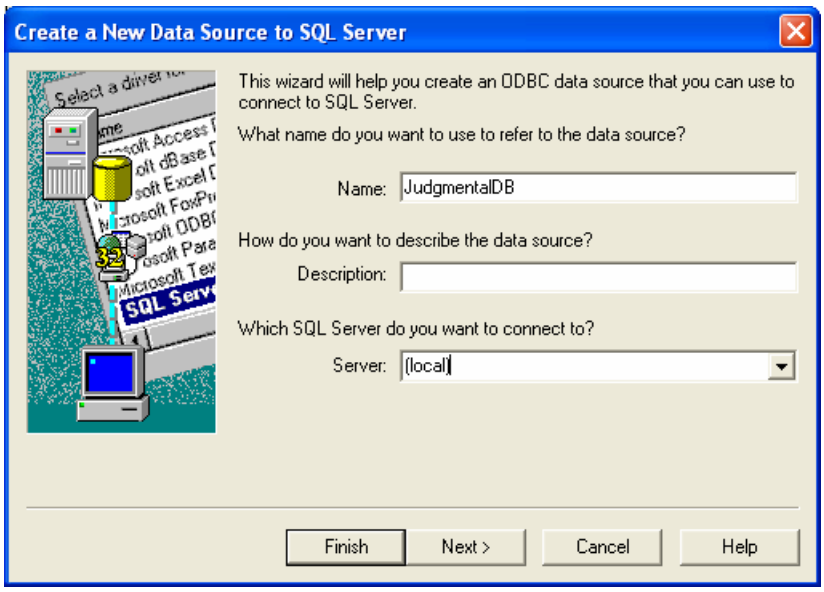
**Βήμα 3ο :** Τελικές ρυθμίσεις. Μέσα από το Control Panel τρέχουμε τη συντόμευση Data Sources (ODBC). Εμφανίζεται η ακόλουθη φόρμα. Έχοντας ενεργοποιημένο το tab User DSN πατάμε το κουμπί Add…



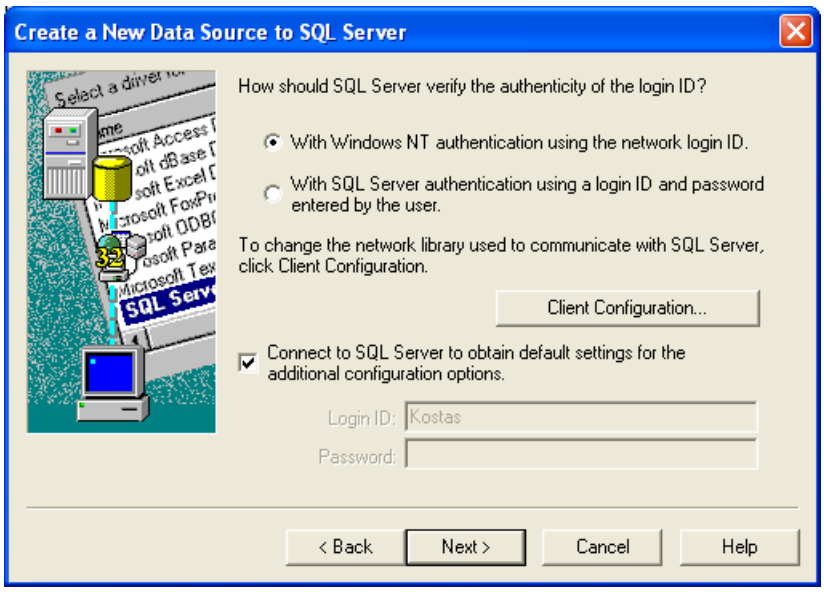

Στη συνέχεια επιλέγουμε SQL Server και πατάμε finish όπως φαίνεται παρακάτω:



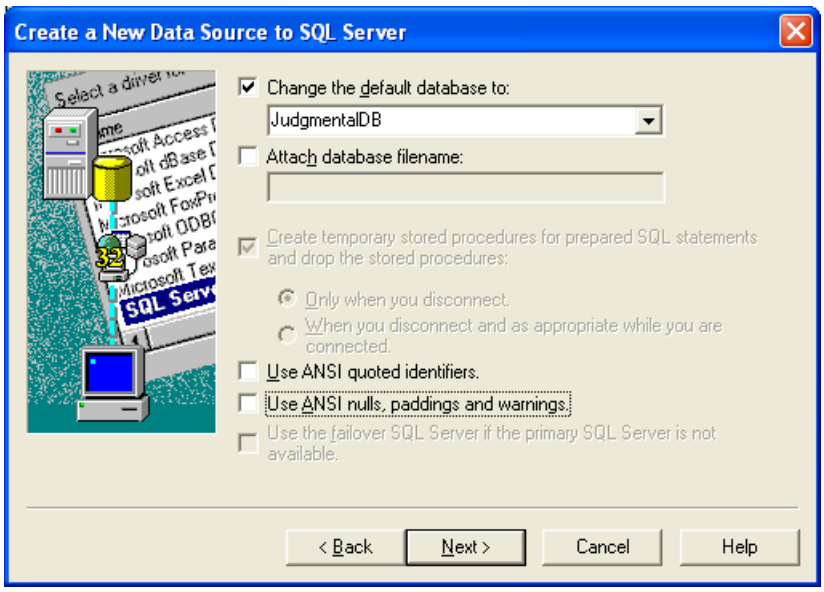

Στη φόρμα της επόμενης σελίδας στο πεδίο Name συμπληρώνουμε JudgmentalDB και στο πεδίο Server γράφουμε (local) μαζί με τις παρενθέσεις. Στη συνέχεια πατάμε Next.



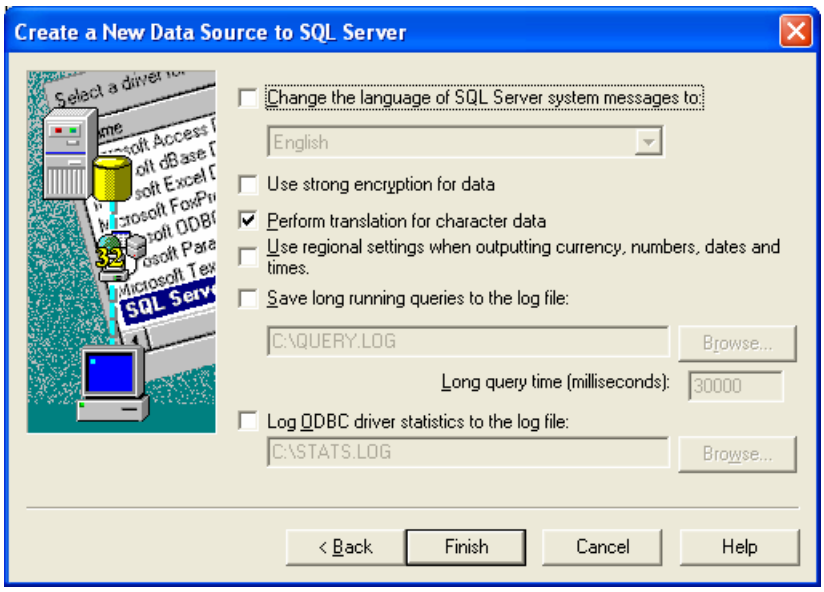

Εμφανίζεται η παρακάτω φόρμα την οποία αφήνουμε ως έχει και πατάμε πάλι next.



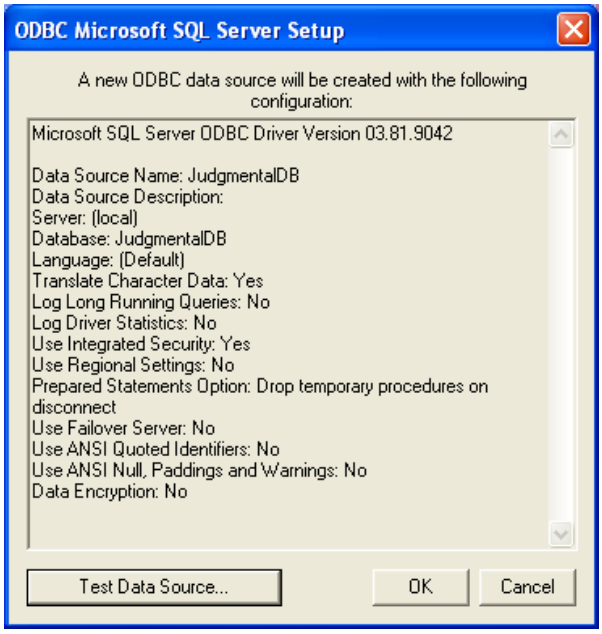

Στην επόμενη φόρμα ενεργοποιούμε την επιλογή Change the default database to: και από το drop down box επιλέγουμε JudgmentalDB. Αν δεν υπάρχει αυτή η επιλογή τότε δεν έχουμε ολοκληρώσει με επιτυχία το βήμα 2. Απενεργοποιούμε τις επιλογές Use Ansi quoted identifiers και Use Ansi nulls, paddings and warnings όπως φαίνεται στην επόμενη σελίδα. Πατάμε next.




Η επόμενη φόρμα εμφανίζεται παρακάτω. Την αφήνουμε ως έχει και πατάμε finish.




Τέλος εμφανίζεται η φόρμα της επόμενης σελίδας στην οποία μπορούμε να ελέγξουμε τη σύνδεση με τη βάση δεδομένων πατώντας το κουμπί Test Data Source. Πατάμε ok και η εγκατάσταση ολοκληρώνεται. Τώρα μπορούμε να τρέξουμε την εφαρμογή από το start menu των windows.


ODBC Microsoft SQL Server Setup
A new ODBC data source will be created with the following configuration:
Microsoft SQL Server ODBC Driver Version 03.81.9042
Data Source Name: JudgmentalDB
Data Source Description:
Server: (local)
Database: JudgmentalDB
Language: (Default)
Translate Character Data: Yes
Log Long Running Queries: No
Log Driver Statistics: No
Use Integrated Security: Yes
Use Regional Settings: No
Prepared Statements Option: Drop temporary procedures on disconnect
Use Failover Server: No
Use ANSI Quoted Identifiers: No
Use ANSI Null, Paddings and Warnings: No
Data Encryption: No
Test Data Source...
OK
Cancel

Κεφάλαιο 8

# ΒΙΒΛΙΟΓΡΑΦΙΑ

1) Λουκάς Χ. Παπαγγελής - Παντελής Σ. Στακιάς, Μέθοδοι Κριτικών Προβλέψεων, Διπλωματική Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Τμήμα Ηλεκτρολόγων Μηχανικών Και Μηχανικών Υπολογιστών, Τομέας Ηλεκτρικών Βιομηχανικών Διατάξεων Και Συστημάτων Αποφάσεων, Μονάδα Προβλέψεων Και Προοπτικής (2002) (Κεφάλαια 1 και 2)

2) Simon Haykin, Neural Networks A Comprehensive Foundation Second Edition (1999)

3) John O. Rawlings, Applied Regression Analysis A Research Tool (1988)

4) Jarrod HollingWorth, Dan Butterfield, Bob Swart, Jamie Allsop, C++ Builder 5 Developers Guide (2001)

5) Judgmental adjustment in time series forecasting using neural networks, Jae Kyu Lee, Chang Seon Yum, Decision Support Systems 22 (1998) 135-154, Elsevier.