Υποστήριξη πρόβλεψης σε εργαλείο διαχείρισης χρονοσειρών

Όθωνας Γκαβαρδίνας

Διπλωματική Εργασία

Επιβλέπων: Π. Βασιλειάδης

Ιωάννινα, Μάρτιος 2020

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κ. Βασιλειάδη Παναγιώτη για την ευκαιρία που μου προσέφερε να ασχοληθώ με ένα τόσο ενδιαφέρον πεδίο όπως είναι η επεξεργασία και ανάλυση χρονοσειρών, καθώς και για την βοήθεια, υποστήριξη και καθοδήγηση που μου παρείχε.

4/3/2020

Γκαβαρδίνας Όθωνας

Περίληψη στα ελληνικά

Τα δεδομένα, που συλλέγονται καθημερινά από χρήστες και οργανισμούς, για διάφορους σκοπούς, εμπεριέχουν συνήθως την έννοια της χρονικής σειράς. Κάθε τιμή μιας σειράς δεδομένων που καθορίζεται από κάποια χρονική στιγμή, έχει προηγούμενη και επόμενη τιμή. Η ανάλυση των δομών που σχηματίζει ο χρόνος απαιτεί ιδιαίτερη μεταχείριση, καθώς η χρονική πληροφορία είναι αρκετά χρήσιμη. Με βάση την εφαρμογή, τα δεδομένα εμφανίζονται άλλοτε στο συνεχή χώρο ως χρονοσειρές και άλλοτε στο διακριτό χώρο ως ακολουθίες. Και στους δύο χώρους ακολουθούνται γενικές ή και ειδικές τεχνικές της επεξεργασίας δεδομένων. Ιδιαίτερο γνώρισμα των συνεχών δεδομένων είναι η ικανότητα πρόβλεψης νέων τιμών. Με κατάλληλο μετασχηματισμό, τα δεδομένα αποκτούν τις απαραίτητες ιδιότητες για παραγωγή καλών προβλέψεων. Ακόμη, τα δεδομένα αποκτούν ταίριασμα με ένα μοντέλο παραγωγής προβλέψεων. Η παραγωγή προβλέψεων χρησιμοποιώντας κάποιο μοντέλο, είναι το πεδίο που απασχολεί αυτή τη διπλωματική εργασία και συγκεκριμένα τα μοντέλα πρόβλεψης ARIMA. Τα μοντέλα ARIMA χρησιμοποιούνται κατά κόρον στην παραγωγή προβλέψεων και είναι γνωστά για τις ακριβείς προβλέψεις τους. Σε αυτή τη Διπλωματική Εργασία, αρχικά περιγράφουμε το θεωρητικό υπόβαθρο των δεδομένων συνάρτηση του χρόνου. Στη συνέχεια, καταγράφουμε υπάρχουσες υλοποιήσεις βιβλιοθηκών, και τέλος, παρουσιάζουμε τη δομή και τη λειτουργία του εργαλείου που κατασκευάσαμε.

Λέξεις Κλειδιά: χρονοσειρά, ακολουθία, εξόρυξη δεδομένων, πρόβλεψη, ARIMA

Abstract

Data, which are collected from different users and organizations, are usually associated with time. Each value in these data, which is associated with a specific time, has a previous time value and also, an after time value. The analysis of these data needs special treatment, as every time instant is important. These data can take the form of a timeseries in a continuous space, or the form of a sequence in a discrete space. Special techniques of data analysis can be used for both spaces. A unique component in timeseries representation is the capability to predict new values. After the right transformation, the timeseries' data may consist of the best attributes to produce useful forecasts. Also, the timeseries' data should fit with a forecast model. Forecasting by using a model is the main process that this diploma analyzes, and especially the ARIMA forecast models. ARIMA forecast models are widely used for forecasting, as they are well known for their precise predictions. At first, in this Diploma Thesis, we describe the data associated with time in theory. Subsequently, we record existing library implementations, and at last, we present the structure and the functionality of the software we created.

Keywords: timeseries, sequence, data mining, forecasting, forecast, prediction, ARIMA

Πίνακας περιεχομένων

Κεφάλαιο 1. Εισαγωγή

1.1 Αντικείμενο της διπλωματικής

Τα δεδομένα είναι το επίκεντρο του ενδιαφέροντος της τελευταίας δεκαετίας. Κάθε λογισμικό και υπηρεσία εξαγάγει και επεξεργάζεται πληροφορίες από τους χρήστες της. Υπάρχουν κάποιες εξαιρέσεις, μα συνήθως τα δεδομένα είναι αυτά που αποτελούν την κινητήρια δύναμη του κέρδους στο λογισμικό. Από τα δεδομένα, μπορούν να παραχθούν συμπεράσματα και από τα συμπεράσματα, να δημιουργηθούν διάφορες διαγνώσεις και να γίνει κατανόηση συμπεριφορών που αφορούν, από ανθρώπινες δραστηριότητες, μέχρι και την κίνηση των πλανητών.

Η εργασία κινείται γύρω από το επιστημονικό πεδίο των δεδομένων που συνδέονται με τιμές χρόνου. Τα δεδομένα αυτά παρουσιάζονται συχνά μέσω γραφημάτων ακμών και κάποιες ενδεικτικές χρήσεις τους είναι στη στατιστική, στην αναγνώριση μοτίβων, στην πρόγνωση καιρού και στη μηχανική, γι' αυτό τα χρονικά δεδομένα έχουν σίγουρα μια πολύτιμη θέση στην εξόρυξη δεδομένων. Η σύνθετη δομή αυτών των δεδομένων απαιτεί ιδιαίτερη μεταχείριση. Τα χρονικά δεδομένα ακολουθούν σύνθετες έννοιες, οι οποίες περιγράφονται, στη συνέχεια, με σαφήνεια, περιεκτικότητα και ακρίβεια στον αναγνώστη.

Υπάρχουν αρκετά εργαλεία για την ανάλυση και επεξεργασία των χρονικών δεδομένων. Είναι σημαντικά τα εργαλεία διαχείρισης χρονοσειρών, καθώς διευκολύνουν τους αναλυτές, οι οποίοι κατέχουν στα χέρια τους μεγάλο πλήθος δεδομένων που συσχετίζονται μεταξύ τους και πρέπει από αυτά να εξάγουν αποφάσεις. Το πρόβλημα όμως είναι ότι ο χειρισμός των υπαρχόντων εργαλείων είναι δύσκολος, καθώς απαιτούν μεγάλο υπόβαθρο και είναι πολύ εξειδικευμένα. Ιδιαίτερα, όταν πρόκειται για πρόβλεψη τιμών, που ανήκουν σε ένα συγκεκριμένο εύρος, με κάποια πιθανότητα, ο αναλυτής θα πρέπει να είναι αρκετά σίγουρος για τα δεδομένα του, πριν την εξαγωγή συμπερασμάτων από αυτά.

Στόχος της εργασίας, είναι η εκμετάλλευση υπαρχόντων βιβλιοθηκών, για την παραγωγή μιας πλήρους εφαρμογής, εύχρηστης και φιλικής προς κάθε χρήστη που επιθυμεί να παραγάγει προβλέψεις που αφορούν χρονικά δεδομένα. Συγκεκριμένα, σε αυτή την εργασία δημιουργούμε μία εφαρμογή για την πρόβλεψη μελλοντικών τιμών δεδομένων χρονοσειρών με μοντέλα ARIMA. Για το σκοπό αυτό, γίνεται η κωδικοποίηση σε βήματα της διαδικασίας πρόβλεψης με μοντέλα ARIMA. Η εφαρμογή, είναι σχηματισμένη έτσι, ώστε να υποστηρίζει τα βήματα αυτής της διαδικασίας. Μέσω μιας γραφικής διεπαφής χρήστη, διαθέτει τη δυνατότητα μετασχηματισμού χρονοσειρών και δημιουργίας μοντέλων που ταιριάζουν σε αυτές, με σκοπό την παραγωγή προβλέψεων μελλοντικών τιμών. Η εφαρμογή διατίθεται ηλεκτρονικά στο σύνδεσμο https://github.com/othonasgkavardinas/java_timeseries_arima.

Μαζί με την εφαρμογή δίνονται τα κατάλληλα εφόδια στον αναγνώστη. Ακόμη, εκτελούνται έλεγχοι της εφαρμογής και παρουσιάζονται τα αποτελέσματά τους. Ιδιαίτερα, η αξιολόγηση της εργασίας γίνεται μέσα από ένα παράδειγμα χρήσης της, και τα αποτελέσματα αποδεικνύουν ότι πράγματι, εκτελώντας μια προκαθορισμένη ακολουθία λειτουργιών, μπορεί ένας αναλυτής να παράγει προβλέψεις για δεδομένα χρονοσειρών με το συγκεκριμένο εργαλείο.

1.2 Περιγραφή κεφαλαίων

Στην εργασία αυτή περιλαμβάνονται τέσσερα επιπλέον κεφάλαια με σκοπό την κάλυψη θεωρίας και εννοιών απαραίτητων για την κατανόηση του αντικειμένου. Επίσης, δίνεται η σχεδίαση και η λειτουργικότητα της εφαρμογής, καθώς και η αξιολόγησή της. Τα περιεχόμενα των κεφαλαίων παρουσιάζονται αναλυτικά στη συνέχεια.

Στο **Κεφάλαιο 2** παρουσιάζονται χρήσιμες έννοιες και ορισμοί που αφορούν το αντικείμενο που μελετάται και έτσι, δίνεται το κατάλληλο υπόβαθρο, γύρω από τις δύο σημαντικές μορφές των χρονικών δεδομένων. Οι μορφές αυτές δεν είναι άλλες από τις χρονοσειρές και τις ακολουθίες. Η πρώτη αφορά δεδομένα του συνεχή χώρου, ενώ η δεύτερη του διακριτού χώρου. Κάθε μία από τις δύο φέρει ξεχωριστή μεταχείριση και έχει ενδιαφέρον η ανάλυση των διαφόρων προβλημάτων και τεχνικών που τις αφορούν. Για παράδειγμα, η κατηγοριοποίηση (classification), είναι ένα πρόβλημα που αφορά και τους δύο τύπους δεδομένων, όμως χειρίζεται με διαφορετικό και σύνθετο στην περιγραφή τρόπο.

Το **Κεφάλαιο 3** αποτελεί το πρακτικό - σχεδιαστικό κομμάτι της εφαρμογής. Αρχικά, ορίζεται το πρόβλημα της πρόβλεψης χρονοσειρών με μοντέλα ARIMA. Για το πρόβλημα,

ορίζεται η ακολουθία βημάτων της μεθόδου ARIMA σε μορφή ψευδοκώδικα και δίνεται η σχηματική αναπαράσταση της. Στη συνέχεια, περιγράφεται λεπτομερώς η διαδικασία επιλογής των απαραίτητων βιβλιοθηκών. Μετά την επιλογή της κατάλληλης βιβλιοθήκης, γίνεται γνωστός ο τρόπος εγκατάστασης της και δίνεται σχηματική αναπαράσταση των τμημάτων που χρησιμοποιήθηκαν από αυτή. Ακόμη, σε αυτό το σημείο περιγράφεται και η αρχιτεκτονική του λογισμικού της διπλωματικής. Ύστερα, προβάλλονται οι έλεγχοι των λειτουργικοτήτων του λογισμικού, και τέλος, εκφράζεται ο τρόπος εγκατάστασής του, μαζί με κάποιες λεπτομέρειες και κάποιες θεμιτές επεκτάσεις που θα μπορούσαν να γίνουν.

Το **Κεφάλαιο 4** πραγματεύεται την απόδειξη της ορθής λειτουργίας του λογισμικού που δημιουργείται, και το σωστό τρόπο χρήσης του. Την απόδειξη αποτελεί ένα αναλυτικό παράδειγμα. Με αυτό το παράδειγμα παρουσιάζεται ο τρόπος χρήσης της εφαρμογής και γίνεται η αξιολόγησή της.

Το **Κεφάλαιο 5** διαθέτει τη σύνοψη των επιτευγμάτων που επέφερε η συγκεκριμένη διπλωματική εργασία, και ποια συμπεράσματα εξήχθησαν τελικά από αυτή. Ακόμη, μέρος του κεφαλαίου αποτελεί ένα σύνολο ιδεών για τη μελλοντική επέκταση του παρόντος αντικειμένου.

Κεφάλαιο 2. Υπόβαθρο

Σε αυτό το κεφάλαιο δίνεται το κατάλληλο υπόβαθρο για τα χρονικά δεδομένα. Παρουσιάζονται πιο αναλυτικά οι δύο κατηγορίες χρονικών δεδομένων, οι χρονοσειρές και οι ακολουθίες. Για κάθε μία περιγράφονται μέθοδοι ανάλυσης και επεξεργασίας αυτών.

2.1 Χρονικά δεδομένα

Σύμφωνα με το [Wiki1] τα χρονικά δεδομένα είναι ακολουθίες δεδομένων, που λαμβάνονται σε χρονικά διαστήματα ίσου μήκους και μπορούν εκφραστούν είτε σε μορφή διακριτής ακολουθίας, είτε σε μορφή συνεχούς χρονοσειράς. Όλοι οι ορισμοί και οι πληροφορίες αυτής της ενότητας προέρχονται από το [AggCC15]. Όπου χρησιμοποιούνται άλλες πηγές, αναφέρονται συμπληρωματικά.

2.1.1 Χρονοσειρές (Timeseries)

Ο χρόνος θεωρείται πως είναι συνεχής και ισομορφικός του χώρου των πραγματικών αριθμών $\mathbb{R}$, και κατά συνέπεια, ενώ διατίθεται ένα διακριτό σύνολο μετρήσεων, υπονοείται ότι υπάρχουν άπειρες τιμές του $\mathbb{R}$, ανάμεσα στις τιμές αυτού του συνόλου. Οι χρονοσειρές εκφράζονται με συνεχείς καμπύλες και οι μετρήσεις τους μπορούν να προκύψουν από αισθητήρες, ιατρικές ή χρηματοοικονομικές και άλλες εφαρμογές. Ακόμη, έχουν δύο ειδών γνωρίσματα, τα contextual, που αφορούν το χρόνο και τα behavioral, που παρατηρούνται στο χρόνο. Με βάση τα γνωρίσματα τους, οι χρονοσειρές διακρίνονται σε μονοδιάστατες και πολυδιάστατες.

- Μονοδιάστατες (Univariate): Οι μονοδιάστατες χρονοσειρές αποτελούνται από ένα behavioral και ένα contextual γνώρισμα. Οι μονοδιάστατες χρονοσειρές καλούνται “διάστασης 1”.

- Πολυδιάστατες (Multivariate): Οι πολυδιάστατες χρονοσειρές αποτελούνται από d behavioral γνωρίσματα και ένα contextual γνώρισμα. Οι πολυδιάστατες χρονοσειρές καλούνται "διάστασης d".

Και οι δύο κατηγορίες χρονοσειρών περιλαμβάνουν n χρονοσφραγίδες. Με τον όρο χρονοσφραγίδα αναφερόμαστε στην τιμή μίας χρονοσειράς μία συγκεκριμένη χρονική στιγμή. Για πολυδιάστατες χρονοσειρές η χρονοσφραγίδα αποτελεί διάνυσμα τιμών. Το διάνυσμα μιας χρονοσφραγίδας είναι $\overline{Y_i} = (y_i^1 \dots y_i^d)$.

2.1.1.1 Ανάλυση χρονοσειρών

Για την ανάλυση των χρονοσειρών, σχηματίζονται μοντέλα, τα οποία μπορούν να διαχωριστούν σε δύο βασικές κατηγορίες.

- Real-time (Πραγματικού χρόνου): Στα μοντέλα ανάλυσης χρονοσειρών πραγματικού χρόνου χρησιμοποιείται ένα μικρό παράθυρο χρόνου, όπου γίνεται η διαδικασία της ανάλυσης. Επίσης, στα μοντέλα αυτά, εφαρμόζονται τεχνικές όπως πρόβλεψη μελλοντικών τιμών, deviation detection, και event detection.
- Retrospective: Στα retrospective μοντέλα ανάλυσης χρονοσειρών περιλαμβάνονται δεδομένα που έχουν συλλεχθεί σε διαφορετικές χρονικές περιόδους στο παρελθόν. Τα δεδομένα αυτά βρίσκονται αποθηκευμένα σε κάποια βάση δεδομένων.

Η ανάλυση δεδομένων χρονοσειρών απαιτεί τα δεδομένα να βρίσκονται σε κατάλληλη μορφή, και για το σκοπό αυτό είναι απαραίτητη η χρήση κάποιων μετασχηματισμών, όταν αυτό χρειάζεται. Οι μετασχηματισμοί εφαρμόζονται σε περιπτώσεις που λείπουν κάποιες ενδιάμεσες τιμές των χρονοσειρών, όταν σε αυτές υπάρχει θόρυβος ή όταν τα δεδομένα δε βρίσκονται σε κάποια κοινά όρια. Στη συνέχεια, αναλύονται περεταίρω τα είδη μετασχηματισμών.

Έλλειψη ενδιάμεσων τιμών (missing values)

Για να γίνει σωστά η ανάλυση των χρονοσειρών πρέπει τα διαστήματα χρόνου να είναι σταθερά, και επίσης να είναι ίδια για τα διάφορα behavioral γνωρίσματα που θα εξετάζονται. Συχνά, στις χρονοσειρές δεν είναι γνωστές κάποιες ενδιάμεσες τιμές. Για την εύρεση αυτών των ενδιάμεσων τιμών χρησιμοποιείται η Γραμμική Παρεμβολή.

$$Y = y_i + \left(\frac{t-t_i}{t_j-t_i}\right) \cdot (y_j - y_i),$$

όπου y_i και y_j είναι οι τιμές της χρονοσειράς στις χρονικές στιγμές t_i και t_j, $i < j$ και $t \in (t_i, t_j)$.

Επίσης, οι ενδιάμεσες τιμές μπορούν να βρεθούν με την Πολυωνυμική Παρεμβολή και spine παρεμβολή.

Αφαίρεση θορύβου (noise removal)

Οι χρονοσειρές περιέχουν θόρυβο που μπορεί να αντιμετωπισθεί με τη χρήση εξομάλυνσης. Κατά την εξομάλυνση αντιμετωπίζεται το πρόβλημα διαχωρισμού του θορύβου και των ενδιαφέροντών ακραίων τιμών (outliers), οντότητες που παρουσιάζουν κάποιες ομοιότητες. Για την αφαίρεση του θορύβου σε δεδομένα χρονοσειρών εφαρμόζεται κάποια από τις παρακάτω μεθόδους.

- Binning: Η μέθοδος binning διαχωρίζει το συνολικό εύρος χρόνου σε διαμερίσεις μήκους k και για κάθε μία κρατά την ενδιάμεση ή τη μέση τιμή. Η ενδιάμεση τιμή είναι προτιμότερη γιατί αυτή δε δίνει βάρος σε ακραίες τιμές (outliers). Ένα πρόβλημα που εμφανίζεται με τη μέθοδο binning είναι ότι χάνονται σημεία με πληροφορία κατά παράγοντα k. Ονομάζεται και Piecewise Aggregate Approximation (PAA).
- Moving-Average Smoothing (MAS): Η μέθοδος MAS χρησιμοποιεί ένα παράθυρο μήκους k ξεκινώντας από την αρχή της χρονοσειράς και το μετακινεί, κάθε φορά κατά μία θέση. Σε κάθε μετακίνηση υπολογίζει τη μέση τιμή των σημείων του παραθύρου. Με αυτόν τον τρόπο υπολογίζονται πολλά επικαλυπτόμενα παράθυρα. Με τη μέθοδο MAS όμως εμφανίζονται καθυστερήσεις και χάνονται κάποια σημεία σε real-time χρονοσειρές. Επίσης, η μέθοδος αυτή, μπορεί να μετατρέψει μια απότομη αύξουσα μεταβολή, σε απότομη φθίνουσα μεταβολή.
- Exponential Smoothing (ES): Η μέθοδος ES χρησιμοποιεί για τον υπολογισμό κάθε τιμής, την προηγούμενή τιμή. Συγκεκριμένα, η νέα εξομαλυμένη τιμή υπολογίζεται από το εξομαλυμένο αποτέλεσμα όλων ή ενός εύρους k προηγούμενων τιμών, και της νέας τιμής, έχοντας αναθέσει βάρος στην κάθε μια που αθροίζει στο 1. Με αυτό τον τρόπο, μπορεί να δίνεται περισσότερη σημασία στις πρόσφατες τιμές. Η μέθοδος ES είναι η προτιμότερη, αφού δε χάνονται σε αυτή τιμές και δεν εμφανίζονται σε αυτή μεγάλες καθυστερήσεις.

Κανονικοποίηση (Normalization)

Οι χρονοσειρές, και ειδικότερα η κατηγορία των πολυδιάστατων χρονοσειρών διακυμαίνονται σε εύρη τιμών χρόνου που μπορεί να διαφέρουν. Η μεταφορά δύο διαφορετικών χρονοσειρών σε κάποιο κοινό εύρος τιμών επιτυγχάνεται με δύο μεθόδους:

- Range Based: Η μέθοδος range based ομαλοποίησης λειτουργεί για προκαθορισμένο διάστημα (min, max) και το αποτέλεσμά της προκύπτει στο διάστημα (0,1).

  $$y_i' = \frac{yi - \min}{\max - min},$$

  με y_i η παλιά τιμή και y_i' η νέα τιμή

- Standardization: Η μέθοδος περιγράφει πόση είναι η απόκλιση σ κάθε τιμής από τη μέση τιμή μ, υπολογίζοντας το z-score z_i, το οποίο όμως δεν προκύπτει στο εύρος τιμών $[0, 1]$. Η μέθοδος αυτή προτιμάται συνήθως.

  $$z_i = \frac{y_i - \mu}{\sigma}$$

Μετασχηματισμοί (data transformation and reduction)

Οι χρονοσειρές, μπορούν να μετατραπούν σε άλλους τύπους δεδομένων για την αξιοποίηση ειδικές μέθοδοι για την ανάλυσή τους.

- Discrete Wavelet Transformation (DWT): Οι χρονοσειρές μετατρέπονται σε μορφή δεδομένων, όπου μπορούν να εφαρμοστούν μέθοδοι πολυδιάστατων δεδομένων χωρίς να υπάρχει η χρονική σχέση μεταξύ των χρονοσφραγίδων. Κατά τον DWT, οι τιμές μιας χρονοσειράς υποδιαιρούνται συνεχώς, κρατώντας τη μέση τιμή των δεδομένων κάθε φορά, μέχρι να παραμείνουν μόνο κάποιες σημαντικές υποδιαιρέσεις της. Τέλος, είναι δυνατή η αναδημιουργία της χρονοσειράς με ακρίβεια μετά την εφαρμογή των μεθόδων.
- Discrete Fourier Transformation (DFT): Οι χρονοσειρές μετατρέπονται σε ημιτονοειδή δεδομένα. Ο DFT είναι κατάλληλος για χρονοσειρές που παρουσιάζουν περιοδικότητα. Με την εφαρμογή του DFT σε χρονοσειρές προκύπτουν δεδομένα σε μιγαδική μορφή, η οποία με κατάλληλη μετατροπή μπορεί να αναπαρασταθεί ως πραγματική. Κατάλληλοι αλγόριθμοι για την εφαρμογή DFT σε χρονοσειρές είναι ο Fast Fourier Transformation (FFT) και ο Discrete Cosine Transformation (DCT) που είναι ο πιο αποτελεσματικός αλλά και πιο σύνθετος.

- Symbolic Aggregate Approximation (SAX): Οι χρονοσειρές μετατρέπονται σε διακριτές ακολουθίες. Κατά το μετασχηματισμό SAX εφαρμόζεται PAA σε μια χρονοσειρά, και έτσι εξάγεται μία αντιπροσωπευτική τιμή από κάθε ίσο διάστημα χρόνου, και στη συνέχεια γίνεται αντιστοίχιση των διαστημάτων αυτών με σύμβολα ακολουθίας. Το κάθε σύμβολο αντιστοιχεί σε ένα ίσο διάστημα των τιμών των behavioral γνωρισμάτων. Επίσης, με το μετασχηματισμό SAX, μειώνονται σε μεγάλο βαθμό οι διαστάσεις των δεδομένων. Για μεγάλες χρονοσειρές η διαδικασία PAA του SAX γίνεται με Gaussian κατανομή.

2.1.1.2 Πρόβλεψη δεδομένων χρονοσειρών (forecasting)

Η διαδικασία πρόβλεψης είναι αυτή, κατά την οποία αξιοποιούνται τιμές χρονικών δεδομένων από το παρελθόν και το παρόν, για την πρόβλεψη μελλοντικών τιμών. Για τη διαδικασία πρόβλεψης, είναι απαραίτητος ο διαχωρισμός των χρονοσειρών σε εργοδικές και μη-εργοδικές.

- Εργοδικές (Stationary): Οι εργοδικές χρονοσειρές διαθέτουν παραμέτρους, όπως η μέση τιμή και η διακύμανση, που είναι σταθερές στο χρόνο. Τονίζεται πως ο θόρυβος αποτελεί εργοδική χρονοσειρά. Για παράδειγμα, ο λευκός θόρυβος έχει μηδενική μέση τιμή, σταθερή διακύμανση και μηδενική συνδιακύμανση.
- Μη-εργοδικές (Non-stationary): Οι μη-εργοδικές χρονοσειρές διαθέτουν παραμέτρους, όπως η μέση τιμή και η διακύμανση μεταβαλλόμενες στο χρόνο. Αποτελούν τον συνηθέστερο τύπο δεδομένων χρονοσειρών και η πρόβλεψη γι' αυτές είναι δύσκολη. Προτιμάται, για την πρόβλεψη των μη-εργοδικών χρονοσειρών, η μετατροπή τους σε εργοδικές και αφού τελειώσει η διαδικασία πρόβλεψης, η επαναφορά τους σε μη-εργοδικές. Η μετατροπή αυτή γίνεται κυρίως με την εφαρμογή λογαριθμικού μετασχηματισμού, διαφορών, και μεθόδων εξάλειψης εποχικότητας. Οι μετασχηματισμοί αυτοί περιγράφονται αναλυτικότερα στην επόμενη υπο-ενότητα.

Μετατροπή των χρονοσειρών σε εργοδικές

Σύμφωνα με το άρθρο [Jebb15], για την ανάλυση των χρονοσειρών απαιτούνται μετατροπές που διαμορφώνουν διάφορα χαρακτηριστικά των χρονοσειρών, έτσι ώστε αποκτήσουν αυτές μια εργοδική μορφή. Οι αλλαγές που μπορούν να γίνουν περιγράφονται στη συνέχεια.

- Χειρισμός της διακύμανσης (variance): Για τη σταθεροποίηση της διακύμανσης, χρησιμοποιείται ο λογαριθμικός μετασχηματισμός, κατά τον οποίο, όσο πιο μεγάλες είναι οι τιμές, τόσο περισσότερο μειώνονται.
- Χειρισμός των τάσεων (trends): Για τη σταθεροποίηση της μέσης τιμής στις χρονοσειρές ακολουθείται η μέθοδος διαφορών (differencing), η οποία διαφοροποιείται ανάλογα με τον αριθμό των προηγούμενων τιμών που χρησιμοποιεί σε τάξεις διαφορών.
    - Διαφορές πρώτης τάξης: Με τις διαφορές πρώτης τάξης κάθε τιμή μιας χρονοσειράς υπολογίζεται αφαιρώντας από αυτή την αμέσως προηγούμενη τιμή $y_i' = y_i - y_{i-1}$. Το μοντέλο που προκύπτει είναι κατάλληλο για χρονοσειρές που έχουν την μορφή απλής κεκλιμένης ευθείας και περιγράφεται ως:

      $$y_{i+1} = y_i + e_{i+1}\,,$$

      όπου e_{i+1} συμβολίζει το λευκό θόρυβο.
    - Διαφορές δεύτερης τάξης: Με τις διαφορές δεύτερης τάξης κάθε τιμή υπολογίζεται αφαιρώντας τις δύο προηγούμενες από αυτή τιμές $y_i'' = y_i' - y_{i-1}' = y_i - 2 \cdot y_{i-1} + y_{i-2}$. Το μοντέλο που προκύπτει είναι κατάλληλο για χρονοσειρές που παρουσιάζουν τάση, η οποία μεταβάλλεται με το χρόνο και περιγράφεται ως:

      $$y_{i+1} = y_i + c + e_{i+1}\,,$$

      όπου e_{i+1} συμβολίζει το λευκό θόρυβο.
    - Διαφορές εποχικότητας: Σε κάποιες περιπτώσεις χρησιμεύει να υπολογίζεται το μοντέλο με βάση τις διαφορές εποχικότητας

      $$y_i' = y_i - y_{i-m}.$$

  Συνήθως, δεν επιλέγονται διαφορές μεγαλύτερης τάξης από τη δεύτερη. Υπάρχουν όμως περιπτώσεις πολυωνυμικών τάσεων σε χρονοσειρές, που απαιτούν διαφορές μεγαλύτερης τάξης. Εάν εφαρμοστούν διαφορές μεγάλου μεγέθους σε μη απαιτητικό μοντέλο, εμφανίζεται το πρόβλημα υπερ-διαφορών. Γίνεται αντιληπτό ότι ένα μοντέλο διαθέτει το πρόβλημα αυτό, όταν, μετά τη διαδικασία των διαφορών, η διακύμανσή του αυξάνεται, ενώ στην περίπτωση που η διαδικασία λειτουργεί σωστά μειώνεται η διακύμανση.
- Χειρισμός της εποχικότητας (seasonality): Η εποχικότητα είναι ένα χαρακτηριστικό των χρονοσειρών που είτε ενσωματώνεται στο μοντέλο και

ύστερα αφαιρείται, είτε αφαιρείται χωρίς να χρησιμοποιηθεί. Η αφαίρεσή της απαιτεί μια σύνθετη διαδικασία, και αφού ολοκληρωθεί καθιστά δυνατό τον εντοπισμό άλλων χαρακτηριστικών της χρονοσειράς.

Έλεγχος εργοδικότητας

Ο έλεγχος αν μια χρονοσειρά είναι εργοδική μπορεί να γίνει με κάποια γνωστά τεστ, και ένα από αυτά είναι το Augmented Dickey-Fuller (ADF), το οποίο ξεκινά από την υπόθεση ότι μια χρονοσειρά είναι μη-εργοδική, και η απόρριψή του καθιστά ένα μοντέλο έτοιμο για παραγωγή προβλέψεων για ένα μοντέλο ARMA.

2.1.1.2.1 Μοντέλα πρόβλεψης

Η διαδικασία πρόβλεψης περιγράφεται συνήθως από κάποιο μοντέλο. Ακολουθούν κάποια από τα μοντέλα που χρησιμοποιούνται.

Πρόβλεψη μονοδιάστατων χρονοσειρών

- Autoregressive Models (AR): Τα AR μοντέλα για πρόβλεψη σε χρονοσειρές χρησιμοποιούν τις αυτοσυσχετίσεις για να επιτύχουν την πρόβλεψη των επόμενων τιμών. Οι αυτοσυσχετίσεις είναι συσχετίσεις μεταξύ γειτονικών χρονοσφραγίδων. Τα AR μοντέλα διαθέτουν ένα παράθυρο μεγέθους p από προηγούμενες τιμές, το οποίο καλείται τάξη (order) του μοντέλου. Ο ορισμός κάθε τιμής ενός μοντέλου $AR(p)$, δηλαδή τάξης p, σύμφωνα με το βιβλίο [HA2018] είναι ο εξής.

$$y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \varphi_p y_{t-p} + \varepsilon_t ,$$

  όπου ε_t συμβολίζει το λευκό θόρυβο. Οι παράμετροι $\varphi_1, \dots, \varphi_p$ προσδίδουν ευελιξία στο μοντέλο, και προκύπτουν από συναρτήσεις βελτιστοποίησης και όπου c αυθαίρετη σταθερά.

- Moving Average (MA): Σύμφωνα με το βιβλίο [HA2018] τα MA μοντέλα χρησιμοποιούν για την πρόβλεψη τιμών σε χρονοσειρές, τα σφάλματα πρόβλεψης των προηγούμενων τιμών. Τα μοντέλα MA διαθέτουν επίσης ένα παράθυρο από q προηγούμενες τιμές, το οποίο καλείται τάξη (order) q του μοντέλου. Ο ορισμός κάθε τιμής ενός μοντέλου $MA(q)$, δηλαδή τάξης q είναι ο εξής.

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} ,$$

όπου ε_t συμβολίζει το λευκό θόρυβο. Οι παράμετροι $\theta_1, \dots, \theta_q$ προσδίδουν ευελιξία στο μοντέλο, και προκύπτουν από συναρτήσεις βελτιστοποίησης και όπου *c* αυθαίρετη σταθερά.

- Autoregressive Moving Average Models (ARMA): Τα ARMA μοντέλα για πρόβλεψη σε χρονοσειρές λειτουργούν όπως και τα AR μοντέλα και μπορούν επίσης να υποστηρίξουν και shocks. Τα ARMA μοντέλα συνδυάζουν τα AR με τα Moving Average Models (MA), που είναι μοντέλα που αξιοποιούν μόνο τα shocks και όχι τα δεδομένα της χρονοσειράς. Ο συνδυασμός των δύο μοντέλων έχει p παραμέτρους για AR terms, και q για MA terms, και καλείται ARMA(p, q). Ιδιαίτερη προσοχή πρέπει να δοθεί στις παραμέτρους αυτές. Για μικρές τιμές των παραμέτρων, το μοντέλο ενδέχεται να μην ταιριάζει καλά στα δεδομένα, ενώ για μεγάλες τιμές ενδέχεται να υπερ-ταιριάξει σε αυτά. Η καλύτερη προσέγγιση θέτει τις τιμές όσο το δυνατόν μικρότερες, και να ταιριάζουν με τα δεδομένα.
- Autoregressive Integrated Moving Average Models (ARIMA): Τα ARMA μοντέλα χρησιμοποιούνται σε εργοδικές χρονοσειρές, ενώ για τις μη-εργοδικές χρησιμοποιούνται τα ARIMA μοντέλα, τα οποία διαθέτουν επιπλέον εφαρμογή διαφορών. Τα μοντέλα ARIMA, αποτελούν το αντικείμενο αυτής της διπλωματικής και περιγράφονται αναλυτικά στο Κεφάλαιο 3.

Πρόβλεψη πολυδιάστατων χρονοσειρών

Μια εφαρμογή μπορεί να αποτελείται από περισσότερες από μία χρονοσειρές, οι οποίες εκτός από αυτοσυσχετίσεις να έχουν και «σταυρωτές» συσχετίσεις (cross-correlations) μεταξύ τους. Για να παραχθεί μια πρόβλεψη γι' αυτές τις χρονοσειρές πρέπει με κάποιον τρόπο να συνδυαστούν και οι δύο συσχετίσεις τους. Στις χρονοσειρές αυτές χρησιμοποιείται η μέθοδος των κρυφών τιμών (hidden values). Κατά τη μέθοδο αυτή μπορεί να μετατραπεί ένας μεγάλος αριθμός χρονοσειρών με «σταυρωτές» συσχετίσεις, σε ένα μικρότερο αριθμό χρονοσειρών που δεν έχουν συσχετίσεις μεταξύ τους. Η μετατροπή γίνεται με τον αλγόριθμο Principal Component Analysis (PCA). Στις μετασχηματισμένες από τον αλγόριθμο χρονοσειρές, εφαρμόζεται κάποιο από τα μοντέλα AR, ARMA ή ARIMA για πρόβλεψη των «κρυφών» τιμών. Τέλος, οι τιμές που προκύπτουν από την πρόβλεψη, επιστρέφονται στην αρχική τους αναπαράσταση.

2.1.1.3 Εύρεση μοτίβων χρονοσειρών (motif detection)

Τα μοτίβα που εντοπίζονται στις χρονοσειρές είναι κάποια συχνά πρότυπα ή σχήματα. Υπάρχουν αρκετοί τρόποι διατύπωσης για το πρόβλημα εντοπισμού μοτίβων.

- Διατίθεται μια μόνο χρονοσειρά και όλες οι εμφανίσεις ενός μοτίβου βρίσκονται σε αυτήν ή διατίθεται μια βάση δεδομένων με πολλές χρονοσειρές και γίνεται αναζήτηση για τουλάχιστον μία εμφάνιση ενός μοτίβου σε κάθε χρονοσειρά.
- Συνεχόμενα μοτίβα, που όλα τα στοιχεία τους είναι συνεχόμενα ή μη-συνεχόμενα μοτίβα, όπου επιτρέπονται κενά ανάμεσα στα στοιχεία τους. Συνήθως, η έρευνα επικεντρώνεται σε συνεχόμενα μοτίβα.
- Multigranularity μοτίβα: Συνήθως, αναζητούνται μοτίβα σε χρονοσειρές σε παράθυρο χρόνου μιας κλίμακας μεγεθών. Όμως, είναι πιθανό κάποια μοτίβα να εμφανίζονται σε διαφορετικές κλίμακες μεγεθών, και έτσι να απαιτούν διάφορα παράθυρα για κάθε μία. Για παράδειγμα, κάποιο μοτίβο μπορεί να εμφανίζεται σε διαστήματα ημερών, ενώ ένα άλλο να εμφανίζεται σε διαστήματα μηνών.

Τα μοτίβα εντοπίζονται σε χρονοσειρές απευθείας με μεθόδους ή γίνεται μετατροπή των χρονοσειρών σε διακριτές ακολουθίες και τότε εφαρμογή μεθόδων διακριτών ακολουθιών σε αυτές. Συγκεκριμένα:

- Distance-based support: Για τον εντοπισμό μοτίβων σε μια χρονοσειρά μπορεί να γίνει διαχωρισμός της χρονοσειράς σε τμήματα και να υπολογισθεί η ομοιότητα ενός μοτίβου αυτά τα τμήματα. Η εύρεση της ομοιότητας μπορεί να επιτευχθεί με τη χρήση Euclidean distance ή DTW. Πρώτα, εντοπίζονται τα διάφορα μοτίβα και στη συνέχεια εξάγονται τα καλύτερα-k με βάση την ομοιότητα. Ο υπολογισμός της Euclidean distance αποτελείται από ένα Nested Loop και έχει πολυπλοκότητα χρόνου $O(n^2)$, ενώ ο υπολογισμός της DTW είναι πιο ακριβός. Ως βελτιστοποίηση γίνεται πρώτα PAA ή SAX στις χρονοσειρές, και στη συνέχεια υπολογίζονται οι αποστάσεις.
- Transformation to sequential mining: Ο εντοπισμός μοτίβων σε χρονοσειρές μπορεί να γίνει με τη μετατροπή τους σε διακριτές ακολουθίες και ύστερα, χρήσης τεχνικών εντοπισμού μοτίβων για δεδομένα ακολουθιών. Η μετατροπή των δεδομένων γίνεται με μεθόδους όπως PAA ή SAX.

Στις χρονοσειρές υπάρχουν επίσης κάποια πρότυπα που εμφανίζουν περιοδικότητα.

Περιοδικά πρότυπα

Τα περιοδικά patterns των χρονοσειρών διαχειρίζονται με τον Discrete Fourier Transformation (DFT), ο οποίος διαχωρίζει τις χρονοσειρές σε n-1 περιοδικές ημιτονοειδείς συνιστώσες. Κάποια από αυτές τις συνιστώσες έχει μεγαλύτερο πλάτος και υπερισχύει των υπολοίπων. Με χρήση του DFT, εκτός από την εύρεση περιοδικών προτύπων, επιτυγχάνεται και μείωση διαστάσεων.

2.1.1.4 Συσταδοποίηση χρονοσειρών (clustering)

Η διαδικασία συσταδοποίησης είναι εκείνη, στην οποία δημιουργούνται συστάδες (clusters) με δεδομένα που παρουσιάζουν ομοιότητα μεταξύ τους. Η διαδικασία μπορεί να εφαρμοστεί δυναμικά κατά την άφιξη των δεδομένων (online) ή στατικά στα δεδομένα μιας βάσης δεδομένων (shape-based).

- Online clustering: Η online συσταδοποίηση δεδομένων χρονοσειρών γίνεται ελέγχοντας ένα παράθυρο πρόσφατου ιστορικού. Οι χρονοσειρές χωρίζονται σε ομάδες, βασισμένες στις συσχετίσεις (correlations) μεταξύ των χρονοσειρών αυτού του παραθύρου. Αφού η διαδικασία γίνεται online, οι διάφορες χρονοσειρές μπορεί να χρειασθεί να μετακινηθούν από συστάδα σε συστάδα. Για παράδειγμα, μια χρονοσειρά την ημέρα x μπορεί να βρίσκεται στη συστάδα c_1, ενώ την ημέρα $(x + 1)$ να μεταφερθεί στη συστάδα c_2. Σε πραγματικό χρόνο, για τις διάφορες χρονοσειρές, χρησιμοποιούνται ειδικές regression-based συναρτήσεις ομοιότητας. Η ομοιότητα των χρονοσειρών σχετίζεται με την προβλεψιμότητα (predictability), δηλαδή, την πρόβλεψη της μιας χρονοσειράς από την άλλη, αν συγκρίνονται δύο χρονοσειρές.
- Shape-based clustering: Κατά τη διαδικασία shape-based clustering σε μια βάση δεδομένων από χρονοσειρές, συγκρίνονται δεδομένα που δε συλλέγονται απαραίτητα σε συνεχόμενες χρονικές στιγμές, αλλά με βάση κάποια σχήματα (shapes). Επίσης, οι χρονοσειρές που εξετάζονται μπορεί να μην είναι συγχρονισμένες στο χρόνο, με διαφορετική κλιμάκωση, και διαφορετικό μήκος.

  Το πρώτο βήμα της διαδικασίας shape-based clustering είναι η επιλογή μιας shape-based μεθόδου ομοιότητας. Οι χρονοσειρές που βρίσκονται στη βάση δεδομένων που διατίθεται, κλιμακώνονται, μετασχηματίζονται ή επεκτείνονται, η κάθε μία διαφορετικά. Ακολουθεί μια απλή περιγραφή κάποιων βασικών shape-based μεθόδων ομοιότητας.

- k-means: Η μέθοδος k-means χρησιμοποιεί την Ευκλείδεια απόσταση για τον υπολογισμό της ομοιότητας και απαιτεί οι χρονοσειρές της βάσης δεδομένων που χρησιμοποιείται, να έχουν ίδιο μήκος και αντιστοιχία 1-1.
- k-medoids: Η μέθοδος k-medoids χρησιμοποιεί διάφορες συναρτήσεις για τον υπολογισμό της ομοιότητας και δεν απαιτεί οι χρονοσειρές που εξετάζονται να έχουν ίδιο μήκος.
- Hierarchical methods: Οι ιεραρχικές μέθοδοι χρησιμοποιούν τεχνικές του δυναμικού προγραμματισμού, οι οποίες λειτουργούν αποδοτικά για μικρές χρονοσειρές, ενώ για μεγαλύτερες έχουν μεγάλο κόστος.
- Graph-based methods: Οι graph-based μέθοδοι χρησιμοποιούν μια συνάρτηση ομοιότητας που παράγει ένα γράφημα ομοιότητας για τα δεδομένα. Κάθε κόμβος του γραφήματος είναι μια μέτρηση της χρονοσειράς, και συνδέεται με τους k-κοντινότερους γείτονές του. Το γράφημα ομοιότητας είναι μη-κατευθυνόμενο και στις ακμές του αναγράφεται η ομοιότητα ανάμεσα σε κάθε ζεύγος κόμβων. Μετά τον υπολογισμό της ομοιότητας των δεδομένων, τα συσταδοποιημένα δεδομένα μετατρέπονται ξανά σε χρονοσειρές.

2.1.1.5 Εντοπισμός ακραίων τιμών χρονοσειρών (outlier detection)

Μια ακραία τιμή δεδομένων χρονοσειρών, είναι εκείνη που διαφέρει από την αναμενόμενη τιμή ή εκείνη που προέκυψε από τη διαδικασία πρόβλεψης, για την ίδια χρονική στιγμή. Η διαδικασία στοχεύει στον εντοπισμό αυτών των τιμών, που διακρίνονται σε δύο βασικές κατηγορίες.

- Point outliers: Οι point outliers, είναι μη αναμενόμενες αλλαγές μιας τιμής της χρονοσειράς.
- Shape outliers: Οι shape outliers αφορούν μη-ομαλότητες (anomalies) σε ολόκληρα παράθυρα των χρονοσειρών. Τα παράθυρα αυτά εντοπίζονται με την προσέγγιση Hotsax, και στη συνέχεια βρίσκεται σε αυτά κάποιος outlier-βαθμός με τη μέθοδο k-means. Οι χρονοσειρές διαχωρίζονται σε παράθυρα, και το κάθε ένα από αυτά συγκρίνεται με τα μη-επικαλυπτόμενα σε αυτό παράθυρα. Τα παράθυρα με την υψηλότερη k-nearest neighbor απόσταση θεωρούνται ακραίες τιμές.

2.1.1.6 Κατηγοριοποίηση χρονοσειρών (classification)

Η διαδικασία κατηγοριοποίησης, είναι αυτή κατά την οποία διαχωρίζονται οι χρονοσειρές σε κατηγορίες, που ορίζονται από κάποιες ετικέτες (labels). Οι χρονοσειρές είναι δυνατό να ανήκουν σε κάποια κατηγορία λόγω της ύπαρξης κάποιας συγκεκριμένης τιμής (point labels), ή κάποιου συγκεκριμένου σχήματος σε αυτές (whole-series labels).

- Point labels: Η κατηγοριοποίηση γίνεται σε χρονοσφραγίδες των χρονοσειρών, όπου αντιμετωπίζεται το πρόβλημα ανίχνευσης γεγονότων (event detection), κατά το οποίο εντοπίζονται σπάνιες και ασυνήθιστες δραστηριότητες της χρονοσειράς. Αρχικά, για κάθε χρονοσειρά χρησιμοποιείται ένα μοντέλο πρόβλεψης από αυτά που αναφέρθηκαν σε προηγούμενο κεφάλαιο και από το μοντέλο εξάγονται κάποιες τιμές σφάλματος. Από τις τιμές αυτές, offline, για d χρονοσειρές, δημιουργούνται κάποιες παράμετροι που καλούνται alarm level $\alpha_1 \dots \alpha_d$. Με βάση το alarm level, σε πραγματικό χρόνο (online), μπορεί να κριθεί εάν μια χρονοσειρά διαθέτει κάποιο χαρακτηριστικό κατά το οποίο, να ανήκει σε συγκεκριμένη κατηγορία.
- Whole-series labels: Η κατηγοριοποίηση των χρονοσειρών μπορεί να γίνει με βάση κάποιο σχήμα σε αυτές. Τότε, δίνονται ετικέτες για ολόκληρες τις χρονοσειρές και εφαρμόζεται κάποια από τις παρακάτω προσεγγίσεις. Ακολουθείται μια διαδικασία εκπαίδευσης ενός δικτύου τύπου κατηγοριοποιητή (classifier), όπου απαιτείται ο διαχωρισμός των δεδομένων σε τμήματα (instances) δύο ειδών, τα εκπαίδευσης και τα επιβεβαίωσης.
    - Wavelet-based rules: Κατά την wavelet-based προσέγγιση, μετατρέπονται οι χρονοσειρές σε κυματομορφές με χρήση της Haar wavelet μεθόδου, η οποία κάνει αποσύνθεση των χρονοσειρών. Έτσι, μπορούν να χρησιμοποιηθούν σχήματα διαφόρων βαθμών λεπτομέρειας (granularity).
    - Nearest neighbor classifier: Κατά την nearest neighbor προσέγγιση, γίνεται χρήση μιας συνάρτησης υπολογισμού ομοιότητας. Για κάθε τμήμα επιβεβαίωσης μιας χρονοσειράς ελέγχεται η ομοιότητα του με τους k-κοντινότερους γείτονες.
    - Graph-based methods: Κατά την graph-based προσέγγιση, δημιουργείται ένα γράφημα που περιλαμβάνει τα τμήματα εκπαίδευσης και επιβεβαίωσης μαζί. Στο γράφημα αυτό δίνονται ετικέτες μόνο στις ακμές των τμημάτων επιβεβαίωσης και σταδιακά ανακαλύπτονται τα οι ετικέτες των ακμών των τμημάτων επιβεβαίωσης.

2.1.2 Ακολουθίες (Sequences)

Κατ' αναλογία με τις χρονοσειρές, ο χρόνος θεωρείται ισομορφικός του χώρου των φυσικών αριθμών ℕ. Τα χρονικά δεδομένα σε μορφή ακολουθίας, αφορούν διακριτές τιμές του χρόνου, και έτσι υποθέτουν πως δεν υπάρχει καμία αλλαγή ανάμεσα σε δύο διαφορετικές τιμές του. Οι διακριτές αυτές ακολουθίες μπορούν να περιγράψουν καταστάσεις ενός συστήματος, βιολογικά δεδομένα ή ενέργειες χρηστών, κάποιες από τις οποίες είναι οι κατάλογοι διαδικτύου (web logs) και οι συναλλαγές (transactions).

2.1.2.1 Εξόρυξη προτύπων ακολουθιών (Sequential Pattern Mining)

Η εξόρυξη των προτύπων σε διακριτές ακολουθίες, είναι η διαδικασία κατά την οποία αναζητούνται συγκεκριμένα πρότυπα σε αυτές. Η διαδικασία αυτή, είναι ανάλογη με τη διαδικασία Frequent Pattern Mining για τα πολυδιάστατα δεδομένα, εστιάζοντας στα χρονικά δεδομένα.

Έστω ότι διατίθενται N ακολουθίες, και η i-οστή περιλαμβάνει n_i στοιχεία (elements) σε χρονική σειρά.

Υπακολουθία (Subsequence): Μια ακολουθία Y λέγεται πως έχει ως υπακολουθία την ακολουθία Z, όταν η Z περιέχει ολόκληρα κάποια στοιχεία της Y ή υποσύνολα αυτών σε ίδια χρονική σειρά με την Y.

Στήριγμα (Support): Έστω μια βάση από ακολουθίες $T = \{Y_1 \dots Y_n\}$, και έστω μια ακολουθία Z. Το υποσύνολο της βάσης T που περιέχει τις ακολουθίες Y_i που έχουν ως υπακολουθία τη Z ονομάζεται στήριγμα για την ακολουθία Z.

Για τη διαδικασία εξόρυξης προτύπων ακολουθιών ένας αλγόριθμος που χρησιμοποιείται είναι ο Generalized Sequence Pattern Mining Algorithm (GSP), ο οποίος μοιάζει με τον Apriori αλγόριθμο. Ο αλγόριθμος GSP βασίζεται στο γεγονός ότι τα αντικείμενα που σχετίζονται θα εμφανίζονται συχνά μαζί.

Το μήκος μιας ακολουθίας, είναι ο αριθμός των αντικειμένων που αυτή περιέχει. Μια ακολουθία με μήκος k καλείται k-υποψήφια (k-candidate).

Ο αλγόριθμος GSP, εφαρμόζει κάποια διαδικασία συνένωσης σε ακολουθίες. Για τη συνένωση, έστω ότι διατίθενται δύο ακολουθίες S_1 και S_2. Η συνένωση των ακολουθιών είναι δυνατή εάν με διαγραφή ενός αντικειμένου από το πρώτο στοιχείο της S_1 προκύπτει

ίδιο αποτέλεσμα με αυτό της διαγραφής ενός αντικειμένου από το τελευταίο στοιχείο της S_2. Για παράδειγμα, στις παρακάτω ακολουθίες ο περιορισμός ισχύει.

$$S_1 = \langle\{Bread, Butter, Cheese\}, \{Cheese, Eggs\}\rangle$$

$$S_2 = \langle\{Bread, Butter\}, \{Milk, Cheese, Eggs\}\rangle$$

Εάν ικανοποιείται αυτός ο περιορισμός, ακολουθείται μία απ' τις δύο παρακάτω περιπτώσεις.

- Αν το τελευταίο στοιχείο της S_2 είναι 1-itemset, προστίθεται (append) ως στοιχείο στο τέλος της S_1. Για παράδειγμα:

$$S_1 = \langle\{Bread, Butter, Cheese\}, \{Cheese, Eggs\}\rangle$$

$$S_2 = \langle\{Bread, Butter\}, \{Cheese, Eggs\}, \{Milk\}\rangle$$

$$join(S_1, S_2) = \langle\{Bread, Butter, Cheese\}, \{Cheese, Eggs\}, \{Milk\}\rangle$$

- Αλλιώς, αν το τελευταίο στοιχείο της S_2 είναι υπερσύνολο του τελευταίου στοιχείου της S_1, γίνεται αντικατάσταση του τελευταίου στοιχείου της S_1 με το τελευταίο στοιχείο της S_2. Για παράδειγμα:

$$S_1 = \langle\{Bread, Butter, Cheese\}, \{Cheese, Eggs\}\rangle$$

$$S_2 = \langle\{Bread, Butter\}, \{Milk, Cheese, Eggs\}\rangle$$

$$join(S_1, S_2) = \langle\{Bread, Butter, Cheese\}, \{Milk, Cheese, Eggs\}\rangle$$

Η περιγραφή του αλγορίθμου GSP μπορεί να γίνει με ένα δέντρο υποψηφίων, που είναι ανάλογο με τα enumeration trees για τον αλγόριθμο Apriori.

Δέντρο Υποψηφίων (Candidate Tree)

Τα δέντρα υποψηφίων είναι δομές που περιγράφουν τα δεδομένα του GSP αλγορίθμου και οι κόμβοι τους παράγονται με τις ακόλουθες πράξεις. Έστω ότι διατίθεται η ακολουθία S.

- Set-wise extension: Η πράξη set-wise extension εφαρμόζεται με τοποθέτηση ενός αντικειμένου στο τελευταίο στοιχείο της S. Με την τοποθέτηση αυτή πρέπει να δημιουργείται μια εκτεταμένη ακολουθία. Επίσης, πρέπει το νέο αντικείμενο που τοποθετείται να είναι λεξικογραφικά μεγαλύτερο από τα προηγούμενα στοιχεία στο συγκεκριμένο στοιχείο.
- Temporal extension: Η πράξη temporal extension εφαρμόζεται με τοποθέτηση ενός στοιχείου με ένα μόνο αντικείμενο στο τέλος της S. Με την τοποθέτηση αυτή πρέπει να δημιουργείται μια εκτεταμένη ακολουθία.

Ακόμη, κάποιες φορές στην εξόρυξη των προτύπων σε ακολουθίες χρησιμοποιούνται κάποιοι περιορισμοί.

Εξόρυξη προτύπων ακολουθιών με περιορισμούς

Η χρήση περιορισμών στη διαδικασία εύρεσης προτύπων σε ακολουθίες είναι αποδοτικότερη όταν οι περιορισμοί εφαρμόζονται κατά τη διάρκεια της GSP μεθόδου. Η χρήση GSP χωρίς περιορισμούς, και στη συνέχεια διαγραφή των υπακολουθιών που δεν ικανοποιούν τους περιορισμούς είναι ασύμφορη. Σε μια τέτοια περίπτωση, τα αποτελέσματα που μας ενδιαφέρουν θα ήταν πολύ λιγότερα από τα αποτελέσματα που προκύπτουν συνολικά, και έτσι η ανάκτηση των αποτελεσμάτων θα είχε μεγάλο κόστος. Ένας περιορισμός που χρησιμοποιείται είναι ο maxspan, κατά τον οποίο η χρονική διαφορά ανάμεσα στο πρώτο και το τελευταίο στοιχείο μιας ακολουθίας δεν επιτρέπεται να έχουν μεγαλύτερο μήκος από μια παράμετρο maxspan. Δύο άλλοι περιορισμοί που χρησιμοποιούνται είναι οι maxgap και mingap.

2.1.2.2 Συσταδοποίηση ακολουθιών (clustering)

Η διαδικασία συσταδοποίησης είναι εκείνη, στην οποία δημιουργούνται συστάδες (clusters) με δεδομένα που παρουσιάζουν ομοιότητα μεταξύ τους.

Κατά τη συσταδοποίηση για τις διακριτές ακολουθίες γίνεται χρήση μιας συνάρτησης ομοιότητας, η οποία πρέπει να επιλεχθεί με προσοχή, γιατί επηρεάζει σημαντικά το αποτέλεσμα. Αναφέρονται κάποιες από τις συναρτήσεις ομοιότητας.

- Match-based measure: Η συνάρτηση αυτή απαιτεί οι ακολουθίες που συγκρίνονται να έχουν ίδιο μήκος, και 1-1 αντιστοιχία μεταξύ τους.
- Dynamic Time Warping: Η συνάρτηση αυτή δεν απαιτεί οι ακολουθίες να έχουν ίδιο μήκος, αφού κάνει έκταση και συρρίκνωση για να τις φέρει σε κατάλληλη μορφή.
- Longest common subsequence: Κατά τη συνάρτηση αυτή οι ακολουθίες συγκρίνονται με βάση τη μεγαλύτερη κοινή υπακολουθία μεταξύ τους.
- Edit distance: Η συνάρτηση αυτή μελετά την ομοιότητα μεταξύ δύο ακολουθιών υπολογίζοντας πόσοι μετασχηματισμοί χρειάζονται έτσι ώστε να μετατραπεί η πρώτη στη δεύτερη.
- Keyword-based similarity: Στη συνάρτηση αυτή εξάγω από τις ακολουθίες κάποια k-grams με τη χρήση κάποιου κυλιόμενου παραθύρου μήκους k. Κάθε k-gram, αποτελεί αναπαράσταση μιας ακολουθίας ως σακίδιο (bag) από τμήματα (segments) μεγέθους k. Τα k-grams μπορούν να αξιοποιηθούν ως λέξεις

(keywords) και να υπολογισθεί η απόστασή τους με τη μέθοδο tf-idf. Επίσης, στην αναπαράσταση με k-grams χάνεται η έννοια της σειράς των δεδομένων, και έτσι επιτρέπεται η χρήση οποιουδήποτε αλγορίθμου εξόρυξης κειμένου (text mining).

- Kernel-based similarity: Η συνάρτηση αυτή, χρησιμοποιείται επίσης για κατηγοριοποίηση ακολουθιών και περιγράφεται στο κεφάλαιο της κατηγοριοποίησης ακολουθιών.

Για συσταδοποίηση δεδομένων ακολουθιών μπορεί να εφαρμοστεί κάποια από τις ακόλουθες μεθόδους.

Μέθοδοι συσταδοποίησης ακολουθιών

- Distance-based methods: Για τις distance-based μεθόδους συσταδοποίησης σε ακολουθίες αρχικά πρέπει να επιλεγεί μια συνάρτηση της ομοιότητας τους. Ακολουθούν τρείς διαφορετικές προσεγγίσεις μιας τέτοιας συνάρτησης. Η πιο απλή προσέγγιση, θα ήταν μια γενικευμένη μορφή του αλγορίθμου k-medoids, ο οποίος όμως δεν θα είχε γνώση του τύπου δεδομένων και της συνάρτησης ομοιότητας που έχει επιλεγεί. Ως δεύτερη προσέγγιση, ο αλγόριθμος CLARANS σε μια γενική του μορφή μπορεί να λειτουργήσει με δεδομένα μορφής ακολουθιών χρησιμοποιώντας τη συνάρτηση ομοιότητας που έχει επιλεγεί. Μια ακόμη προσέγγιση, θα ήταν η χρήση γενικευμένης μορφής των hierarchical methods, οι οποίες χρειάζονται όμως χρόνο πολυπλοκότητας $O(n_2)$ για τον έλεγχο της ομοιότητας σε κάθε ζεύγος αντικειμένων.
- Graph-based methods: Κατά τις graph-based μεθόδους γίνεται αντιστοίχιση των δεδομένων σε ένα γράφημα ομοιότητας. Στο γράφημα οι κόμβοι αναπαριστούν τα δεδομένα της ακολουθίας και οι ακμές συνδέουν τους k-κοντινότερους γειτονικούς κόμβους. Το γράφημα είναι μη-κατευθυνόμενο. Μετά τη συσταδοποίηση, τα δεδομένα επαναφέρονται στην αρχική τους αναπαράσταση.
- Subsequence-based methods: Η ευθυγράμμιση ακολουθιών είναι η τοποθέτηση τους σε κοινά όρια, έτσι ώστε να είναι δυνατή σύγκρισή τους. Στις προηγούμενες δύο μεθόδους χρησιμοποιείται η καθολική ευθυγράμμιση (global alignment), η οποία γίνεται σε όλο το μήκος της ακολουθίας και είναι ασύμφορη για μεγάλες ακολουθίες της μεγάλης πιθανότητας θορύβου που τις χαρακτηρίζει. Οι subsequence-based μέθοδοι για συσταδοποίηση ακολουθιών χρησιμοποιούν τοπική ευθυγράμμιση (local alignment), και έτσι, ξεχωρίζουν τα χρήσιμα μέρη τους από τα θορυβώδη. Υπάρχουν πολλές εκδοχές αυτών των μεθόδων. Κάποιες εκδοχές τους χρησιμοποιούν μετατροπή των ακολουθιών σε k-grams, που όμως

είναι επίσης ευαίσθητα στο θόρυβο και δεν προτιμώνται. Κάποιες άλλες εκδοχές κρατούν από κάθε ακολουθία τις συχνές υπακολουθίες της και μετατρέπουν τα δεδομένα σε ειδική αναπαράσταση υπακολουθιών. Για την αναπαράσταση αυτή μπορεί να δημιουργηθεί η κατάλληλη bag-of-words αναπαράσταση και ύστερα, να γίνει η συσταδοποίηση με αλγορίθμους εξόρυξης δεδομένων κειμένων.

- Probabilistic clustering: Κατά τις πιθανοτικές μεθόδους συσταδοποίησης, κάθε σύμβολο μιας ακολουθίας έχει εμφανιστεί με κάποια πιθανότητα, η οποία χρησιμοποιεί στατιστικές συσχετίσεις με τα προηγούμενα σύμβολα. Αυτή είναι και η βασική αρχή των Markovian Models, που περιγράφονται στη συνέχεια. Η ομοιότητα ανάμεσα σε μια ακολουθία και μια συστάδα υπολογίζεται από την πιθανότητα εμφάνισης των συμβόλων στη συστάδα. Αφού βρεθεί η ομοιότητα, γίνεται χρήση ενός distance-based αλγορίθμου, όπως ο CLUSEQ ή ακολουθείται η προσέγγιση των Mixture of Hidden Markov Models. Ακολουθεί περιγραφή των δύο αυτών προσεγγίσεων.
    - Ο αλγόριθμος CLUSEQ: Ο αλγόριθμος CLUSEQ, χαρακτηρίζεται ως similarity-based iterative partitioning algorithm. Ο αλγόριθμος χρησιμοποιεί την ομοιότητα που υπολογίζεται από κάποιο Markovian Model. Ένα θετικό στοιχείο του αλγορίθμου CLUSEQ είναι ότι στη χρήση του καθορίζεται αυτόματα ο αριθμός των συστάδων. Στην αρχικοποίηση του αλγορίθμου υπάρχει μια μόνο συστάδα, και στη συνέχεια, σε κάθε του επανάληψη δημιουργούνται περισσότερες συστάδες για τις ξεχωριστές ακολουθίες. Οι συστάδες που είναι πολύ όμοιες με τις ως τώρα ευρεθείσες, διαγράφονται. Ο αλγόριθμος διαθέτει ένα κατώφλι t, με το οποίο καθορίζει την ελάχιστη τιμή ομοιότητας που μπορεί να έχει μια ακολουθία για να ενταχθεί σε μια συστάδα. Μια ακολουθία, μπορεί να βρίσκεται σε καμία, μία ή περισσότερες συστάδες. Ο αλγόριθμος λειτουργεί μέχρι να μην υπάρχει καμία αλλαγή στις συστάδες σε δύο διαδοχικές επαναλήψεις.
    - Mixture of Hidden Markov Models: Στη γενική του μορφή, ένα μείγμα (mixture) αριθμητικών δεδομένων όπου κάθε στοιχείο του έχει παραχθεί από την Γκαουσιανή (Gaussian) κατανομή μπορεί προσεγγιστεί με ένα generative mixture model. Αντίστοιχα, για τα δεδομένα ακολουθιών χρησιμοποιούνται τα Hidden Markov Models (HMM). Τα HMM αποτελούν από μόνα τους μια μορφή μείγματος, όμως εδώ θα αναφερθεί η περίπτωση μείγματος δύο σταδίων, που είναι αντίστοιχη της γενικής μορφής.

Γίνεται η υπόθεση ότι τα δεδομένα της ακολουθίας που διατίθεται έχουν προέλθει από μείγμα k κατανομών $G_1 \dots G_k$, όπου κάθε G_i αποτελεί ένα HMM. Για κάθε κατανομή υπάρχει μια πιθανότητα επιλογής a_i, δηλαδή έχουμε $a_i \dots a_k$ πιθανότητες συνολικά. Κατά τη διαδικασία που ακολουθείται επιλέγεται μία από τις k κατανομές με πιθανότητα a_i, για παράδειγμα η r-οστή, και τότε παράγεται από την G_r μια ακολουθία. Στη συνέχεια ακολουθούνται δύο αλγοριθμικά βήματα. Πρώτα εκτελείται το E-step και στη συνέχεια το M-step. Η προσέγγιση όμως αυτή μπορεί να είναι εξαιρετικά αργή, και αυτό συμβαίνει αφού τα HMM απαιτούν μια, ακριβή υπολογιστικά, φάση εκπαίδευσης. Στη συνέχεια, υπάρχει ενότητα που αναλύει σε βάθος τα HMM καθώς και τα αλγοριθμικά βήματα που αναφέρθηκαν.

2.1.2.3 Εντοπισμός ακραίων τιμών ακολουθιών (outlier detection)

Ως ακραία τιμή θεωρείται σε δεδομένα ακολουθιών, θεωρείται εκείνη που διαφέρει από την αναμενόμενη τιμή, για την ίδια χρονική στιγμή. Για τα δεδομένα μορφής διακριτών ακολουθιών, οι ακραίες τιμές χωρίζονται σε δύο κατηγορίες, τους position και combination outliers. Με τη χρήση ενός μοντέλου ακολουθιών εξάγονται κάποιες τιμές από πρόβλεψη, και από την απόκλιση που έχουν οι πραγματικές τιμές από αυτές της πρόβλεψης, χαρακτηρίζονται οι ακραίες τιμές. Για τις position outliers είναι κατάλληλα τα Markovian Models, ενώ για τις combination outliers τα Hidden Markov Models.

- Position outliers: Οι position outliers αποτελούν τιμές μιας ακολουθίας που έχουν μικρή πιθανότητα ταιριάσματος με την προβλεπόμενη ή αναμενόμενη τιμή γι' αυτές. Για την εύρεση αυτών των τιμών διατίθενται ένα σύνολο από ακολουθίες εκπαίδευσης (training) και μια ακολουθία επιβεβαίωσης (test). Επίσης, χρησιμοποιείται ένα μικρό παράθυρο από προηγούμενα σύμβολα, και επιπρόσθετα, καινούρια σε κάποιες ειδικές περιπτώσεις. Το παράθυρο αυτό ονομάζεται πεδίο Short Memory, και χάρη σε αυτό σε μια ακολουθία $V = \alpha_1 \dots \alpha_i \dots$ η πιθανότητα $P(\alpha_i|\alpha_1 \dots \alpha_{i-1})$ μπορεί να προσεγγιστεί από τις πιο πρόσφατες τιμές, μόνο τιμές της ακολουθίας, δηλαδή $P(\alpha_i|\alpha_{i-k} \dots \alpha_{i-1})$ με μικρή παράμετρο k.

  Ο συνδυασμός ενός πεδίου Short Memory και μιας Markov Chain παράγει μια ακολουθία, η οποία παίρνει τιμές από ένα αλφάβητο Σ και καλείται Markov Model (MM).

- First-order Markov Models ονομάζονται τα μοντέλα στα οποία κάθε κατάσταση (state) αντιστοιχίζεται με το τελευταίο σύμβολο μιας ακολουθίας S, που έχει προκύψει από το αλφάβητο Σ.
- k-order Markov Models ονομάζονται τα μοντέλα στα οποία κάθε κατάσταση αντιστοιχίζεται με τα τελευταία k σύμβολα μιας ακολουθίας S, που έχει προκύψει από το αλφάβητο Σ.

Το μέγεθος της μνήμης που χρησιμοποιείται σε ένα MM είναι ίσο με την τάξη (order) k του μοντέλου. Η αύξηση της τάξης πρέπει να γίνεται με προσοχή, καθώς για υψηλές τιμές της υπάρχει η πιθανότητα υπερταιριάσματος (overfitting). Ακόμη, τα MM αναπαρίστανται με ένα γράφημα, που έχει ως κόμβους τις διάφορες καταστάσεις (states) και ακμές τις μεταβάσεις κάθε κατάστασης σε κατάσταση.

Ο υπολογισμοί των Markov Models μπορεί να είναι εξαιρετικά αργοί. Για παράδειγμα σε μοντέλο τάξης k, οι προγενέστερες τιμές μιας τιμής μπορούν να είναι $|\Sigma|^k$. Τότε, ο υπολογισμός της πιθανότητας $P(a_i|a_{i-k} \dots a_{i-1})$ θα χρειαστεί να προσπελαστούν οι τιμές $a_{i-k} \dots a_{i-1}$. Η προσπέλαση απαιτεί πολύ χρόνο για μια on-the-fly προσέγγιση, αλλά και φόρτωση των τιμών, εάν αυτές είναι προϋπολογισμένες και δεν έχουν οργανωθεί σωστά. Η λύση βρίσκεται στα Probabilistic Suffix Trees, τα οποία φορτώνουν προϋπολογισμένες ακολουθίες αποδοτικά.

Probabilistic suffix trees (PST)

Τα suffix trees, αποτελούν δομές αποθήκευσης όλων τον υπακολουθιών μιας ακολουθίας σε μια βάση δεδομένων. Τα PST είναι γενικευμένα suffix trees, που διαθέτουν, μαζί με κάθε ακολουθία, και τις πιθανότητες να παραχθεί στο μέλλον οποιοδήποτε από τα σύμβολα της ακολουθίας μιας βάσης δεδομένων. Ένα PST μεγέθους το πολύ k μπορεί να αποθηκεύσει τις πιθανότητες ενός Markov Model τάξης k. Ένα αρνητικό των suffix trees, είναι ότι ο αριθμός των κόμβων του μπορεί να φτάσει τους $\sum_{i=0}^{k} |\Sigma|^i$. Για την αντιμετώπιση αυτού του προβλήματος χρησιμοποιείται κλάδεμα (pruning). Τα PST είναι ειδικές δομές δεδομένων που διαθέτουν όλες της καταλήξεις (suffixes). Ένας κόμβος που βρίσκεται σε βάθος k, αναπαριστά μια ακολουθία με μήκος k. Κάθε κόμβος έχει ένα διάνυσμα Σ, όπου για κάθε σύμβολο του αλφαβήτου διατίθεται η πιθανότητα παραγωγής του. Η διαδικασία κλαδέματος βελτιστοποιεί τα PST, αφού, αφαιρεί τα επιθέματα που εμφανίζονται σπάνια, και συνεπώς έχουν τη μικρότερη πιθανότητα εμφάνισης.

Εάν διατίθεται μια ακολουθία $a_1 \dots a_i \dots a_n$, και γίνεται εντοπισμός αν το στοιχείο a_i είναι position outlier, τότε, είτε υπολογίζεται η πιθανότητα $P(a_i|a_1 \dots a_{i-1})$, ή μετά από κλάδεμα, ελέγχεται η παράμετρος Short Memory για να βρεθεί το longest suffix, δηλαδή η ακολουθία $a_j \dots a_{i-1}$, για την οποία υπολογίζεται η πιθανότητα $P(a_i|a_j \dots a_{i-1})$.

- Combination outliers: Στην περίπτωση των combination outliers, γίνεται αναζήτηση σε ακολουθίες για συνδυασμούς συμβόλων, που δεν έχουν την αναμενόμενη συμπεριφορά. Διατίθενται ένα σύνολο από ακολουθίες εκπαίδευσης και επίσης, μια ακολουθία επιβεβαίωσης. Η ακολουθία επιβεβαίωσης έχει συνήθως μεγαλύτερο μήκος από τις υπόλοιπες στο σύνολο. Εξ' αιτίας του μεγάλου μήκους της, είναι δύσκολο να κριθεί εάν ολόκληρη η ακολουθία επιβεβαίωσης είναι κανονική ή μη-ομαλή (anomaly). Έτσι, διαχωρίζονται οι ακολουθίες σε παράθυρα, επικαλυπτόμενα ή μη. Τα παράθυρα αυτά ονομάζονται comparison units. Για την εύρεση combination outliers χρησιμοποιείται κάποιο από τα παρακάτω μοντέλα.
    - Distance-based Models: Στα distance-based μοντέλα εύρεσης combination outliers σε ακολουθίες, υπολογίζεται η ομοιότητα του κάθε comparison unit με όλα τα ισοδύναμα σε αυτό παράθυρα μιας ακολουθίας εκπαίδευσης. Από την απόσταση των k-κοντινότερων παραθύρων στην ακολουθία εκπαίδευσης υπολογίζεται το anomaly score. Για την εύρεση της ομοιότητας χρησιμοποιούνται proximity functions και similarity functions όπως οι παρακάτω.
        - Simple matching coefficient: Η συνάρτηση αυτή υπολογίζει την ομοιότητα με βάση τον αριθμό των συμβόλων που είναι ίδια σε δύο ακολουθίες ίδιου μήκους.
        - Normalized longest common subsequence: Η συνάρτηση αυτή, για δύο ακολουθίες T_1 και T_2 εντοπίζει τη μεγαλύτερη κοινή υπακολουθία $L(T_1, T_2)$, και την κανονικοποιεί. Με τη συνάρτηση αυτή, είναι δυνατός ο υπολογισμός της ομοιότητας για δύο ακολουθίες διαφορετικού μήκους, αλλά απαιτεί αρκετό χρόνο.
        - Edit distance: Η συνάρτηση αυτή υπολογίζει την ομοιότητα δύο ακολουθιών με βάση τον ελάχιστο αριθμό αλλαγών (edits) που χρειάζεται η μία για να μετατραπεί στην άλλη. Το υπολογιστικό κόστος της συνάρτησης ενδέχεται να είναι μεγάλο.
        - Compression-based dissimilarity: Η συνάρτηση αυτή προέρχεται από το επιστημονικό πεδίο της Θεωρίας Πληροφορίας. Λειτουργεί

συγκρίνοντας ένα μέτρο που ονομάζεται description length του συνδυασμού δύο ακολουθιών με το άθροισμα των description lengths της κάθε μιας ξεχωριστά.

- Frequency-based Models: Στα frequency-based μοντέλα, επιλέγονται συγκεκριμένα διαστήματα από το χρήστη που ονομάζονται comparison units U_j. Στη συνέχεια, εντοπίζεται η συχνότητα εμφάνισης των comparison units στις ακολουθίες εκπαίδευσης και στην ακολουθία επιβεβαίωσης. Επειδή υπάρχει εξάρτηση των συχνοτήτων αυτών με το μέγεθος της κάθε ακολουθίας, γίνεται κανονικοποίηση των συχνοτήτων. Το anomaly score A, για κάθε ακολουθία εκπαίδευσης T_i υπολογίζεται από την αφαίρεση της κανονικοποιημένης συχνότητας εμφάνισης του comparison unit U_j της ακολουθίας ελέγχου με την αντίστοιχη συχνότητα της ακολουθίας εκπαίδευσης. Η απόλυτη μέση τιμή αυτών των scores αποτελεί το τελικό anomaly score ενός από τα comparison units.

  Συχνότητα εμφάνισης comparison unit U_j στη sequence T: $f(T, U_j)$

  Normalized συχνότητα: $\hat{f}(T, U_j) = \frac{f(T,U_j)}{|T|}$

  Anomaly score: $A(T_i, V, U_j) = \hat{f}(V, U_j) - \hat{f}(T_i, U_j)$

- Hidden Markov Models: Τα μοντέλα αυτά περιγράφονται στην επόμενη ενότητα.

2.1.2.4 Hidden Markov Models

Τα Hidden Markov Models (HMM) είναι πιθανοτικά μοντέλα, που παράγουν ακολουθίες με μια σειρά από μεταβάσεις ανάμεσα στις διάφορες καταστάσεις μιας αλυσίδας Markov. Τα μοντέλα είναι χρήσιμα για την κατηγοριοποίηση, τη συσταδοποίηση και για τον εντοπισμό ακραίων τιμών ακολουθιών. Ενώ στα Markov Models, θεωρούνται γνωστές οι τελευταίες k θέσεις (positions), τα HMM είναι ντετερμινιστικά, δηλαδή σε αυτά δε λαμβάνονται υπόψιν παλαιότερες θέσεις. Οι διάφορες καταστάσεις (states) του μοντέλου δεν είναι φανερές στο χρήστη, ο οποίος διαθέτει μόνο μια ακολουθία από διακριτές παρατηρήσεις που κάνει ο ίδιος. Ο χρήστης λοιπόν, κάνει μια εκτίμηση, που ενδέχεται να μην ταιριάζει με το πραγματικό μοντέλο, και τότε, να επηρεάσει αρνητικά τη διαδικασία μάθησης του μοντέλου. Το μέγεθος του μοντέλου εκτιμάται με βάση την πολυπλοκότητα της εφαρμογής, και μια εσφαλμένη εκτίμηση οδηγεί σε υπερταίριασμα του μοντέλου.

Έστω ότι διατίθεται μια ακολουθία με n καταστάσεις $M = \{s_1 \dots s_n\}$, και ένα σύνολο συμβόλων $\Sigma = \{\sigma_1 \dots \sigma_{|\Sigma|}\}$, που έχει παραχθεί από μεταβάσεις καταστάσεων της ακολουθίας. Μας ενδιαφέρει η εύρεση τριών παραμέτρων.

1. Η πιθανότητα, κατά την οποία το σύμβολο σ_i παράγεται από την κατάσταση s_j, που εκφράζεται ως $\theta_j(\sigma_i)$.
2. Η μετάβαση από την s_i στην s_j, που εκφράζεται ως p_{ij}.
3. Οι πιθανότητες της αρχικής κατάστασης, που εκφράζονται ως $\pi_1 \dots \pi_n$.

Το μοντέλο έχει τοπολογία γραφήματος $G = (M, A)$ με M καταστάσεις και A πιθανές μεταβάσεις. Η διαδικασία εύρεσης των τριών παραμέτρων σε ένα HMM χωρίζεται σε τρείς φάσεις, την αξιολόγηση, την εξήγηση και την εκπαίδευση.

Αξιολόγηση (Evaluation)

Στο στάδιο της αξιολόγησης, διατίθεται ακολουθία εκτίμησης $V = a_i \dots a_m$ ή αντίστοιχα κάποιο comparison unit U_i, και ελέγχεται η πιθανότητα ταιριάσματός του με μοντέλο HMM. Κατά το στάδιο της εκτίμησης γίνεται αναζητούνται όλες οι n^m πιθανές ακολουθίες από states στο HMM, και υπολογίζεται η πιθανότητα κάθε μιας, βασισμένη στην παρατηρούμενη ακολουθία, στην πιθανότητα παραγωγής των συμβόλων της και στην πιθανότητα μετάβασης από κατάσταση σε κατάσταση. Το άθροισμα αυτών των πιθανοτήτων καλείται πιθανότητα ταιριάσματος (fit). Ο υπολογισμός της πιθανότητας ταιριάσματος είναι μια διαδικασία πρακτικά απίθανη, γιατί απαιτεί την απαρίθμηση εκθετικού αριθμού πιθανοτήτων. Για να προσεγγιστεί αυτός ο αριθμός, αξιοποιείται η γνώση ότι τα πρώτα r σύμβολα, που συνοψίζονται στην τιμή της r-στης κατάστασης, μπορούν να υπολογιστούν αναδρομικά από τα $r - 1$ προηγούμενα σύμβολα.

Έστω $a_r(V, s_j)$ η πιθανότητα τα πρώτα r σύμβολα στην ακολουθία V να παραχθούν από ένα μοντέλο, με τελευταία state την s_j.

$$a_r(V, s_j) = \sum_{i=1}^{n} a_{r-1}(V, s_i) \cdot p_{ij} \cdot \vartheta^j(a_r)$$

Η παραπάνω τιμή υπολογίζεται αναδρομικά για $r = 1(1)m$.

Η πρώτη τιμή είναι η $a_1(V, s_j) = \pi_j \cdot \theta^j(a_1)$, αφού δεν έχει προηγούμενες τιμές.

Τελικά, προκύπτει η πιθανότητα ταιριάσματος της ακολουθίας V, που συμβολίζεται ως $F(V)$.

$$F(V) = \sum_{j=1}^{n} a_m(V, s_j)$$

Ο αλγόριθμος υπολογισμού της πιθανότητας ταιριάσματος ονομάζεται Forward Algorithm, και έχει πολυπλοκότητα χρόνου $O(n^2m)$. Κατά τον αλγόριθμο, είναι πολύ πιθανό οι ακολουθίες που ταιριάζουν σε πολύ μικρό βαθμό να είναι μη-ομαλές (anomalies).

Εξήγηση (Explanation)

Κατά το στάδιο της εξήγησης, αναζητείται η καλύτερη δυνατή ακολουθία καταστάσεων που θα μπορούσε να παραχθεί την δοθείσα από το χρήστη, ακολουθία επιβεβαίωσης. Η εξήγηση αν μια ακολουθία ταιριάζει ακριβώς με ένα τμήμα των δεδομένων ή ότι δεν ταιριάζει καθόλου με αυτά, αποτελεί δύσκολο πρόβλημα. Προσεγγιστικά, λοιπόν, αρκεί η εύρεση μιας ακολουθίας που ταιριάζει με τα δεδομένα όσο το δυνατόν καλύτερα. Ο Viterbi Algorithm μπορεί αποδοτικά να εντοπίσει την ακολουθία που ταιριάζει καλύτερα με κάποια δεδομένα. Για την εφαρμογή του αλγορίθμου διατίθεται μια ακολουθία επιβεβαίωσης $V = a_1 \dots a_m$. Στο στάδιο της εξήγησης, αναζητούνται όλες οι n^m πιθανές ακολουθίες καταστάσεων στο HMM, και υπολογίζεται η πιθανότητα της κάθε μιας, βασισμένη στην παρατηρούμενη ακολουθία, στην πιθανότητα παραγωγής των συμβόλων της και στην πιθανότητα μετάβασης από κατάσταση σε κατάσταση. Η μέγιστη από αυτές τις πιθανότητες καλείται most likely path. Ισχύει ότι, κάθε sub-path ενός optimal state path είναι βέλτιστο στην παραγωγή μιας corresponding subsequence από σύμβολα, και έτσι, μπορεί να χρησιμοποιηθεί δυναμικός προγραμματισμός.

Στον υπολογισμό του most likely path, αξιοποιείται η γνώση ότι τα πρώτα r σύμβολα, που συνοψίζονται στην τιμή της r-στης κατάστασης μπορούν να υπολογιστούν αναδρομικά από τα $r-1$ προηγούμενα σύμβολα. Έστω $\delta_r(V, s_j)$ η πιθανότητα της καλύτερης κατάστασης ακολουθίας για την παραγωγή των πρώτων r συμβόλων, στην V με τελευταία κατάσταση την s_j.

$$\delta_r(V, s_j) = MAX_{i=1}^{n}(V, s_j)p_{ij} \cdot \vartheta^j(a_r)$$

Η παραπάνω τιμή υπολογίζεται αναδρομικά για $r = 1(1)m$.

Η πρώτη τιμή είναι η $\delta_1(V, s_j) = \pi_j \cdot \theta^j(a_1)$, αφού δεν έχει προηγούμενες τιμές. Η διαδικασία πραγματοποιείται με πολυπλοκότητα χρόνου $O(n^2m)$.

Εκπαίδευση (Training)

Η εκπαίδευση στα HMM είναι δύσκολη και δεν υπάρχει βέλτιστος αλγόριθμος γι' αυτή. Σε πολλά σενάρια η εκπαίδευση γίνεται αποδοτικά με τον αλγόριθμο Baum-Welch, που

ονομάζεται και Forward-Backward αλγόριθμος και ακολουθεί την ΕΜ προσέγγιση, η οποία περιγράφεται στη συνέχεια. Έστω ότι διατίθεται η ακολουθία $T = a_1 \dots a_m$.

- Forward Probability: Η forward πιθανότητα συμβολίζεται $a_r(T, s_j)$ και αναφέρεται στα πρώτα r σύμβολα της ακολουθίας, με τελευταία κατάσταση την s_j. Η πιθανότητα αυτή υπολογίζεται από το μέτρο ταιριάσματος (fit).
- Backward Probability: Η backward πιθανότητα συμβολίζεται $\beta_r(T, s_j)$ και αναφέρεται στα σύμβολα που βρίσκονται μετά τα r πρώτα σύμβολα της ακολουθίας T, χωρίς να περιλαμβάνει την r-στη θέση (position) της ακολουθίας, και άρα μη περιέχοντας την κατάσταση s_j. Η πιθανότητα αυτή υπολογίζεται από το μέτρο ταιριάσματος, με άθροισμα ανάποδης σειράς, και με αρχικοποίηση $\beta_{|T|}(T, s_j) = 1$.
- $\psi_r(T, s_i, s_j)$: Πιθανότητα η r-στη θέση της T να αντιστοιχίζεται με την κατάσταση s_i, ενώ η (r+1)-στη θέση με την κατάσταση s_j.
- $\gamma_r(T, s_i)$: Πιθανότητα η r-οστή θέση της ακολουθίας T να αντιστοιχίζεται με την κατάσταση s_i.

Η προσέγγιση ΕΜ, ξεκινά με τυχαία αρχικοποίηση των παραμέτρων $\pi(\cdot)$, $\theta(\cdot)$ και p, και στη συνέχεια εκτιμά τις πιθανότητες $\alpha(\cdot)$, $\beta(\cdot)$, $\psi(\cdot)$ και $\gamma(\cdot)$.

- E-step: Κατά το βήμα E-step του ΕΜ αλγορίθμου εκτιμώνται οι πιθανότητες $\alpha(\cdot)$, $\beta(\cdot)$, $\psi(\cdot)$ και $\gamma(\cdot)$ απ' τις παραμέτρους $\pi(\cdot)$, $\theta(\cdot)$ και $p(\cdot)$.

  Για το βήμα αυτό, υπολογίζονται οι πιθανότητες $\alpha(\cdot)$ και $\beta(\cdot)$ από τους Forward και Backward αλγορίθμους καθώς και οι πιθανότητες ψ_r και γ_r.

  - $\psi_r(T, s_i, s_j) = \alpha_r(T, s_i) \cdot p_{ij} \cdot \theta_j(\alpha_{r+1}) \cdot \beta_{r+1}(T, s_j)$, και στη συνέχεια γίνεται κανονικοποίηση έτσι ώστε το άθροισμα των διαφόρων ζευγών να βρίσκεται στο διάστημα $[i, j] = 1$.
  - $\gamma_r(T, s_i) = \psi_r(T, s_i, s_j)$, με σταθερή παράμετρο i, και μεταβλητή παράμετρο j.
- M-step: Κατά το βήμα M-step του ΕΜ αλγορίθμου εκτιμώνται οι παράμετροι $\pi(\cdot)$, $\theta(\cdot)$ και $p(\cdot)$ απ' τις πιθανότητες $\alpha(\cdot)$, $\beta(\cdot)$, $\psi(\cdot)$ και $\gamma(\cdot)$.
  - Για το βήμα αυτό χρησιμοποιείται η binary indicator function $I(a_r, \sigma_k)$, η οποία έχει την τιμή 1 για δύο όμοια σύμβολα, και την τιμή 0 διαφορετικά.

$$\pi(j) = \gamma_1(T, s_j), \qquad p_{ij} = \frac{\sum_{r=1}^{m-1} \psi_r(T, s_i, s_j)}{\sum_{r=1}^{m-1} \gamma_r(T, s_i)},$$

$$\theta^i(\sigma_\kappa) = \frac{\sum_{r=1}^{m} I\,(a_r, \sigma_k) \cdot \gamma_r(T, s_i)}{\sum_{r=1}^{m} \gamma_r\,(T, s_i)}$$

2.1.2.5 Κατηγοριοποίηση ακολουθιών (classification)

Η κατηγοριοποίηση, είναι η διαδικασία κατά την οποία διαχωρίζονται οι ακολουθίες σε κατηγορίες, διακρίνονται από ετικέτες. Για τη διαδικασία αυτή, διατίθενται ένα μοντέλο εκπαίδευσης N ακολουθιών $S_1 \dots S_N$ και κάποιες ετικέτες κατηγοριών $1(1)k$, και ζητείται η δημιουργία ενός μοντέλου που θα δέχεται άγνωστες ακολουθίες και θα καθορίζει γι' αυτές κατάλληλες ετικέτες. Στη συνέχεια αναφέρονται κάποιες από τις μεθόδους που χρησιμοποιούνται για κατηγοριοποίηση ακολουθιών.

- Nearest neighbor classifier: Κατά τη μέθοδο nearest neighbor, για κάθε τμήμα της ακολουθίας επιβεβαίωσης, αναζητούνται οι k-κοντινότεροι γείτονες στα δεδομένα εκπαίδευσης με τη χρήση μιας συνάρτησης ομοιότητας ακολουθιών. Το τμήμα της ακολουθίας εκπαίδευσης, θα χαρακτηριστεί με την ετικέτα που διαθέτει ο κοντινότερος από τους γείτονές της. Το μέγεθος των γειτόνων k μπορεί να υπολογισθεί βέλτιστα με την τεχνική leave-one-out cross-validation. Η επιλογή της συνάρτησης ομοιότητας επηρεάζει την αποδοτικότητα της μεθόδου, και ακόμα, δεν μπορούν να επιλεγούν καθολικές συναρτήσεις ομοιότητας λόγω της ύπαρξης τμημάτων με θόρυβο στις ακολουθίες. Συνήθως, χρησιμοποιούνται keyword-based συναρτήσεις ομοιότητας, όπου παράγονται n-grams από μια ακολουθία και συγκρίνεται η ομοιότητά των n-grams ως διανύσματα.
- Graph-based Methods: Οι Graph-based μέθοδοι για κατηγοριοποίηση ακολουθιών αποτελούν semisuperivised αλγορίθμους, γιατί συνδυάζουν τη γνώση των τμημάτων εκπαίδευσης και επιβεβαίωσης μαζί. Στις μεθόδους αυτές, τα δεδομένα αναπαρίστανται σε ένα γράφημα $G(V, A)$, με σύνολο κόμβων V τον συνδυασμό των τμημάτων εκπαίδευσης και επιβεβαίωσης, και σύνολο ακμών A τις μη κατευθυνόμενες ακμές που συνδέουν τους k-κοντινότερους γείτονες κάθε κόμβου. Οι κόμβοι του G, που έχουν προκύψει από τα τμήματα εκπαίδευσης χαρακτηρίζονται από το χρήστη με τις κατάλληλες ετικέτες, ενώ οι κόμβοι των τμημάτων επιβεβαίωσης είναι μη χαρακτηρισμένοι. Ζήτημα του αλγορίθμου είναι να βρεθούν οι ετικέτες των κόμβων των τμημάτων επιβεβαίωσης.
- Rule-based methods: Οι rule-based μέθοδοι είναι ιδιαίτερα χρήσιμες σε ακολουθίες που διαθέτουν θορυβώδη τμήματα ή τμήματα που δε σχετίζονται με κάποια από τις διαθέσιμες ετικέτες. Στις μεθόδους αυτές, χρησιμοποιείται η

τεχνική binarization, κατά την οποία γίνεται μετατροπή των διακριτών ακολουθιών σε δυαδικές χρονοσειρές, ίδιου μήκους με τις ακολουθίες. Ύστερα, οι δυαδικές χρονοσειρές μπορούν να μετατραπούν έτσι, ώστε να ακολουθούν την αναπαράσταση πολυδιάστατων κυματομορφών (multidimensional wavelet representations). Η δεύτερη μετατροπή γίνεται με αντιστοίχιση της κάθε χρονοσειράς σε πολυδιάστατο διάνυσμα χρησιμοποιώντας τον μετασχηματισμό κυματομορφής (wavelet transformation) και ψηφιοποιώντας το αποτέλεσμα. Ο συνδυασμός των χαρακτηριστικών των πολυδιάστατων αυτών δεδομένων, συμβάλει στη δημιουργία ενός συνόλου κανόνων. Μια νέα ακολουθία, αφού αρχικά μετασχηματιστεί με τον τρόπο που αναφέρθηκε, μπορεί να κατηγοριοποιηθεί με βάση την τήρηση ή όχι αυτών των κανόνων.

- Kernel support vector machines: Οι kernel support vector machines μπορούν να κατασκευάσουν κατηγοριοποιητές (classifiers) με τη χρήση της kernel similarity ανάμεσα στα τμήματα εκπαίδευσης και επιβεβαίωσης. Οι μηχανές αυτές μπορούν να αγνοήσουν τα χαρακτηριστικά (features) των δεδομένων, διαθέτοντας μόνο τις kernel-based similarities $K(Y_i, Y_j)$ ανάμεσα σε κάθε ζεύγος από τις ακολουθίες. Κατά την κατηγοριοποίηση ακολουθιών, με τη χρήση τέτοιων κατηγοριοποιητών, χειριζόμαστε τις ακολουθίες ως αλφαριθμητικά (strings). Ακολουθούν κάποιοι kernels που χρησιμοποιούνται.
  - Bag-of-words kernel: Στον bag-of-words kernel, τα αλφαριθμητικά αντιμετωπίζονται ως σακίδια αλφαβήτων (bags of alphabets). Ο feature map $\Phi(\cdot)$, που μοιάζει με τον vector-space μετασχηματισμό μετατρέπεται ένα αλφαριθμητικό σε vector-space αναπαράσταση. Αν $\overline{V(Y_i)}$ η vector-space αναπαράσταση ενός αλφαριθμητικού, τότε η kernel-based similarity περιγράφεται με τον ακόλουθο τρόπο.

$$\Phi(Y_i) = \overline{V(Y_i)}$$

$$K(Y_i, Y_j) = \Phi(Y_i) \cdot \Phi(Y_j) = \overline{V(Y_i)} \cdot \overline{V(Y_i)}$$

    Το κύριο πρόβλημα του kernel αυτού είναι ότι χάνεται η πληροφορία της ακολουθιακής σειράς στα αλφάβητα. Ως αποτέλεσμα αυτού, ο kernel είναι αποδοτικός για μεγάλα αλφάβητα, ενώ πρακτικά άχρηστος για μικρά αλφάβητα.
  - Spectrum kernel: Ο spectrum kernel, αποτελεί λύση στο πρόβλημα των bag-of-words kernels, εξάγοντας k-άδες συμβόλων από τα αλφαριθμητικά για την κατασκευή του vector-space representation, αντί για μεμονωμένα σύμβολα.

Για παράδειγμα σε μια ακολουθία DNA, "$ATGCGATGG$" με αλφάβητο $\Sigma = \{A, C, T, G\}$ και μέγεθος τμήματος $k = 3$, προκύπτουν τα εξής αποτελέσματα.

$$ATG(2), TGC(1), GCG(1), CGA(1), GAT(1), TTG(1)$$

, όπου στις παρενθέσεις αναγράφεται η συχνότητα εμφάνισης κάθε k-άδας. Η αναπαράσταση αυτή συμβάλει στη δημιουργία ενός feature map $\Phi(\cdot)$.

Η πιο απλή προσέγγιση ελέγχει όλες τις k-άδες ενός αλφαριθμητικού. Για να προκύψει ένα καλύτερο αποτέλεσμα, χρησιμοποιείται η mismatch neighborhood στον kernel. Τότε, αντί να προστεθούν σε κάθε k-άδα μόνο οι εμφανίσεις της, προθέτονται επίσης σε αυτή οι εσφαλμένες εμφανίσεις της κατά m σύμβολα. Για παράδειγμα, με $m = 1, k = 3$ και την k-άδα ATG , στην θέση της, θα προστεθεί εκτός από τις εμφανίσεις της, η εμφάνιση των:

$$CTC, GTG, TTG, ACG, AAG, AGG, ATC, ATA, ATT$$

Με αυτή την προσέγγιση μπορεί να αντιμετωπισθεί η παρουσία θορύβου στα δεδομένα που διαχειρίζεται κάποιος χρήστης.

Ο spectrum kernel λειτουργεί αποδοτικά για μια δομή trie ή μια δομή suffix tree. Το κύριο θετικό χαρακτηριστικό ενός τέτοιου kernel είναι η αποδοτική εύρεση ομοιότητας σε αλφαριθμητικά, ακόμη και διαφορετικού μήκους.

- Weighted degree kernel: Ο weighted degree kernel, διατηρεί τη σειρά ανάμεσα στις k-άδες αλφαριθμητικών, ορίζοντας άμεσα την $K(Y_i, Y_j)$ χωρίς να έχει ορίσει κάποιο feature map $\Phi(\cdot)$ για τα αλφαριθμητικά.

Probabilistic Methods: Κατά τις probabilistic μεθόδους δημιουργείται ένα HMM για κάθε τάξη δεδομένων, και το HMM αυτό εκπαιδεύεται με τον Baum-Welch αλγόριθμο και υπολογίζεται η πιθανότητα ταιριάσματος (fit).

Κεφάλαιο 3. Σχεδίαση & Υλοποίηση

Στο παρόν κεφάλαιο ορίζονται τα βασικά στοιχεία που χρειάζονται για πρόβλεψη σε χρονοσειρές με μοντέλα ARIMA, τα οποία λαμβάνονται υπόψιν στη σχεδίαση μιας εφαρμογής παραγωγής τέτοιων προβλέψεων. Με τα εφόδια αυτά, σχεδιάζεται και περιγράφεται ο κύριος αλγόριθμος της εφαρμογής. Ακόμη, αναλύεται η διαδικασία επιλογής και εγκατάστασης κάποιων απαραίτητων βιβλιοθηκών. Στη συνέχεια, αναλύεται η αρχιτεκτονική του παραχθέντος λογισμικού, γίνονται έλεγχοι σε αυτό και παρουσιάζονται τα αποτελέσματά τους. Τέλος, περιγράφεται η ακολουθία βημάτων που απαιτούνται για την εκτέλεση του λογισμικού και αναφέρονται νέες ιδέες για επέκταση του. Η εφαρμογή βρίσκεται ηλεκτρονικά στο σύνδεσμο https://github.com/othonasgkavardinas/java_timeseries_arima.

3.1 Πρόβλεψη χρονοσειρών με μοντέλα ARIMA

Στο πλαίσιο αυτής της διπλωματικής εργασίας, χρησιμοποιείται το μοντέλο ARIMA, για την πρόβλεψη μελλοντικών τιμών σε χρονοσειρές. Στη συνέχεια, αναλύονται έννοιες που χρησιμοποιούνται κατά τη σχεδίαση και υλοποίηση ενός τέτοιου μοντέλου, σύμφωνα με την πηγή [Jebb15].

3.1.1 Ορολογία

Ξεκινούμε με τους βασικούς όρους:

Τάση (Trend): Η τάση είναι το χαρακτηριστικό μιας χρονοσειράς, κατά το οποίο, η καμπύλη των δεδομένων της ακολουθεί μια ανοδική ή καθοδική κλίση. Η κλίση αυτή, ενδέχεται να διατηρείται σταθερή ή να μεταβάλλεται. Η μέση τιμή σε μία τάση μεταβάλλεται.

Εποχικότητα (Seasonality): Η εποχικότητα είναι ένα χαρακτηριστικό μιας χρονοσειράς, κατά το οποίο, η καμπύλη των δεδομένων αποτελείται από επαναλήψεις κάποιου μοτίβου αυξήσεων και μειώσεων σταθερού μήκους. Τα μοτίβα παρατηρούνται σε διάφορες κλίμακες διαστημάτων. Για παράδειγμα, εποχικότητα μπορεί να εμφανιστεί εβδομαδιαία, ή και μηνιαία.

Αυτοσυσχέτιση (Autocorrelation): Η αυτοσυσχέτιση είναι ένα χαρακτηριστικό μιας χρονοσειράς, που αποτιμά το βαθμό στον οποίο κάθε τιμή της εξαρτάται από τις προηγούμενες από αυτή τιμές.

Καθυστέρηση (Lag): Ο όρος αυτός, αφορά προηγούμενες χρονικές περιόδους μιας τιμής χρονοσειράς. Για παράδειγμα, η lag-1 αυτοσυσχέτιση, παρατηρείται όταν κάθε τιμή σε μια χρονοσειρά εξαρτάται από την αμέσως προηγούμενή της, ενώ η lag-4 αυτοσυσχέτιση, παρατηρείται όταν κάθε τιμή εξαρτάται από τις 4 προηγούμενες τιμές.

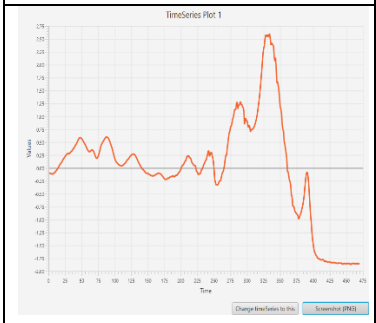
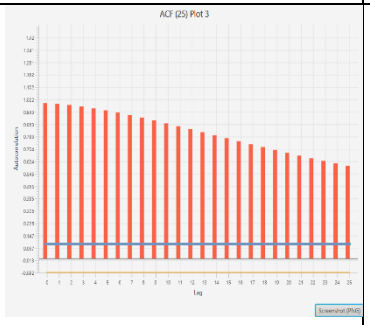
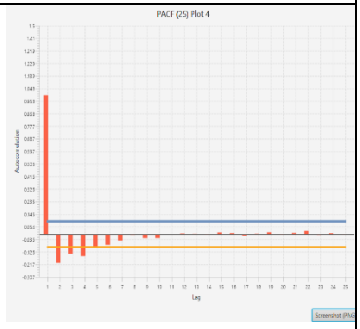
Συνάρτηση αυτοσυσχέτισης (Autocorrelation Function – ACF): Η συνάρτηση αυτοσυσχέτισης, αναπαριστά την ισχύ της αυτοσυσχέτισης σε μια χρονοσειρά για διάφορες καθυστερήσεις. Η μορφή της ACF, βοηθά στην κατάλληλη επιλογή παραμέτρων MA(q) ενός μοντέλου ARIMA. Η καθυστέρηση, της οποίας η αναπαράστασή θα εμφανίσει απότομη πτώση, είναι εκείνη που θα καθορίσει την παράμετρο q.

Μερική συνάρτηση αυτοσυσχέτισης (Partial Autocorrelation Function – PACF): Η συνάρτηση ACF ενδέχεται να προβάλλει την ισχύ κάποιων καθυστερήσεων οι οποίες να είναι επηρεασμένες από προηγούμενες καθυστερήσεις. Για παράδειγμα, η lag-3 αυτοσυσχέτιση, εμπεριέχει ως ποσότητα, τις lag-1 και lag-2 αυτοσυσχετίσεις. Η συνάρτηση PACF αντιμετωπίζει το φαινόμενο αυτό, μελετώντας την κάθε καθυστέρηση, χωρίς να λαμβάνει υπ' όψη τις προηγούμενες. Η μορφή της PACF, βοηθά στην κατάλληλη επιλογή παραμέτρων AR(p) ενός μοντέλου ARIMA. Η καθυστέρηση, της οποίας η αναπαράστασή θα εμφανίσει απότομη πτώση, είναι εκείνη που θα καθορίσει την παράμετρο p.

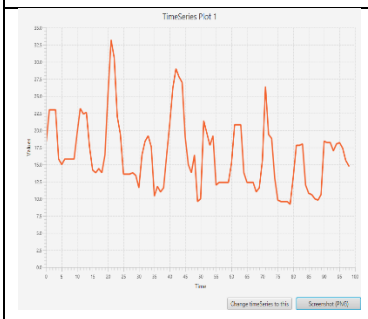
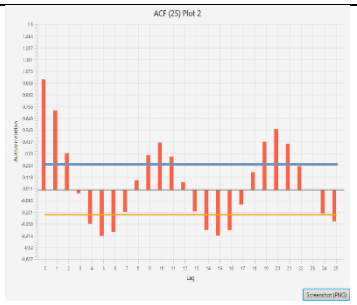
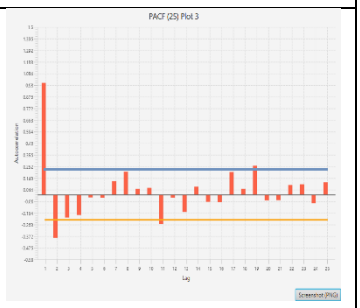
Όρια των γραφημάτων ACF και PACF: Τα όρια υπολογίζονται από τον τύπο:

$$-\left(\frac{1}{N}\right) \pm \left(\frac{2}{\sqrt{N}}\right)$$

όπου N το μέγεθος της χρονοσειράς αναλύεται.

Original timeseries	ACF until lag-25	PACF until lag-25
		

Σχήμα 3.1 *Χρονοσειρά από τα δεδομένα Beef_TEST του UCR Archive, μαζί με τα γραφήματα ACF και PACF μέχρι και lag-25. Στο PACF, φαίνεται ότι θα μπορούσε να επιλεχθεί ως παράμετρος q η τιμή 4.*

Original timeseries	ACF until lag-25	PACF until lag-25
		

Σχήμα 3.2 *Χρονοσειρά από τα δεδομένα household_preview μαζί με τα γραφήματα ACF και PACF μέχρι και lag-25. Στο ACF γράφημα, φαίνεται πως υπάρχει εποχικότητα. Στο PACF, φαίνεται ότι θα μπορούσε να επιλεχθεί ως παράμετρος q η τιμή 2, 11 ή 19.*

Εργοδικότητα (Stationarity): Μια χρονοσειρά καλείται εργοδική, αν διαθέτει σταθερή μέση τιμή, σταθερή διακύμανση και ακόμη, αν δεν παρουσιάζει εποχικότητα. Οι εργοδικές χρονοσειρές, λόγω των χαρακτηριστικών τους, είναι πολύ εύκολο να προβλεφθούν. Για να καταφέρουμε να μετατρέψουμε μια χρονοσειρά σε εργοδική, πρέπει κάποια από τα χαρακτηριστικά της να διαγραφούν ή να μοντελοποιηθούν. Συγκεκριμένα, οι τάσεις, από τον ορισμό τους, δεν έχουν σταθερή μέση τιμή, και επίσης,

η μέση τιμή μεταβάλλεται με την εποχικότητα. Μια χρονοσειρά, είναι απαραίτητο να είναι εργοδική, έτσι ώστε να εφαρμοστεί σε αυτή ένα μοντέλο ARIMA.

Ακολουθούν κάποιοι επιπλέον ορισμοί με βάση την πηγή [HA2018].

Υπολειπόμενη ποσότητα (residuals): Ως υπολειπόμενη ποσότητα σε ένα μοντέλο χρονοσειρών, χαρακτηρίζεται μια χρονοσειρά με στοιχεία, εκείνα που περισσεύουν στο ταίριασμα μοντέλου-δεδομένων. Συνήθως, η υπολειπόμενη ποσότητα εκφράζει τη διαφορά των παρατηρούμενων τιμών με εκείνες που προκύπτουν από το μοντέλο, δηλαδή

$$e_t = y_t - \hat{y}_t$$

Η ποσότητα αυτή προσφέρει πληροφορία σχετικά με την ποιότητα του μοντέλου στο οποίο ανήκει. Συγκεκριμένα, ένα μοντέλο που έχει ταιριάξει αρκετά καλά με τα δεδομένα εισόδου, πρέπει να διαθέτει μηδενική μέση τιμή, και να μην υπάρχουν συσχετίσεις μεταξύ των υπολειπόμενων ποσοτήτων του του.

Akaike's Information Criterion (AIC): Το AIC, είναι μια βασική μετρική, με την οποία μπορούν να συγκριθούν δύο ή περισσότερα μοντέλα ARIMA. Το μοντέλο που διαθέτει τη μικρότερη τιμή AIC, είναι συνήθως αυτό που ταιριάζει περισσότερο στα δεδομένα εισόδου. Σύμφωνα με την πηγή [StatHT1]ο ορισμός του μοντέλου είναι ο εξής:

$$AIC = -2(log\_likelihood) + 2K$$

Στην εξίσωση, όπου K είναι ο αριθμός των παραμέτρων του μοντέλου, και $log\_likelihood$ είναι ένα μέτρο του ταιριάσματος (fit) του μοντέλου. Όσο μεγαλύτερο είναι το $log\_likelihood$, τόσο καλύτερα έχει ταιριάξει το μοντέλο.

Σύνολα εκπαίδευσης και επιβεβαίωσης (Training and test sets): Η επιβεβαίωση της ορθής λειτουργίας ενός μοντέλου παραγωγής προβλέψεων για χρονοσειρές, απαιτεί το διαχωρισμό των δεδομένων σε δύο βασικά σύνολα. Το πρώτο, είναι το σύνολο εκπαίδευσης, το οποίο διαθέτει το μεγαλύτερο μέρος των δεδομένων και χρησιμοποιείται για το ταίριασμα μοντέλου-δεδομένων. Το δεύτερο, είναι το σύνολο επιβεβαίωσης, το οποίο, αμέσως μετά το ταίριασμα του μοντέλου, χρησιμοποιείται για τη σύγκριση των προβλέψεων που παράγει το μοντέλο, με πραγματικές τιμές.

Όρια στην πρόβλεψη: Το αποτέλεσμα μιας πρόβλεψης από ένα μοντέλο χρονοσειρών, είναι μια στοχαστική διαδικασία, η οποία διαθέτει ένα άνω, και αντίστοιχα ένα κάτω φράγμα, που χαρακτηρίζουν το χώρο στον οποίο θα βρίσκονται οι νέες τιμές.

Μοντέλο ARIMA: Το μοντέλο ARIMA και οι παραλλαγές του είναι από τα καλύτερα μοντέλα για την πρόβλεψη μελλοντικών τιμών σε χρονοσειρές, που χαρακτηρίζονται από αυτοσυσχέτιση. Το μοντέλο αυτό, διαθέτει τρία βασικά τμήματα, τα οποία απαιτούν παραμετροποίηση από το χρήστη. Οι παράμετροι που θα εισάγει ο χρήστης, θα έχουν καθοριστικό ρόλο στην ποιότητα των παραγόμενων προβλέψεων. Στη συνέχεια θα γίνει αναφορά των τριών αυτών βασικών τμημάτων.

I(d) Integrated: Το τμήμα αυτό είναι υπεύθυνο για τη μετατροπή μιας χρονοσειράς εισόδου, σε εργοδική. Είναι το πρώτο βήμα που ακολουθείται στην παραγωγή ενός τέτοιου μοντέλου, διότι προσδίδει στη χρονοσειρά χαρακτηριστικά που είναι απαραίτητα για την παραγωγή προβλέψεων. Εφαρμόζονται λοιπόν, διαφορές τάξης d. Η τάξη είναι μια παράμετρος που επιλέγεται από το χρήστη. Συνήθως, η επιλογή της παραμέτρου περιορίζεται στην πρώτη, ή και δεύτερη τάξη. Ένας εμπειρικός κανόνας στην επιλογή της τάξης είναι ότι, η καλύτερη τάξη είναι εκείνη που θα προσφέρει τη μεγαλύτερη μείωση της διακύμανσης. Αντιθέτως, η αύξηση της διακύμανσης είναι ένας εμπειρικός κανόνας υπερ-ταιριάσματος.

AR(p) Autoregressive: Η αυτό-παλινδρόμηση αποτελεί σημαντικό μέτρο μοντελοποίησης της αυτοσυσχέτισης. Το τμήμα αυτό του μοντέλου, ακολουθεί τη λογική ότι, οι τιμές μιας χρονοσειρά μοντελοποιούνται ως συνάρτηση των προηγούμενων τιμών. Έτσι, η πρόβλεψη για νέες τιμές θα βασίζεται στις αμέσως προηγούμενές της. Είναι σημαντική η επιλογή του πλήθους των προηγούμενων τιμών που απαιτείται για κάθε τιμή. Οι k προηγούμενες τιμές, καλούνται παρατηρήσεις καθυστέρησης-k. Η παράμετρος k, πρέπει να οριστεί από το χρήστη, και η επιλογή της είναι καθοριστική για την ποιότητα των προβλέψεων του μοντέλου. Ένας καλός εμπειρικός κανόνας για την επιλογή των παραμέτρων είναι η παρατήρηση μιας αργής μείωσης της ισχύος στην αναπαράσταση της ACF και επίσης μιας απότομης πτώσης της ισχύος μετά από κάποιο lag p στην αναπαράσταση PACF. Επίσης, ένας εμπειρικός κανόνας επιβεβαίωσης του ότι έχουν συμπεριληφθεί όλες οι αυτοσυσχετίσεις, είναι η παρατήρηση της υπολειπόμενης ποσότητας, η οποία πρέπει να έχει τη μορφή λευκού τυχαίου θορύβου.

MA(q) Moving-Average: Με τη χρήση του τμήματος MA, ένα μοντέλο ARIMA, μπορεί να ενσωματώσει και τυχαία shocks. Τα shocks είναι τιμές που επηρεάζουν τη συμπεριφορά μιας χρονοσειράς, χωρίς όμως να εμφανίζονται στην ίδια τη χρονοσειρά με κάποια μορφή. Τα shocks αποτελούν συσχέτιση της χρονοσειράς με την υπολειπόμενη σε αυτή ποσότητα. Συγκεκριμένα, το τμήμα αυτό βασίζεται στο γεγονός ότι, οι τιμές του σφάλματος των προηγούμενων τιμών μπορεί να επηρεάζουν τις επόμενες τιμές. Το πλήθος q των προηγούμενων τιμών σφάλματος που λαμβάνονται για κάθε τιμή, επιλέγεται από το χρήστη, και είναι καθοριστικό, για την ποιότητα των προβλέψεων μελλοντικών τιμών. Ένας εμπειρικός κανόνας για την παράμετρο q, είναι αν υπάρχει μια σταθερή μείωση της ισχύος της αναπαράστασης της PACF, και ταυτόχρονα μια απότομη πτώση μετά από q lags στην αναπαράσταση της ACF. Ο συνδυασμός των AR και MA όρων μειώνει το πλήθος των παραμέτρων, μειώνοντας έτσι και την πολυπλοκότητα του μοντέλου, σε σύγκριση με τα μοντέλα με αποκλειστικούς όρους AR ή MA.

3.1.2 Περιγραφή διαδικασίας πρόβλεψης με μοντέλα ARIMA

Με βάση το θεωρητικό υπόβαθρο για τα μοντέλα ARIMA παρουσιάζεται η ακολουθία βημάτων που απαιτείται για την πραγματοποίηση προβλέψεων σε χρονοσειρές μέσω αυτών των μοντέλων. Η ροή αυτή προήλθε από την πηγή [Jebb15].

1. Εισαγωγή της χρονοσειράς εισόδου, προβολή του διαγράμματος ACF και αναζήτηση σε αυτό, για διάφορες καθυστερήσεις.
   1.1. Εάν δεν παρατηρείται η ύπαρξη μεγάλων τιμών αυτοσυσχέτισης:
      1.1.1. Αποτυχία και **ΕΞΟΔΟΣ**.
   1.2. Αλλιώς: **Πήγαινε 2**.
2. Προβολή διαγράμματος χρονοσειράς εισόδου και έλεγχος εργοδικότητας.
   2.1. Αν είναι εργοδική (δηλαδή αν διαθέτει σταθερή μέση τιμή, σταθερή διακύμανση και δεν έχει εποχικότητα): **Πήγαινε 4**.
   2.2. Αλλιώς: **Πήγαινε 3**.
3. Μετατροπή της χρονοσειράς εισόδου σε εργοδική.
   3.1. Αν η διακύμανση δεν είναι σταθερή:
      3.1.1.Εφαρμογή φυσικού λογαρίθμου. (βλ. Ενότητα 2.1.1.2)
   3.2. Αν παρατηρείται εποχικότητα:
      3.2.1.Διαγραφή εποχικότητας.

3.3. Αν παρατηρείται τάση:

3.3.1. Εφαρμογή διαφορών.

4. Διαχωρισμός των δεδομένων σε σύνολα εκπαίδευσης και ελέγχου. Συνήθως, επιλέγεται το πρώτο 80% των σημείων για την εκπαίδευση, και το υπόλοιπο 20% για τον έλεγχο.
5. Προβολή των διαγραμμάτων ACF και PACF. Από αυτά, μπορεί να διακριθεί εάν χρειάζεται η προσθήκη όρων AR και MA.

5.1. Αν το ACF πέφτει ομαλά, ενώ το PACF απότομα μετά από p lags:

5.1.1.Προσθήκη p AR όρων.

5.2. Αν το PACF πέφτει ομαλά, ενώ το ACF απότομα μετά από q lags:

5.2.1.Προσθήκη q MA όρων.

6. Σε ένα επιτυχημένο μοντέλο, η υπολειπόμενη ποσότητα δεν πρέπει να περιλαμβάνει αυτοσυσχετίσεις. Οποιαδήποτε αυτοσυσχέτιση σε αυτή δείχνει πως το μοντέλο μπορεί να επιλεχθεί καλύτερα.

Γίνεται προβολή της υπολειπόμενης ποσότητας.

6.1. Αν η υπολειπόμενη ποσότητα δεν είναι λευκός τυχαίος θόρυβος:

6.1.1. **Πήγαινε 5**.

6.2. Παράγωγή μοντέλου ARIMA.

7. Προβολή διαθέσιμων μοντέλων ARIMA. (αρχικά 1).

7.1. Επιθυμεί ο χρήστης να σχεδιάσει περισσότερα μοντέλα;

7.1.1. **Πήγαινε 5**.

7.2. Επιλογή μοντέλου με βάση το δείκτη AIC. (Akaike's Information Criterion)

8. Επαλήθευση της ικανότητας παραγωγής πρόβλεψης του μοντέλου που επιλέχθηκε.

8.1. Ενσωμάτωση δεδομένων του συνόλου εκπαίδευσης στο μοντέλο.

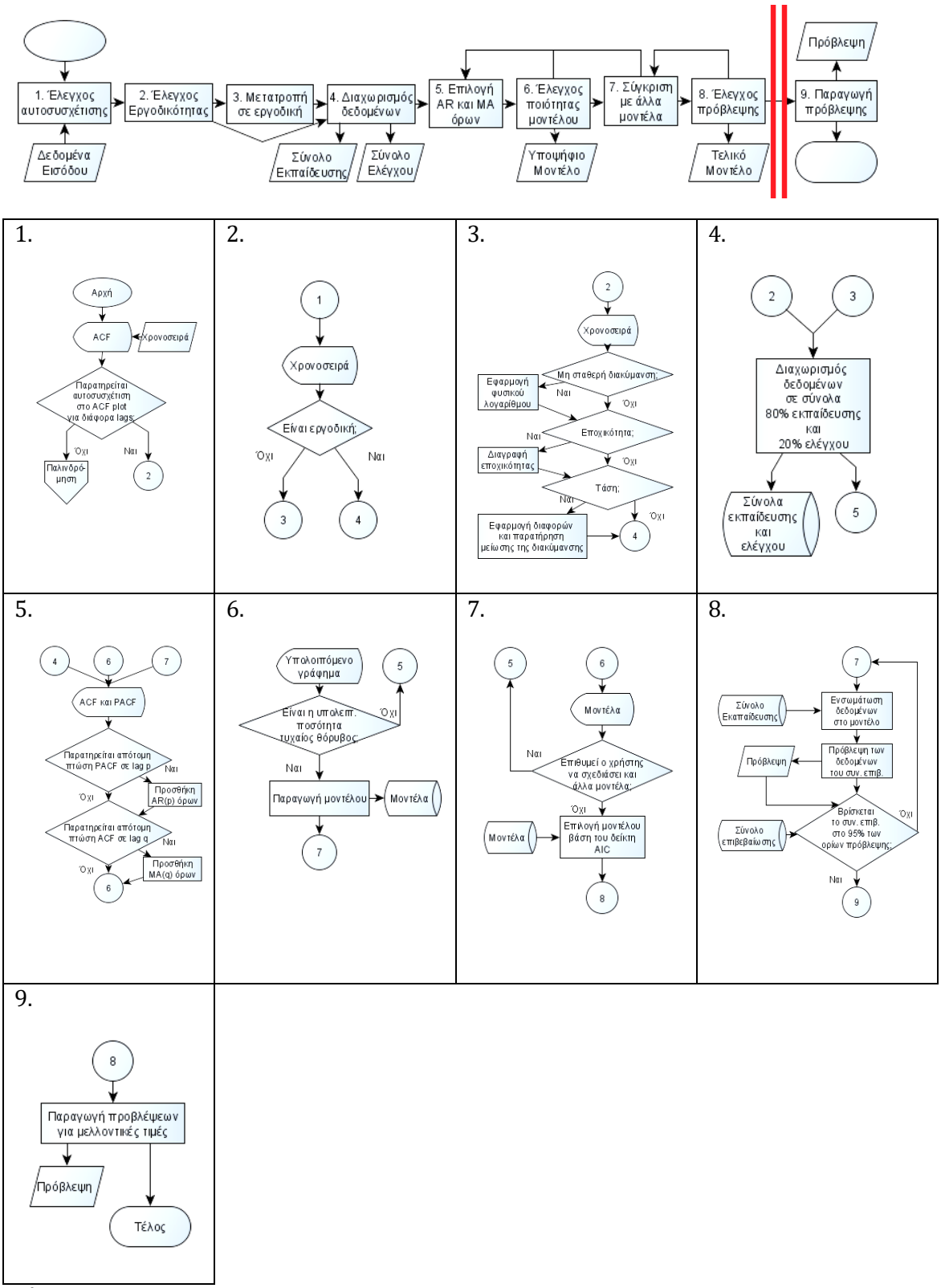
8.2. Παραγωγή πρόβλεψης των στοιχείων του συνόλου ελέγχου.

8.3. Αν δεν βρίσκονται οι τιμές του συνόλου εκπαίδευσης στο 95% των ορίων πρόβλεψης.

8.3.1. **Πήγαινε 7**.

9. Παραγωγή προβλέψεων για μελλοντικές τιμές.

Η παραπάνω διαδικασία αναπαρίσταται στο Σχήμα 3.3.



Σχήμα 3.3 *Ροή εργασίας για μοντελοποίηση και πρόβλεψη με τη μέθοδο ARIMA.*

3.2 Μελέτη υπάρχοντος λογισμικού

Στην ενότητα αυτή εξετάζονται διάφορες βιβλιοθήκες που αφορούν χρονοσειρές από το Github. Σκοπός είναι η συλλογή πληροφοριών που χρειάζονται στη σχεδίαση και υλοποίηση της εφαρμογής πρόβλεψης.

Αρχικά ελέγχθηκαν οι εξής βιβλιοθήκες δημιουργημένες στη γλώσσα προγραμματισμού Java για τη διαχείριση και υποστήριξη πρόβλεψης σε χρονοσειρές.

- https://github.com/signaflo/java-timeseries
- https://github.com/patrickzib/SFA
- https://github.com/seninp/HOTSAX
- https://github.com/Workday/timeseries-forecast
- https://github.com/elki-project/elki

Από τις βιβλιοθήκες επιλέχθηκαν οι java-timeseries και timeseries-forecast, αφού υποστηρίζουν μοντέλα πρόβλεψης ARIMA. Στη συνέχεια, αναλύθηκαν και συγκρίθηκαν οι δύο βιβλιοθήκες. Τα αποτελέσματα της σύγκρισης παρουσιάζονται στον *Πίνακα 3.1*.

Από τα συστήματα του πίνακα επιλέχθηκε το java-timeseries του signaflo, στο οποίο, στο εξής θα αναφερόμαστε ως java-timeseries. Χρειάσθηκε κατάλληλη επεξεργασία της βιβλιοθήκης, για την ορθή λειτουργία της. Συγκεκριμένα,

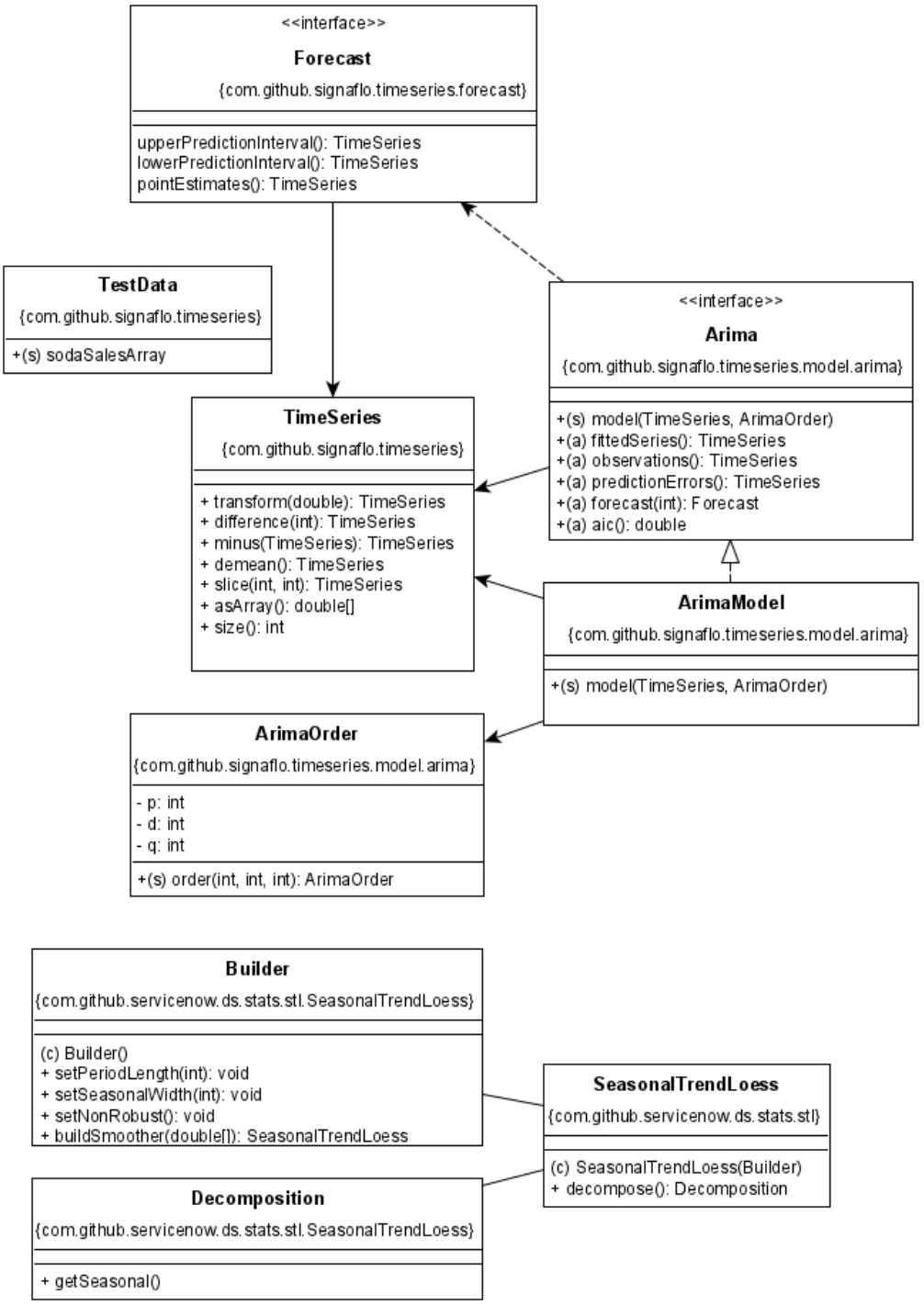
- Εγκαταστάθηκε το εργαλείο gradle έκδοσης 4.8.
- Εγκαταστάθηκε η Java 1.8, για συγχρονισμό με το εργαλείο gradle.
- Ανακτήθηκε η βιβλιοθήκη από τη σελίδα του github.
- Στην πλατφόρμα δημιουργίας εφαρμογών σε Java Eclipse, η βιβλιοθήκη έγινε 'Import as gradle project'.
- Ύστερα από την ενσωμάτωση, δύο από τις κλάσεις της βιβλιοθήκης, εμφάνιζαν κάποια σφάλματα.
- Βρέθηκε ότι, πιθανώς τα προβλήματα προκαλούνταν από κάποια αδυναμία συγχρονισμού της βιβλιοθήκης lombok-1.18, η οποία χρησιμοποιείται ως dependency στην βιβλιοθήκη προς ενσωμάτωση.
- Το πρόβλημα αντιμετωπίσθηκε προσθέτοντας κάποιους setters και getters.
- Η βιβλιοθήκη μετατράπηκε σε αρχεία βιβλιοθήκης jar. (δύο αρχεία timeseries.jar και math.jar)

	https://github.com/signaflo/java-timeseries	https://github.com/Workday/timeseries-forecast
Καλή σχεδίαση	Ναι	Ναι
Documentation	Καλό	Αρκετά καλό
Ποιότητα Κώδικα	Καλή	Αρκετά καλή
Εύκολος εντοπιστμός-σφαλμάτων	Ναι	Ναι
Έλεγχοι	Ναι	Ναι
Περνάνε Σωστά;	Ναι (~90%)	Ναι
Είδος ελέγχων	Αναλυτικά και οργανωμένα	Αναλυτικά
Github commits	637	97
Github stars	122	45
Github issues	30	1
Github fixed issues	7	1
Github last commit	29/10/2019	20/5/2019
Αριθμός ατόμων που ασχολούνται	ατομικό	ομαδικό
Αριθμός εξαρτήσεων	15	1
Άδεια Χρήσης	Free	MIT
Χρονοσειρές - Μαθηματικά	Βρίσκονται σε ξεχωριστά πακέτα κλάσεων	Βρίσκονται μαζί, αλλά οργανωμένα
Μετρικές	AIC, ACF, Ljung-Box, RMSE	Hannan Rissanen, RMSE
Οπτικοποίηση χρονοσειράς	Ναι	Όχι
Οπτικοποίηση γραφήματος ACF	Ναι	Όχι
Υποστήριξη ARIMA models	Ναι	Ναι
Υποστήριξη ARMA models	Ναι	Ναι
Υποστήριξη regression models	Ναι	Όχι
Υποστήριξη Random Walk models	Ναι	Όχι
Αναπαράσταση δεδομένων	Οι χρονοσειρές ανήκουν στην κλάση TimeSeries, και τα σημεία τους είναι πίνακας από double[].	Οι χρονοσειρές αναπαρίστανται με πίνακες double[].
Λειτουργικότητα	Κατασκευή μοντέλων Linear Regression, Mean, Random Walk, ARMA, ARIMA μοντέλων. Γραφική αναπαράσταση χρονοσειράς, ACF και scatter plots. Παραγωγή προβλέψεων για τα παραπάνω μοντέλα. Υπολογισμός AIC για ARIMA.	Κατασκευή μοντέλων ARIMA Παραγωγή προβλέψεων για μοντέλα ARIMA

Πίνακας 3.1 *Κριτήρια σύγκρισης υλοποιήσεων για αναπαράσταση και διαχείριση χρονοσειρών.*

Αφού έγινε η κατάλληλη επεξεργασία, από τη βιβλιοθήκη java-timeseries αξιοποιήθηκαν τα τμήματα που φαίνονται στο *Σχήμα 3.4*. Στο σχήμα, φαίνονται επίσης τμήματα μιας

άλλης βιβλιοθήκης, της stl-decomp-4j (https://github.com/ServiceNow/stl-decomp-4j), τα οποία αξιοποιήθηκαν για την εύρεση της εποχικότητας σε μια χρονοσειρά.



Σχήμα 3.4 *Τμήματα των βιβλιοθηκών java-timeseries και stl-decomp-4j που χρησιμοποιήθηκαν.*

Ακολουθεί η ανάλυση του Σχήματος 3.4.

- Η κλάση **TimeSeries** αποτελεί τη δομή μιας χρονοσειράς και τις διάφορες λειτουργίες που μπορούν να εκτελεστούν σε αυτή.

- Οι κλάσεις **Arima**, **ArimaModel** και **ArimaOrder**, αφορούν τη δομή ενός μοντέλου και τις λειτουργίες που μπορούν να εκτελεστούν σε αυτό.
- Η κλάση **Forecast** περιλαμβάνει τις μεθόδους που μπορούν να χρησιμοποιηθούν σε μια πρόβλεψη.
- Η κλάση **TestData** περιλαμβάνει παραδείγματα χρονοσειρών.
- Οι κλάσεις **SeasonalTrendLoess**, **Builder** και **Decomposition** περιλαμβάνουν τη λειτουργικότητα εξαγωγής της εποχικότητας μιας χρονοσειράς.

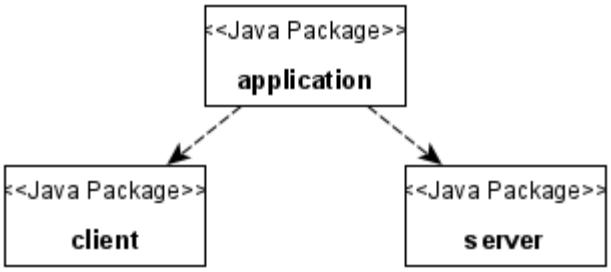
3.3 Σχεδίαση εφαρμογής

Το λογισμικό, που βρίσκεται στη σελίδα https://github.com/othonasgkavardinas/java_timeseries_arima σχεδιάστηκε με βάση την αρχιτεκτονική πακέτων του *Σχήματος 3.5* και κλάσεων του *Σχήματος 3.6*, όπου ο client, διαχειρίζεται την κλάση MainController, και επίσης την κλάση State. Ο server, είναι υπεύθυνος μόνο για την κλάση Server, ακολουθώντας έτσι μια stateless προσέγγιση, που δεν προσδίδει φόρτο σε αυτόν. Σύμφωνα με την αρχιτεκτονική αυτή:

- Η κλάσεις **Main** και **MainController**, αποτελούν τα κύρια μέρη της γραφικής διεπαφής χρήστη (GUI). Η πρώτη εκκινεί την εφαρμογή και η δεύτερη διαθέτει τη λειτουργικότητα των διαφόρων listeners της εφαρμογής. Η κλάση MainController είναι αρκετά σύνθετη και γι' αυτό και δεν αναλύεται στο σχήμα.
- Η κλάση **HelpText** κρατά σε κάποιες δομές βοηθητικό κείμενο που αναγράφεται εάν ο χρήστης πιέσει την επιλογή Help->
- Η κλάση **Details** κρατά σε κάποιες δομές βοηθητικό κείμενο που αναγράφεται στην αριστερή μπάρα της εφαρμογής, και εναλλάσσεται καθώς ο χρήστης αλληλοεπιδρά με αυτή.
- Η κλάση **ForecastFile** διαχειρίζεται τη δημιουργία ενός αρχείου, και αποθήκευση σε αυτό μιας πρόβλεψης.
- Η κλάση **Visualizer** παράγει κάθε γραφική αναπαράσταση που ζητείται από την εφαρμογή.
- Οι κλάσεις **RecordInput**, **RecordInputFactory**, **ColumnInput** και **UCRInput**, αφορούν τη δημιουργία φόρμας εισαγωγής

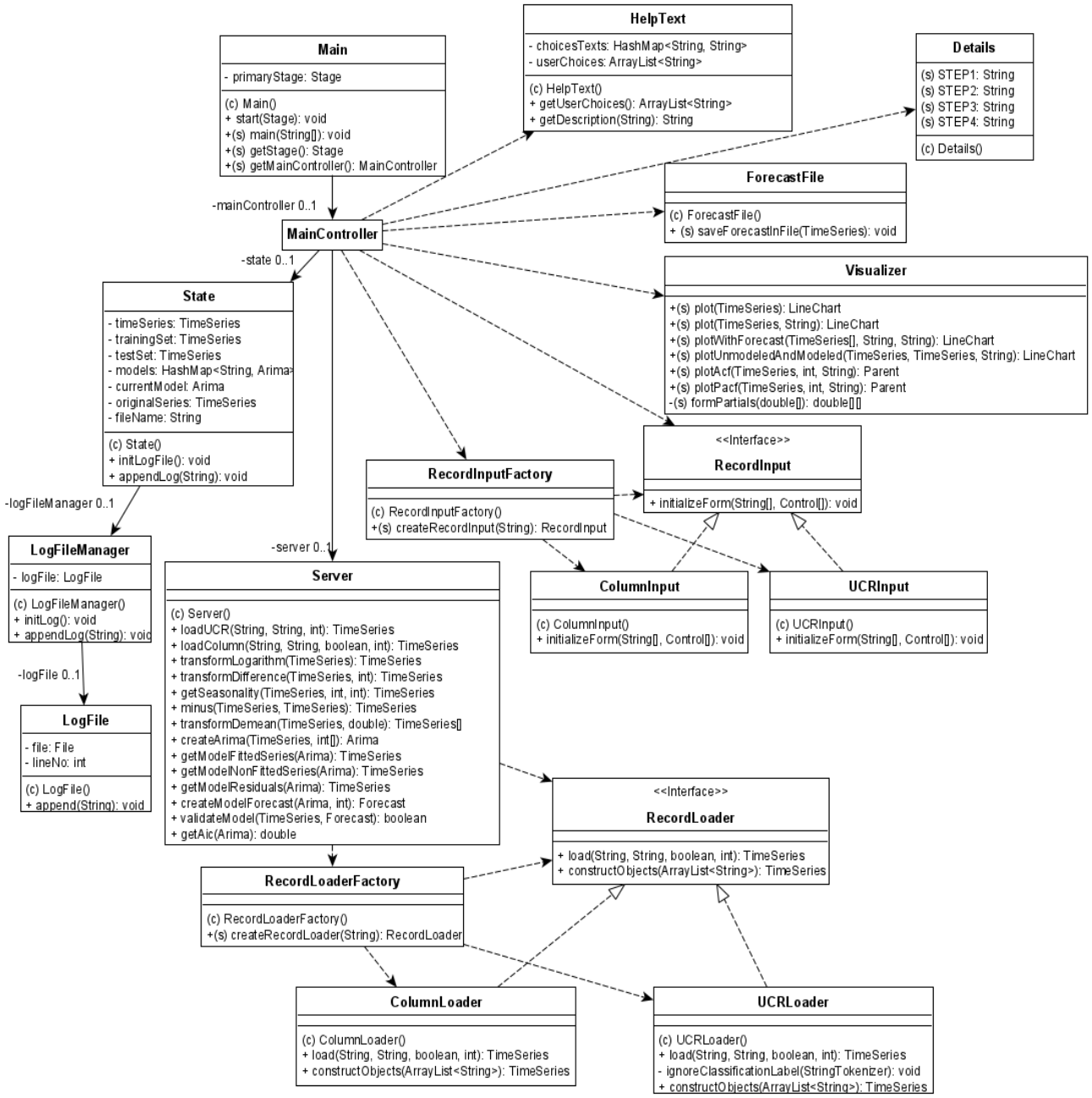
στοιχείων για την ανάγνωση κάποιου αρχείου εισόδου από το χρήστη.

- Η κλάση **State** περιλαμβάνει τα δεδομένα που απαιτούνται για τη σωστή λειτουργία της εφαρμογής. Για παράδειγμα, αυτή είναι που διαθέτει τη χρονοσειρά που φορτώνεται.



Σχήμα 3.5 *Διάγραμμα πακέτων του λογισμικού.*

- Οι κλάσεις **LogFileManager** και **LogFile** αφορούν την καταγραφή του ιστορικού χρήσης της εφαρμογής σε αρχείο. Η κλάση LogFile, είναι η δομή, και η κλάση LogFileManager είναι υπεύθυνη για τη διαχείριση του αρχείου.
- Η κλάση **Server** περιέχει της λειτουργίες της εφαρμογής. Για παράδειγμα σε αυτήν γίνεται η επεξεργασία μιας χρονοσειράς, ή ενός μοντέλου.
- Οι κλάσεις **RecordLoader**, **RecordLoaderFactory**, **ColumnLoader** και **UCRLoader**, διαχειρίζονται το άνοιγμα ενός αρχείου εισόδου.



Σχήμα 3.6 *Διάγραμμα κλάσεων του λογισμικού.*

3.4 Έλεγχος λογισμικού

Στην παρούσα ενότητα παρουσιάζονται οι έλεγχοι που έγιναν στην εφαρμογή. Οι έλεγχοι εστιάζονται στην κλάση Server.java, η οποία αποτελεί έναν stateless server με λειτουργίες που μπορούν να κληθούν, από κάποιον client, ο οποίος θεωρείται πως διαθέτει και ένα αντικείμενο της κλάσης State.java. Για τον έλεγχο της δημιουργούμενης

εφαρμογής σχηματίστηκαν Junit tests. Στον *Πίνακα 3.2* αναγράφεται η αντιστοιχία των διαφόρων tests με τις λειτουργίες, οι οποίες ελέγχονται ως προς την ορθότητα.

Μέθοδος ελέγχου	**Λειτουργία**	**Περιγραφή**
loadUCRTest()	Ανάγνωση αρχείου UCR	Σύγκριση τιμών που φορτώνονται από το αρχείο με τις πραγματικές τιμές του αρχείου.
loadColumnTest()	Ανάγνωση αρχείου Column	Σύγκριση τιμών που φορτώνονται από το αρχείο με τις πραγματικές τιμές του αρχείου.
transformLogarithmTest()	Εφαρμογή λογαρίθμου σε χρονοσειρά	Σύγκριση των αποτελεσμάτων της συνάρτησης λογαρίθμου του server, με τα αποτελέσματα της συνάρτησης λογαρίθμου της ίδιας ακολουθίας στην R.
transformDifferenceTest()	Διαφορές χρονοσειράς	Σύγκριση των αποτελεσμάτων της συνάρτησης πρώτων και δεύτερων διαφορών του server, με τις αντίστοιχες διαφορές στην R.
minusTest()	Αφαίρεση χρονοσειρών στοιχείο προς στοιχείο	Αφαίρεση χρονοσειράς από χρονοσειρά και σύγκριση με αναμενόμενο αποτέλεσμα.

transformDemeanTest()	Αφαίρεση μέσου χρονοσειράς	Σύγκριση των αποτελεσμάτων της συνάρτησης αφαίρεσης μέσου του server, με την αντίστοιχη στην R.
convertToArrayTest()	Μετατροπή TimeSeries object σε double[] object.	Γίνεται η μετατροπή μιας χρονοσειράς σε πίνακα και ελέγχονται οι τιμές του πίνακα με τις τιμές της χρονοσειράς αν είναι οι ίδιες.
sizeTest()	Μέγεθος χρονοσειράς	Ελέγχεται η μέθοδος του server, αν επιστρέφει το πραγματικό μήκος μιας χρονοσειράς.
splitTimeSeriesTest()	Διαχωρισμός χρονοσειράς σε σύνολα εκπαίδευσης και ελέγχου	Ελέγχεται αν γίνεται σωστά ο διαχωρισμός, συγκρίνοντας με αναμενόμενα αποτελέσματα.
createArimaTest()	Δημιουργία ARIMA μοντέλου	Δημιουργείται ένα ARIMA μοντέλο και ελέγχεται αν οι τάξεις (orders) του μοντέλου είναι οι δοθείσες.
getModelFittedSeriesTest()	Εξαγωγή αντιστοιχισμένων δεδομένων μοντέλου	Σύγκριση των αντιστοιχισμένων δεδομένων ενός μοντέλου ARIMA της εφαρμογής, με τα αντιστοιχισμένα δεδομένα του ίδιου μοντέλου στην R.

getModelNonFittedSeriesTest()	Εξαγωγή μη αντιστοιχισμένων δεδομένων μοντέλου	Σύγκριση των μη αντιστοιχισμένων δεδομένων ενός μοντέλου ARIMA, με τα πηγαία δεδομένα.
getModelResidualsTest()	Εξαγωγή υπολειπόμενης ποσότητας από μοντέλο	Σύγκριση της υπολειπόμενης ποσότητας μοντέλου ARIMA της εφαρμογής, με την υπολειπόμενη ποσότητα του ίδιου μοντέλου στην R.
validateModelTest()	Έλεγχος ποιότητας πρόβλεψης μοντέλου	Εφαρμογή ελέγχου σε μοντέλο που αναμένεται να ταιριάζει με τα δεδομένα.
getAicTest()	Παραγωγή μετρικής AIC για μοντέλο	Σύγκριση του AIC ενός μοντέλου ARIMA της εφαρμογής, με το αντίστοιχο AIC του ίδιου μοντέλου στην R.
Σημείωση: Στους παραπάνω ελέγχους, όπου χρειάστηκε σύγκριση με τα αποτελέσματα της R, γίνονται κατάλληλες αλλαγές έτσι ώστε να συμβαδίζουν τα αποτελέσματα ως προς την ακρίβεια.		

Πίνακας 3.2 *Πίνακας ελέγχων για τις διάφορες λειτουργικότητες της εφαρμογής.*

Όλοι οι έλεγχοι πραγματοποιήθηκαν με επιτυχία.

3.5 Οδηγίες εγκατάστασης

Η παρούσα εφαρμογή αναπτύχθηκε στην προγραμματιστική γλώσσα Java 13.0.2.

Χρησιμοποιήθηκαν για την υλοποίηση τα εξής εργαλεία:

- Eclipse IDE for Java Version 2019-12
- JavaFX SDK 13.0.2
- JavaFX Scene Builder 11.0.0

Για να λειτουργήσει η συγκεκριμένη εφαρμογή απαιτείται η εγκατάσταση των εργαλείων Eclipce, JavaFX και της γλώσσας Java.

Αφού εγκατασταθούν τα παραπάνω, πρέπει να γίνει προσθήκη του JavaFX στο Eclipse και στο project.

3.5.1 Προσθήκη του εργαλείου JavaFX

Για την προσθήκη του JavaFX στο Eclipse και στο project ακολουθείται η εξής σειρά βημάτων:

- Στο MenuBar του Eclipse, επιλέγετε Help->Eclipse Marketplace…, και εκεί γίνεται αναζήτηση και εγκατάσταση για το πρόσθετο e(fx)clipse. (Παρούσα έκδοση 3.6.0)
- Στο MenuBar του Eclipse, επιλέγετε Window->Preferences και εκεί επιλέγετε από το αριστερό Menu JavaFX. Θα εμφανιστεί στα δεξιά σας η επιλογή JavaFX 11+ SDK, όπου επιλέγετε το φάκελο όπου έχει εγκατασταθεί το JavaFX. (Για παράδειγμα, στο σύστημα ανάπτυξης της εφαρμογής επιλέχθηκε το μονοπάτι 'C:\Program Files\JFX\javafx-sdk-13.0.1'.)
- Στο Eclipse, επιλέγετε να ανοίξετε την εφαρμογή ανοίγοντας την κλάση application.Main και επιλέγοντας στο MenuBar, Run->Run. Η εκτέλεση του προγράμματος αναμένεται να αποτύχει. Ύστερα επιλέγετε από το MenuBar, Run->Run Configurations, και πρέπει να δοθούν κάποιες εντολές εκτέλεσης. Για να δοθούν οι εντολές, επιλέγετε αριστερά κάτω από την καρτέλα Java Applications την προηγούμενη μη-επιτυχημένη εκτέλεση και πάνω δεξιά, επιλέγετε την καρτέλα Arguments. Στην καρτέλα αυτή, στο δεύτερο textbox που καλείται VM arguments γράφετε τις παραμέτρους:
  "--module-path "C:\Program Files\JFX\javafx-sdk-13.0.1\lib" --add-modules=javafx.controls,javafx.fxml",
  Όπου πρέπει να αντικαταστήσετε το module path με το φάκελο lib της εγκατάστασης του JavaFX σας.

3.6 Υποστηριζόμενα αρχεία

Η συγκεκριμένη εφαρμογή μπορεί να κάνει ανάγνωση χρονοσειρών από δύο τύπους αρχείων:

- UCR File: Ένα αρχείο που διαθέτει μια χρονοσειρά ως γραμμή, η πρώτη τιμή της οποίας είναι μια ετικέτα κατηγοριοποίησης, και έτσι παραλείπεται. (Η δομή των αρχείων προκύπτει από τις χρονοσειρές του UCR Archive 2018)
  Η ανάγνωση τέτοιων αρχείων απαιτεί από το χρήστη να εισάγει το μονοπάτι αρχείου, ένα χαρακτήρα ή φράση που διαχωρίζει τις διάφορες τιμές μιας χρονοσειράς, και η θέση της γραμμής που περιέχει τα δεδομένα της χρονοσειράς.
- Column File: Ένα αρχείο που διαθέτει μια χρονοσειρά ως στήλη.
  Η ανάγνωση τέτοιων αρχείων απαιτεί από το χρήστη να εισάγει το μονοπάτι αρχείου, ένα χαρακτήρα ή φράση που διαχωρίζει τις διάφορες τιμές χρονοσειράς και να καθοριστεί η θέση της στήλης στην οποία βρίσκεται η χρονοσειρά.

3.7 Επεκτασιμότητα του λογισμικού

Για το παρόν λογισμικό, μελλοντικά, θα ήταν θεμιτό να εισαχθούν τα εξής τμήματα λειτουργικότητας:

- Υλοποίηση datetime-axis για τη βιβλιοθήκη java-fx, έτσι ώστε, στις γραφικές αναπαραστάσεις να αναγράφονται οι χρονοσφραγίδες των χρονοσειρών.
- Σχηματισμός κατάλληλου Junit test, για έλεγχο ορθότητας των αποτελεσμάτων των μεθόδων getSeasonality(), getAutoCorrelationUpToLag() και formPartialsForPacf().

Ακόμη, σημειώνεται πως στον κώδικα, περιλαμβάνονται οι εξής hard-coded λειτουργικότητες ενδέχεται να χρειασθούν συντήρηση:

- Η εύρεση της εποχικότητας μιας χρονοσειράς της μεθόδου getSeasonality(), της κλάσης Server γίνεται με χρήση τμήματος της βιβλιοθήκης servicenow.
- Η μέθοδος formPartialsForPacf() της κλάσης Server, η οποία χρησιμοποιεί την τεχνική Durban-Leverson, και εισάχθηκε από την πηγή https://github.com/XuMeng-NTU/CSC489/blob/master/Weka/src/weka/filters/timeseries/PACF.java.

Κεφάλαιο 4. Παράδειγμα χρήσης

Στο κεφάλαιο αυτό περιγράφεται ο τρόπος λειτουργίας της εφαρμογής που δημιουργήθηκε. Μέσα από ένα παράδειγμα χρήσης, προβάλλονται οι διάφορες επιλογές που παρέχονται στο χρήστη και τα αποτελέσματα τους. Το παράδειγμα του κεφαλαίου προέκυψε από πειραματισμό και είναι κατάλληλο για την επίδειξη των δυνατοτήτων της εφαρμογής. Ο σκοπός του είναι να επαληθευτεί η λειτουργικότητα του λογισμικού και να τονισθεί η ευχρηστία που παρέχει στους ενδιαφερόμενους χρήστες. Το παράδειγμα εφαρμόστηκε σε φορητό υπολογιστή (laptop) απλής χρήσης, μέτριων δυνατοτήτων. Το υλικό του περιλαμβάνει τα εξής:

- Επεξεργαστής: Intel i5-4200U
- Μνήμη: 8.00 GB
- Λειτουργικό σύστημα: Windows 10 Home x64

4.1 Δεδομένα εισόδου

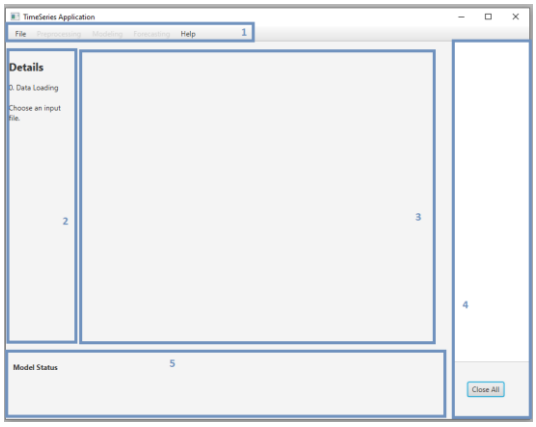
Τα δεδομένα που εισάγονται στο ακόλουθο παράδειγμα ανήκουν στο UCR Archive 2018, και συγκεκριμένα, βρίσκονται στο αρχείο Car_TEST.tsv. Ένα αντίγραφο αυτού του αρχείου βρίσκεται στο φάκελο \resources, εντός της εφαρμογής. Πρόκειται για μια χρονοσειρά που πληροί τις προϋποθέσεις για το σχηματισμό ενός απλού, κατανοητού παραδείγματος.

4.2 Παρουσίαση παραδείγματος

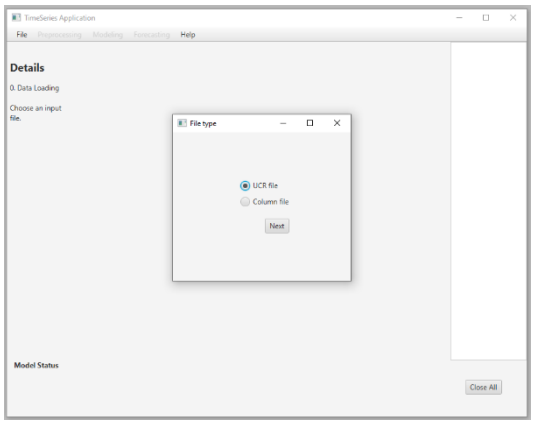
Αρχικά, ο χρήστης εκκινεί την εφαρμογή. Το πρώτο παράθυρο που εμφανίζεται στο χρήστη, χωρίζεται σε 5 βασικά μέρη.

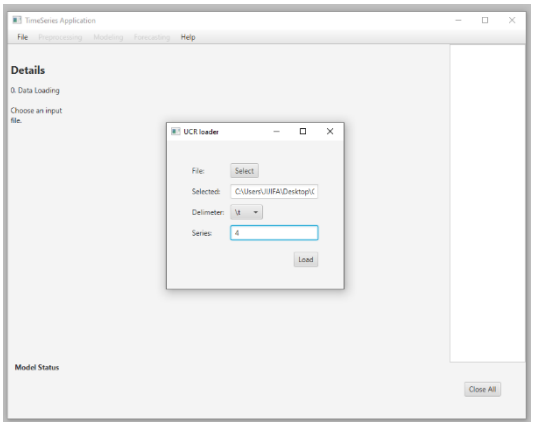
1) Βασικό μενού λειτουργιών του χρήστη.

2) Στήλη ακολουθίας βημάτων και ειδικών οδηγιών.
3) Πλαίσιο γραφήματος της επιλεχθείσας χρονοσειράς.
4) Στήλη διαχείρισης ανοικτών γραφημάτων.
5) Πλαίσιο εμφάνισης στοιχείων του επιλεχθέντος μοντέλου.



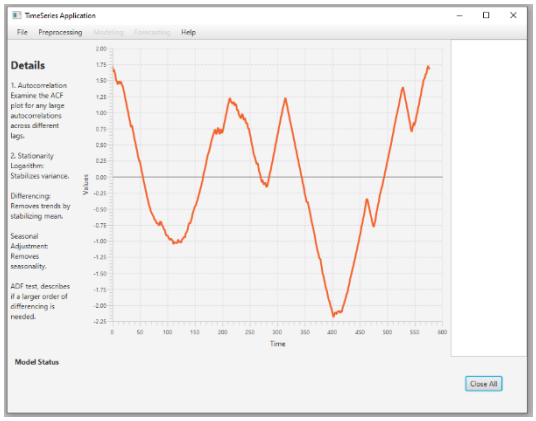
Αρχικά, πρέπει να επιλεχθεί κάποιο υποστηριζόμενο αρχείο. Από το βασικό μενού, επιλέγεται File->Load Data.



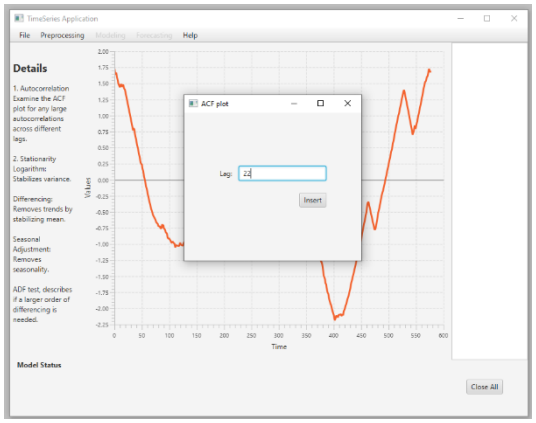


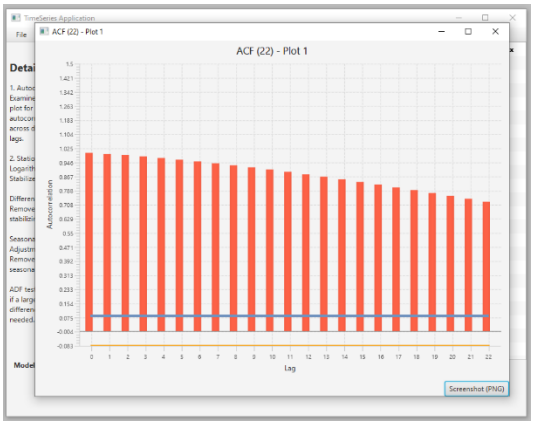
Αν η φόρτωση του αρχείου ήταν επιτυχής, στο πλαίσιο γραφήματος θα εμφανιστεί η χρονοσειρά που εισάχθηκε.

Ο χρήστης έχει τη δυνατότητα να επανέλθει στα πηγαία δεδομένα, οποιαδήποτε στιγμή, στη διάρκεια εκτέλεσης της εφαρμογής, και αφού τα δεδομένα έχουν φορτωθεί επιτυχώς, επιλέγοντας File->Load original.

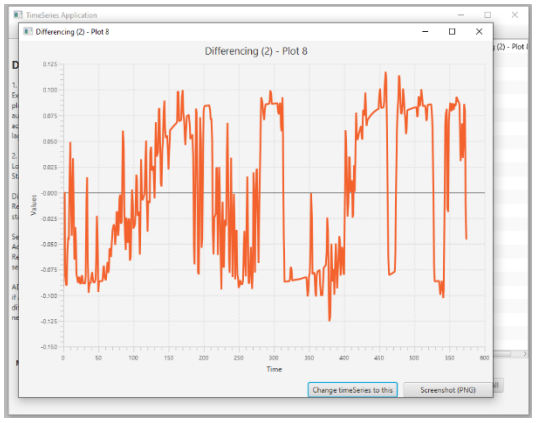


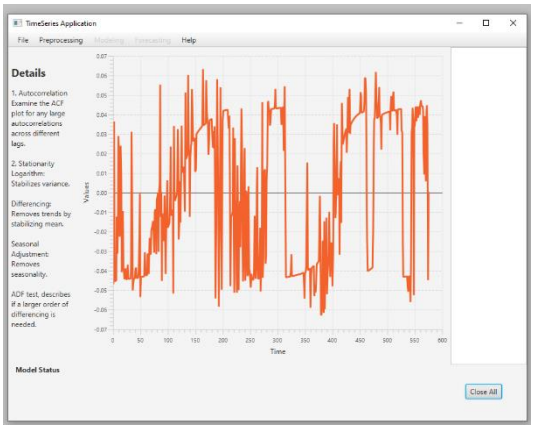
Αφού έχει φορτωθεί η χρονοσειρά του χρήστη, η εφαρμογή, επιτρέπει σε αυτόν, να τροποποιήσει τα δεδομένα της. Για τροποποίηση, στο βασικό μενού, βρίσκεται η επιλογή Preprocessing. Κατά τη φάση της τροποποίησης, πρέπει αρχικά να ελεγχθεί εάν υπάρχουν αυτοσυσχετίσεις στο διάγραμμα ACF, με την επιλογή Preprocessing->Plot ACF. Με την επιλογή, εμφανίζεται πλαίσιο εισαγωγής lag από το χρήστη για την εμφάνιση του κατάλληλου γραφήματος.



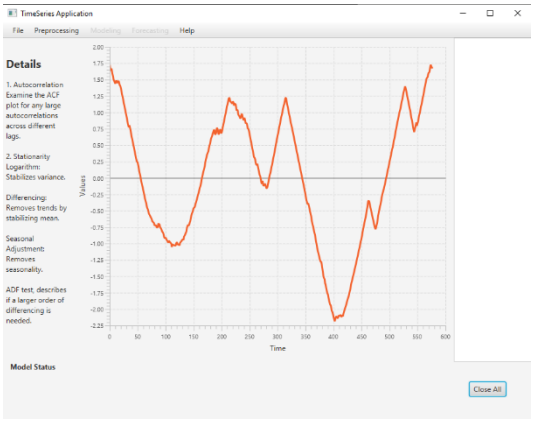


Στο διάγραμμα είναι εμφανές πως υπάρχουν αυτοσυσχετίσεις. Ο χρήστης μπορεί να συνεχίσει την επεξεργασία με εφαρμογή διαφορών δευτέρου βαθμού στη χρονοσειρά, επιλέγοντας Preprocessing->Differencing, και εισάγοντας τον αριθμό 2, που υποδηλώνει το βαθμό διαφορών. Επίσης, αφού τον ικανοποιεί το αποτέλεσμα, μπορεί να πατήσει το κουμπί 'Change TimeSeries to this' κάτω δεξιά, για να κρατήσει την αλλαγή που έγινε στη χρονοσειρά. Εάν δεν πατήσει το κουμπί, δεν γίνεται καμία αλλαγή στην αρχική χρονοσειρά.

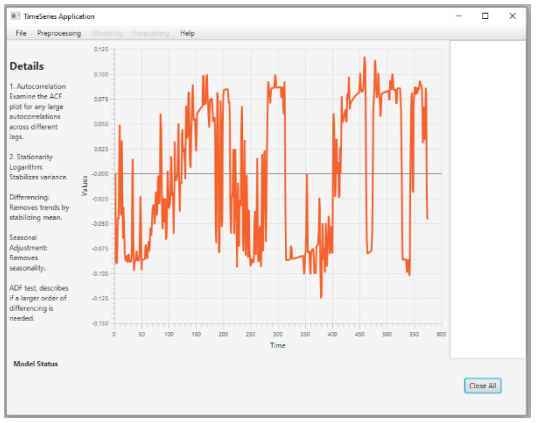




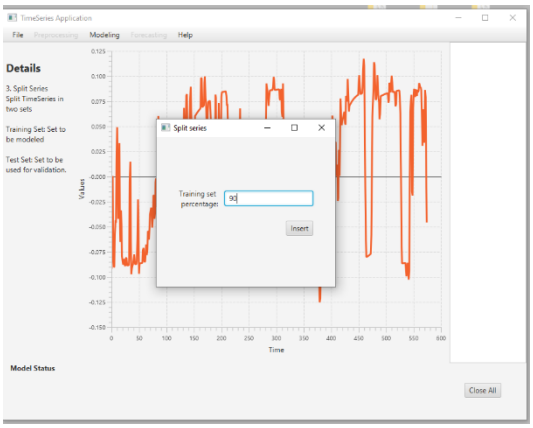
Σε περίπτωση που ο χρήστης αποφασίσει να αναιρέσει τις αλλαγές που έχει κάνει στα δεδομένα που εισήγαγε, υπάρχει η επιλογή File->Load Original, με την οποία η χρονοσειρά επιστρέφει στη μορφή που είχε όταν διαβάσθηκε από το αρχείο του χρήστη.



Στο παρόν παράδειγμα, οι διαφορές δεύτερης τάξης είναι θεμιτές, οπότε έστω ότι ο χρήστης ξανά κάνει τις αλλαγές που έκανε στα πρώτα βήματα. Ακόμη, υποστηρίζονται και άλλοι μετασχηματισμοί, οι οποίοι εφαρμόζονται με παρόμοιο τρόπο. Αν ο χρήστης είναι αρκετά σίγουρος πως η χρονοσειρά είναι εργοδική, επιλέγει Preprocessing->Done, έτσι ώστε να προχωρήσει στη φάση της μοντελοποίησης.

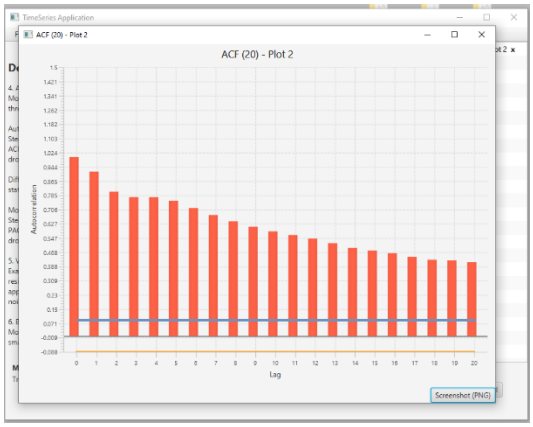


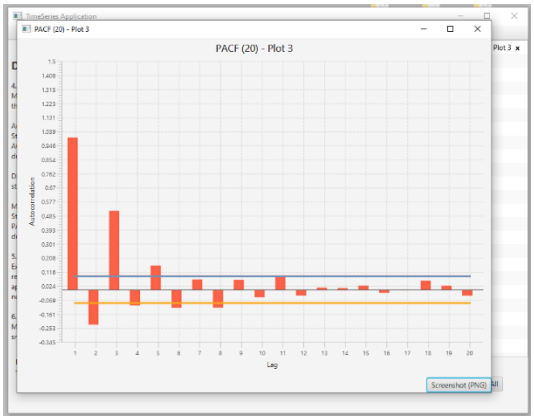
Κατά τη φάση της μοντελοποίησης ενεργοποιείται στο βασικό μενού, η επιλογή Modeling. Ως πρώτο βήμα της μοντελοποίησης πρέπει ο χρήστης να διαχωρίσει τα δεδομένα της χρονοσειράς που διαθέτει σε σύνολο εκπαίδευσης και σύνολο επιβεβαίωσης. Για τον διαχωρισμό, επιλέγει Modeling->Split series, και καθορίζει το ποσοστό των δεδομένων που ανήκει στο σύνολο εκπαίδευσης.



Αφού διαχωριστούν τα δεδομένα, στο πλαίσιο εμφάνισης στοιχείων μοντέλου, αναγράφεται το επιλεχθέν ποσοστό. Αν σε κάποια στιγμή ο χρήστης θέλει να αλλάξει αυτή του την επιλογή, μπορεί να επιλέξει ξανά Modeling->Split series, όμως τότε θα αναγκαστεί να διαγράψει όλα τα μοντέλα που έχει δημιουργήσει ως εκείνη τη στιγμή.

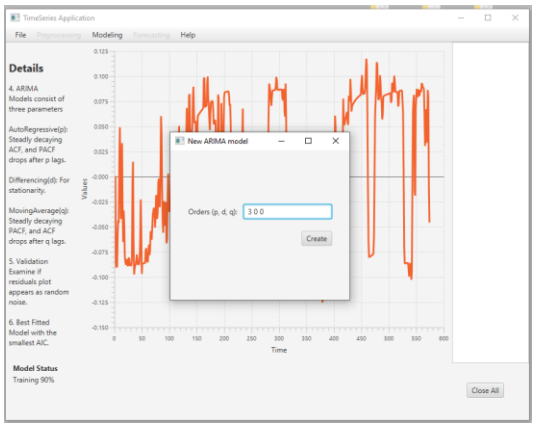
Στη συνέχεια, ο χρήστης παράγει και επεξεργάζεται μοντέλα, που αφορούν το σύνολο εκπαίδευσης. Για να καθοριστεί το πλήθος των όρων p και q ενός ARIMA(p, d, q), ο χρήστης μπορεί να προβάλλει τα διαγράμματα ACF και PACF των δεδομένων.

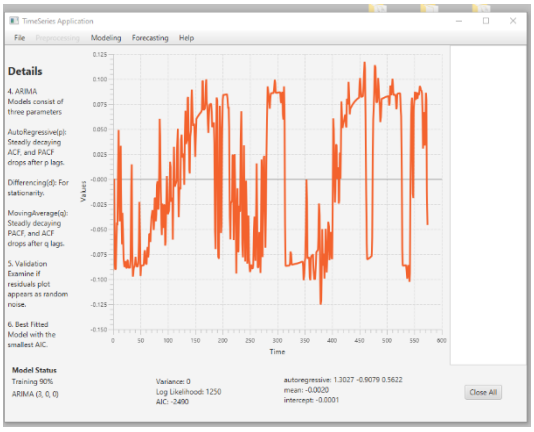




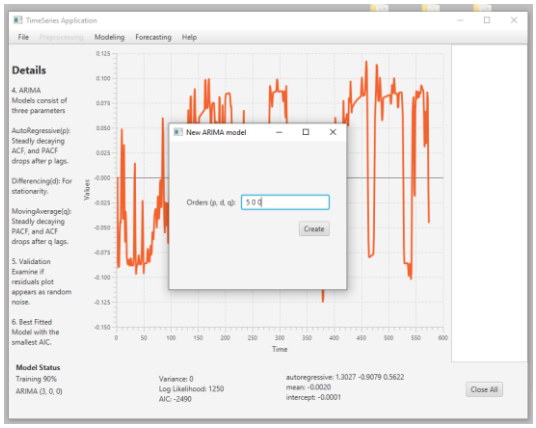
Στα παραπάνω διαγράμματα, είναι φανερή η ανάγκη προσθήκης 8, 6 ή λιγότερων p όρων. Ο χρήστης μπορεί να δημιουργήσει ένα μοντέλο, με βάση τα στοιχεία που έχει αντλήσει επιλέγοντας ARIMA->Create Arima.

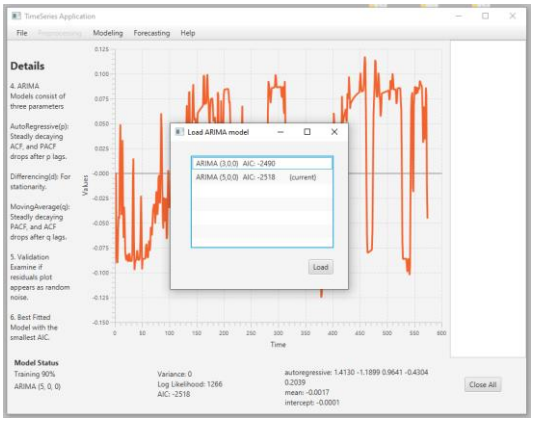
Προσοχή! Η εκτίμηση από τη γραφική αναπαράσταση δεν είναι απόλυτη, και η πρόβλεψη με μοντέλα ARIMA απαιτεί συνεχείς επανεκτιμήσεις του μοντέλου μέχρις ότου, να επιβεβαιωθεί πως παράγει καλές προβλέψεις. Έστω ότι επιλέγει ο χρήστης τους (3, 0, 0) όρους, μιας και η προσθήκη πολλών όρων συχνά επιφέρει υπερ-ταίριασμα σε μοντέλα. Στο πλαίσιο εμφάνισης στοιχείων μοντέλου αναγράφονται οι πληροφορίες του μοντέλου.





Για επίδειξη κάποιων λειτουργιών της εφαρμογής, έστω ότι δημιουργεί επίσης, το μοντέλο (5, 0, 0) με τον ίδιο τρόπο. Με τη δημιουργία του (5, 0, 0) μοντέλου, γίνεται αυτόματα και η επιλογή του ως το τρέχον μοντέλο και αντίστοιχα μεταβάλλονται και οι πληροφορίες του πλαισίου εμφάνισης στοιχείων μοντέλου. Ο χρήστης μπορεί να φορτώσει (load) κάποιο από τα μοντέλα που έχει δημιουργήσει με την επιλογή Modeling->Load ARIMA.

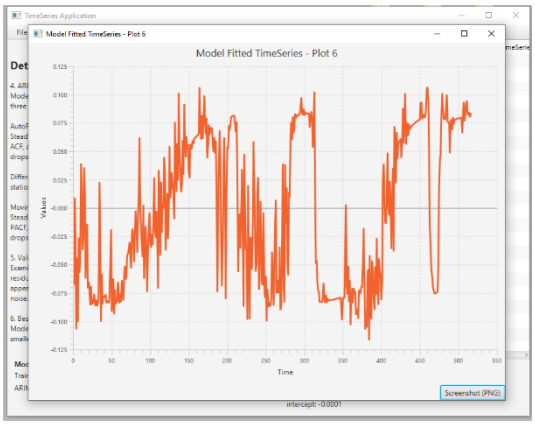




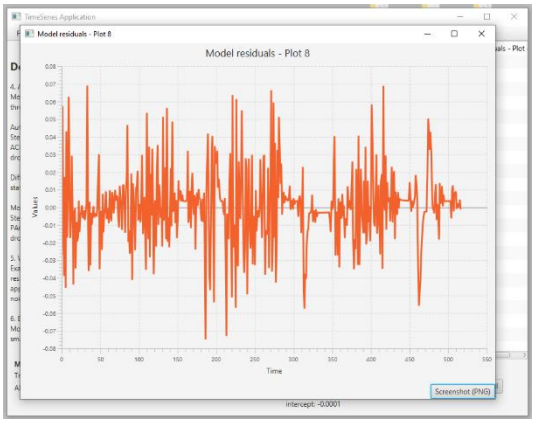
Αντίστοιχα, μπορεί να διαγράψει κάποιο μοντέλο με την επιλογή Modeling->Delete Arima. Για τη φόρτωση και διαγραφή μοντέλου, αναγράφεται σε κάθε μια επιλογή, ο δείκτης AIC και ακόμη, εάν είναι το μοντέλο είναι το τρέχον. Συνήθως, ο χαμηλότερος δείκτης AIC υποδηλώνει καλύτερο ταίριασμα. Ακόμη, τονίζεται πως δεν είναι εφικτή η

διαγραφή του τρέχοντος μοντέλου. Στο παράδειγμά, ο χρήστης κρατά το μοντέλο (5, 0, 0) καθώς έχει και το χαμηλότερο δείκτη AIC.

Στο υπό-μενού Modeling, βρίσκεται επίσης η επιλογή 'Plot series', που εμφανίζει το σύνολο εκπαίδευσης, και η επιλογή 'Plot with original', που εμφανίζει το σύνολο εκπαίδευσης πριν και μετά το ταίριασμα με το μοντέλο στο ίδιο γράφημα.



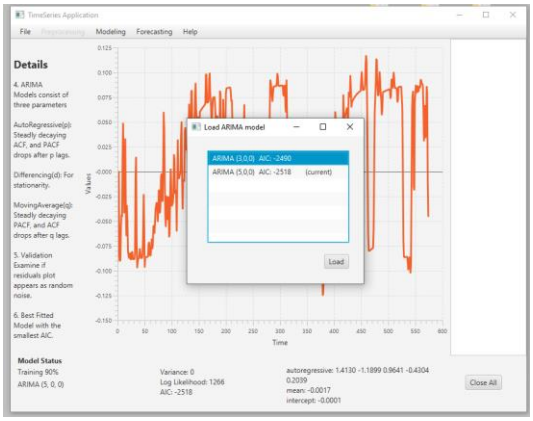
Ένας καλός έλεγχος, αν το μοντέλο έχει ταιριάξει καλά με τα δεδομένα του συνόλου εκπαίδευσης, είναι η εξέταση της υπολειπόμενης ποσότητας του μοντέλου. Με την επιλογή Modeling->Plot residuals, εμφανίζεται στο χρήστη η αναπαράσταση της υπολειπόμενης ποσότητας, και έτσι, δίνεται σε αυτόν η δυνατότητα να ελέγξει αν έχει τη μορφή τυχαίου θορύβου. Στο παρακάτω σχήμα το γράφημα μοιάζει με τυχαίο θόρυβο. Επίσης, μπορεί να εξετασθεί αν το γράφημα ACF της υπολειπόμενης ποσότητας, περιέχει αυτοσυσχετίσεις, με την επιλογή Modeling->Plot ACF of residuals. Αν αυτή διαθέτει αυτοσυσχετίσεις, τότε, συνήθως το μοντέλο δεν έχει ταιριάξει καλά με τα δεδομένα.

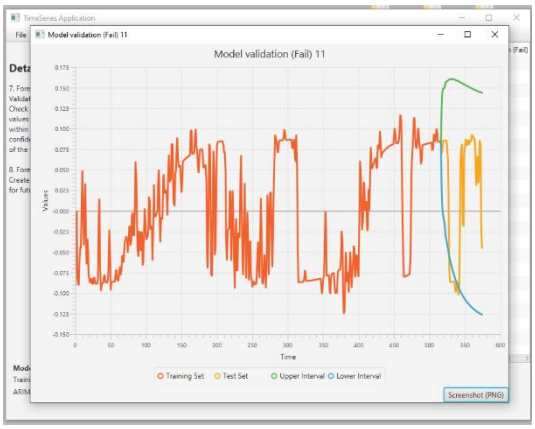


Στο παράδειγμα, φαίνεται πως υπάρχουν αυτοσυσχετίσεις. Παρ' όλα αυτά, έστω ότι ο χρήστης προχωρά στη φάση της παραγωγής προβλέψεων. Στη φάση αυτή, έχει δύο επιλογές. Η πρώτη, είναι να δοκιμάσει αν το μοντέλο μπορεί να προβλέψει το σύνολο

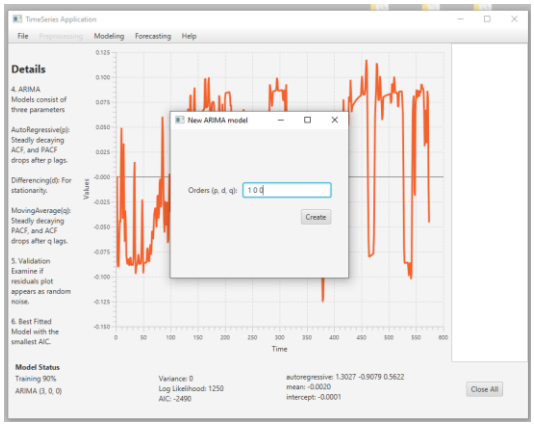
ελέγχου που έχει εξαγάγει από τα δεδομένα. Η δεύτερη επιλογή, είναι να επιλέξει, πόσες τιμές στο μέλλον θα ήθελε να προβλέψει και να παραγάγει μια νέα πρόβλεψη. Ο χρήστης θα μπορούσε να ελέγξει την ικανότητα πρόβλεψης επιλέγοντας Forecasting->Validation. Στον τίτλο του γραφήματος που προκύπτει, αναγράφεται η ετικέτα (Pass), εάν οι τιμές του συνόλου επιβεβαίωσης βρίσκονται στο 95% των ορίων της πρόβλεψης που έγινε, και αλλιώς (Fail).

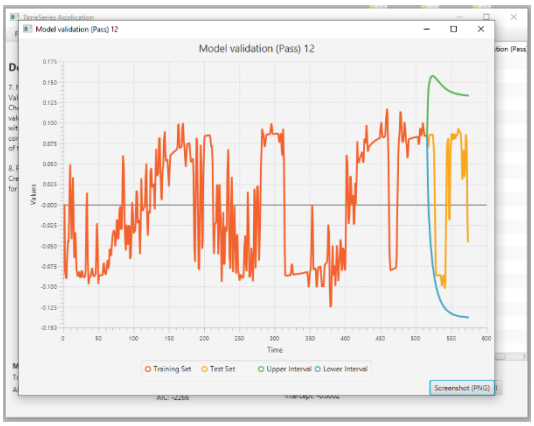
Σύμφωνα με το σχήμα και την ετικέτα (Fail), τα αποτελέσματα του ελέγχου είναι αρνητικά. Έστω ότι ο χρήστης φορτώνει το μοντέλο (3, 0, 0), και ελέγχει την ικανότητα πρόβλεψής του.





Τα αποτελέσματα είναι επίσης αρνητικά, οπότε, θα ήταν σωστό να συνεχίσει με τη δημιουργία ενός νέου μοντέλου με λιγότερες παραμέτρους, γιατί ίσως έχει υπερ-ταιριάξει το μοντέλο με τα δεδομένα του χρήστη. Έστω ότι ο χρήστης αποφασίζει να δοκιμάσει την ικανότητα πρόβλεψης του μοντέλου (1, 0, 0).





Τα αποτελέσματα είναι θετικά. Έχει εντοπίσει λοιπόν, ένα μοντέλο που μπορεί να προβλέψει τα δεδομένα εισόδου. Επόμενο και τελευταίο βήμα της διαδικασίας, αποτελεί η παραγωγή νέων προβλέψεων. Με την επιλογή Forecasting->New Forecast, ο χρήστης πληκτρολογεί τον αριθμό των μελλοντικών τιμών που θέλει να προβλέψει. Στο παράδειγμα ζητά 5 τιμές.

Η διαδικασία της πρόβλεψης ολοκληρώθηκε επιτυχώς και έχει δημιουργηθεί ένα αρχείο forecasts/forecast##########, που περιλαμβάνει τις τιμές του αποτελέσματος. Στο όνομα του αρχείου, όπου '##########' είναι ένας μοναδικός αριθμός που προκύπτει από την τρέχουσα ημερομηνία και ώρα.

4.2.1 Επιπλέον δυνατότητες και παρατηρήσεις

4.2.1.1 Παράθυρα γραφημάτων

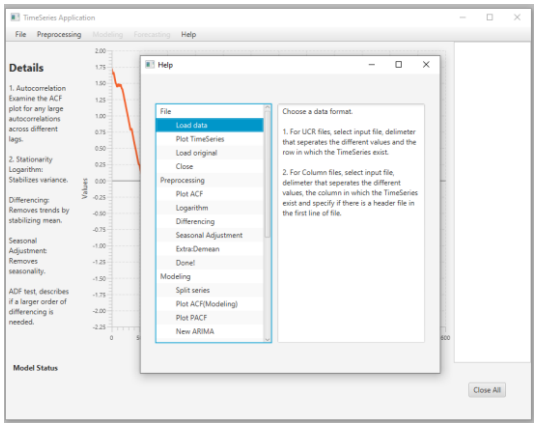
- Κάθε γράφημα της εφαρμογής ανοίγει σε ένα νέο παράθυρο, και διαθέτει την επιλογή αποθήκευσής του σε PNG, με το πάτημα του κουμπιού 'Screenshot (PNG)'

κάτω και δεξιά. Επίσης, όπου χρειάζονται τιμές εισόδου από το χρήστη, ζητούνται από ένα μικρό παράθυρο εισόδου, όπως φαίνεται και στα παραπάνω παραδείγματα εικόνων.

- Σε κάθε γράφημα δίδεται ένας ακολουθιακός αριθμός.
- Η στήλη διαχείρισης ανοιχτών γραφημάτων εμφανίζει ονομαστικά όλα τα ανοικτά παράθυρα που αφορούν γραφήματα. Ο χρήστης μπορεί να κλείσει ένα παράθυρο επιλέγοντας την ετικέτα 'x', δίπλα από το όνομα το παράθυρο της επιλογής του. Επίσης, ο χρήστης μπορεί να κλείσει όλα τα ανοικτά παράθυρα που αφορούν γραφήματα πιέζοντας το κουμπί 'Close All' κάτω και δεξιά.

4.2.1.2 Επιλογή βοήθειας (Help)

Μια αγγλική περιγραφή της λειτουργικότητας κάθε δυνατότητας προς το χρήστη και επιπλέον λειτουργιών της εφαρμογής βρίσκεται στο βασικό μενού, στην επιλογή Help->Show Manual.

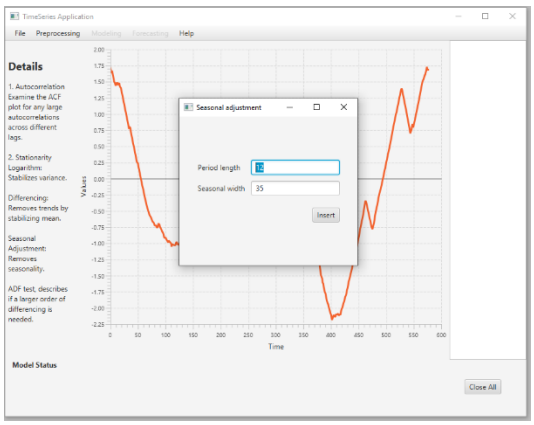


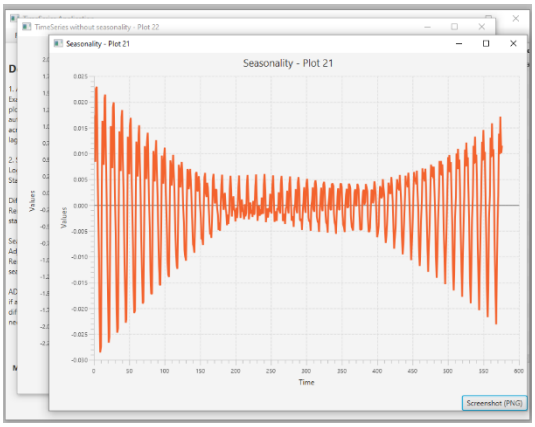
4.2.1.3 Αρχείο καταγραφής ιστορικού (log file)

Η εφαρμογή αποθηκεύσει το ιστορικό χρήσης, από την εκκίνησή της έως και τη λήξη της σε ένα αρχείο με όνομα logs/log##########. Στο όνομα του αρχείου, όπου '##########' είναι ένας μοναδικός αριθμός που προκύπτει από την τρέχουσα ημερομηνία και ώρα.

4.2.1.4 Αφαίρεση εποχικότητας (Προσοχή)

Η εφαρμογή υποστηρίζει ως μετασχηματισμό τη διαγραφή εποχικότητας με την επιλογή Preprocessing->Seasonal Adjustment, η οποία αποτελεί σύνθετη διαδικασία και δεν έχει ελεγχθεί πλήρως.


TimeSeries Application
File
Preprocessing
Modeling
Forecasting
Help
Details
1. Autocorrelation
Examine the ACF plot for any large autocorrelations across different lags.
2. Stationarity
Logarithm:
Stabilizes variance.
Differencing:
Removes trends by stabilizing mean.
Seasonal Adjustment:
Removes seasonality.
ADF test, describes if a larger order of differencing is needed.
Model Status
Seasonal adjustment
Period length
12
Seasonal width
35
Insert
Values
Time
Close All


TimeSeries without seasonality - Plot 22
Seasonality - Plot 21
Seasonality - Plot 21
Values
Time
Screenshot (PNG)

Κεφάλαιο 5. Επίλογος

Στην ενότητα αυτή συνοψίζεται τη συνεισφορά και τα αποτελέσματα της εργασίας και παρατίθενται σκέψεις για μελλοντικές επεκτάσεις της.

5.1 Σύνοψη και συμπεράσματα

Η εργασία κινήθηκε γύρω από το επιστημονικό πεδίο των δεδομένων που συνδέονται με τιμές χρόνου. Στην περιοχή αυτή συναντάται μεγάλος όγκος δεδομένων και ένα ευρύ πλήθος εργαλείων και βιβλιοθηκών, τα περισσότερα από τα οποία είναι δύσχρηστα και απαιτούν ιδιαίτερες γνώσεις. Στόχος της εργασίας αυτής ήταν η δημιουργία μίας εύχρηστης στο χρήστη εφαρμογής για την πρόβλεψη χρονοσειρών μέσω μοντέλων ARIMA. Για το σκοπό αυτό παρουσιάστηκαν τα δύο είδη χρονικών δεδομένων, οι χρονοσειρές και οι ακολουθίες. Δόθηκε ένα εκτεταμένο θεωρητικό υπόβαθρο σχετικό με το αντικείμενο και μία αναλυτική περιγραφή των μοντέλων ARIMA και της χρήσης τους.

Στη συνέχεια, για τη σχεδίαση και την υλοποίηση της εφαρμογής μελετήθηκαν έτοιμες βιβλιοθήκες από το GitHub, από τις οποίες επιλέχθηκαν κάποιες για αξιολόγηση και χρήση. Έπειτα, το υλικό που επιλέχθηκε χρησιμοποιήθηκε στην ανάπτυξη της εφαρμογής, για την οποία δημιουργήθηκε και μία γραφική διεπαφή φιλική προς το χρήστη.

Μέσω Junit tests πιστοποιήθηκε η ορθή λειτουργία της εφαρμογής. Επιπλέον δόθηκε ένα παράδειγμα ως οδηγός χρήσης της και οδηγίες για απαραίτητες εγκαταστάσεις.

5.2 Μελλοντικές επεκτάσεις

Στην ενότητα αυτή παρουσιάζονται ιδέες για επεκτάσεις που μπορούν να πραγματοποιηθούν στο παρόν λογισμικό στο μέλλον. Μια επιπλέον λειτουργία που

μπορεί να γίνει είναι η υλοποίηση αντίστροφων μετασχηματισμών στις προβλέψεις, έτσι ώστε να προσφέρονται στο χρήστη οι τιμές που αντιστοιχούν στα πηγαία δεδομένα. Επίσης, θα μπορούσε να πραγματοποιηθεί υλοποίηση μετρικών που χρησιμοποιούνται για τον έλεγχο εργοδικότητας (stationarity) και για τον έλεγχο καλού ταιριάσματος μοντέλου (fit). Μία ακόμη ιδέα είναι η μοντελοποίηση της εποχικότητας, με επέκταση των υπαρχόντων μοντέλων στα μοντέλα SARIMA (Seasonal ARIMA). Τέλος θα μπορούσε να γίνει πλήρης αξιοποίηση των λειτουργιών της βιβλιοθήκης java-timeseries ή και να διαγραφεί κάθε εξάρτηση από τη βιβλιοθήκη.

Βιβλιογραφία

[AggCC15] Charu C. Aggarwal. Data Mining: The Textbook, pp. 439-508, Springer, 2015.

[Wiki1] "Χρονολογικές Σειρές", Διαθέσιμο στη wikipedia.

(τελευταία προσπέλαση: 11/2/2019)

[Jebb15] A.T. Jebb, L. Tay, W. Wang, Q. Huang. *Time series analysis for psychological research: Examining and forecasting change*, vol. 6, pp. 727, Frontiers in Psychology, Jun. 2015.

[HA2018] R.J. Hyndman, & G. Athanasopoulos. *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2, May 2018. (τελευταία προσπέλαση: 12/12/2019)

[StatHT1] "Akaike's Information Criterion: Definition, Formulas»", Διαθέσιμο στο σύνδεσμο https://www.statisticshowto.datasciencecentral.com/akaikes-information-criterion/ .

(τελευταία προσπέλαση: 18/2/2020)