



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Μονο- και Πολυμεταβλητές Αυτοπαλίνδρομες Τεχνικές για
Συμπύεση Δεδομένων σε Ασύρματα Δίκτυα Αισθητήρων**

Ευάγγελος Δ. Κογχυλάκης

Επιβλέποντες: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής

ΑΘΗΝΑ

Μάρτιος 2015

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μono- και Πολυμεταβλητές Αυτοπαλίνδρομες Τεχνικές για Συμπίεση Δεδομένων σε
Ασύρματα Δίκτυα Αισθητήρων

Ευάγγελος Δ. Κογχυλάκης
A.M.: M1254

ΕΠΙΒΛΕΠΟΝΤΕΣ: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής
Αθανασία Αλωνιστιώτη, Επίκουρος Καθηγητής

Μάρτιος 2015

ΠΕΡΙΛΗΨΗ

Τα ασύρματα δίκτυα αισθητήρων βρίσκουν διαρκώς όλο και περισσότερες εφαρμογές σε πλείστα επιστημονικά πεδία καθώς η δυνατότητα συλλογής πληροφοριών από μεμονωμένες και κινητές μονάδες καθίσταται αναγκαία. Αναπόσπαστο κομμάτι των δικτύων αισθητήρων επίσης είναι και η μεταφορά των δεδομένων, τα οποία πολλές φορές είναι κρίσιμα και για αυτόν τον λόγο πρέπει να υπάρχει ένας μηχανισμός ανάκαμψης σε περίπτωση απώλειας πληροφορίας. Οι αισθητήρες που αποτελούν τις δομικές μονάδες των δικτύων χαρακτηρίζονται από το γνωστό πρόβλημα των πεπερασμένων ενεργειακών πόρων που διατίθενται από τον κάθε αισθητήρα.

Σκοπός αυτής της εργασίας είναι η εισαγωγή νέων μεθόδων επεξεργασίας των δεδομένων ώστε να συμπιέζεται η χρήσιμη πληροφορία πριν αποσταλεί στο υπόλοιπο δίκτυο. Επίσης η νέα μέθοδος επεξεργασίας υποβοηθείται από έναν αλγόριθμο πρόβλεψης του πλαισίου σε περίπτωση χαμένης πληροφορίας αλλά και περαιτέρω συμπίεσης.

Ένας τέτοιος φύσης αλγόριθμος σχεδιάστηκε και υλοποιήθηκε. Ο αλγόριθμος αυτός συνδυάζει την συμπίεση της πληροφορίας επί του πλαισίου διατηρώντας την περισσότερη διαθέσιμη πληροφορία αλλά και την δυνατότητα πρόβλεψης των πλαισίων. Χρησιμοποιήθηκε η ανάλυση των κύριων συνιστωσών καθώς και το αυτοπαλίνδρομο διανυσματικό υπόδειγμα. Ο αλγόριθμος συγκρίθηκε με άλλους μονομεταβλητούς αλγόριθμους πρόβλεψης χρησιμοποιώντας διαφορετικές μετρικές και για διαφορετικούς τύπους δεδομένων. Τα αποτελέσματα είναι ενθαρρυντικά και ειδικότερα ο μέσος όρος του μέσου τετραγωνικού σφάλματος εμφανίζεται ικανοποιητικός σε σχέση με τους υπόλοιπους αλγορίθμους.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Συστήματα Διάχυτου Υπολογισμού

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: συμπίεση πληροφορίας πλαισίου, πρόβλεψη πλαισίου, αυτοπαλίνδρομο διανυσματικό υπόδειγμα, ανάλυση κύριων συνιστωσών, πολυμεταβλητό διανυσματικό μοντέλο

ABSTRACT

Wireless sensor networks are found in numerous applications in most scientific fields as the collection of information from individual and mobile units becomes necessary. An integral part of the network of sensors is the actual transfer of data, which is often critical and for this particular reason there should be a recovery mechanism in case of information loss. The sensors which are the building blocks of networks are characterized by the well-known problem of limited energy resources.

The purpose of this paper is to introduce new data processing methods based on which the relevant information is compressed before it is sent to the rest of the network. Also the new algorithm has an added module which is a frame prediction algorithm where the lost information is being predicted efficiently also used to further compress each frame.

Such an algorithm was designed and implemented. This algorithm combines compressing the information while retaining most of its significance and the predictability of the information. Principal component analysis and the vector autoregressive model were used. The algorithm was compared with other univariate prediction algorithms using different metrics for different datasets. The results are promising and in particular the average mean squared error is satisfactory compared to the other algorithms.

SUBJECT AREA: Pervasive Computing

KEYWORDS: Contextual Information Compression, Context prediction, Vector Autoregressive Model, Principal Component Analysis, Multivariate Vector Model

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	12
1. ΕΙΣΑΓΩΓΗ.....	13
1.1 Διάχυτος Υπολογισμός.....	13
1.2 Δίκτυα αισθητήρων	14
1.3 Δομή του προβλήματος	15
1.4 Παρουσίαση Εργασίας.....	15
2. ΜΕΙΩΣΗ ΔΙΑΣΤΑΣΕΩΝ	17
2.1 Αλγόριθμος SVD	17
2.2 Αλγόριθμος PCA.....	18
2.3 Κριτήρια εφαρμογής PCA	19
2.4 Αλγόριθμος εφαρμογής PCA.....	20
3. ΜΕΘΟΔΟΙ.....	22
3.1 Υπόδειγμα Αυτοπαλίνδρομου Διανύσματος.....	22
3.2 Προεκβολή.....	23
3.3 Προεκβολή παρεμβολής.....	24
3.4 Αφελής προσέγγιση	24
4. ΑΠΟΤΕΛΕΣΜΑΤΑ.....	25
4.1 Εισαγωγή	25
4.1.1 Μέσο τετραγωνικό σφάλμα	25
4.1.2 Ψευδώς αρνητικά	25
4.1.3 Ψευδώς θετικά	25
4.1.4 Συμπίεση	26

4.2	Δεδομένα από τον τομέα ναυσιπλοΐας.....	26
4.2.1	Συνολικό ποσοστό διακύμανσης	26
4.2.2	Ποσοστό σφάλματος	33
4.2.3	Σύγκριση με άλλες μεθόδους	37
4.3	SML2010	46
4.3.1	Σύγκριση μεθόδων.....	46
4.4	Κανονική κατανομή	54
4.4.1	Σύγκριση μεθόδων.....	55
5.	ΣΥΜΠΕΡΑΣΜΑΤΑ	62
6.	ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	63
	ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ	64
	ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ	65
	ΠΑΡΑΡΤΗΜΑ Ι.....	66
	ΑΝΑΦΟΡΕΣ.....	77

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Προεκβολή συνάρτησης	24
Σχήμα 2: Διαφορά μεταξύ προβλεπόμενης τιμής και επόμενης τιμής χρησιμοποιώντας τον ARFIT και 60% της διακύμανσης (πρώτη μεταβλητή)	27
Σχήμα 3: Μέσος όρος του μέσου τετραγωνικού σφάλματος χρησιμοποιώντας τον ARFIT και 60% της διακύμανσης.....	28
Σχήμα 4: Διαφορά μεταξύ προβλεπόμενης τιμής και επόμενης τιμής χρησιμοποιώντας τον ARFIT και 75% της διακύμανσης (πρώτη μεταβλητή)	29
Σχήμα 5: Μέσος όρος του μέσου τετραγωνικού σφάλματος χρησιμοποιώντας τον ARFIT και 75% της διακύμανσης.....	29
Σχήμα 6: Διαφορά μεταξύ προβλεπόμενης τιμής και επόμενης τιμής χρησιμοποιώντας τον ARFIT και 90% της διακύμανσης (πρώτη μεταβλητή)	30
Σχήμα 7: Μέσος όρος του μέσου τετραγωνικού σφάλματος χρησιμοποιώντας τον ARFIT και 90% της διακύμανσης.....	31
Σχήμα 8: Μέσος όρος για το μέσο τετραγωνικό σφάλμα	31
Σχήμα 9: False positive	32
Σχήμα 10: False negative	32
Σχήμα 11: Συμπίεση	33
Σχήμα 12: Διαφορά μεταξύ των πραγματικών τιμών και των τιμών που προβλέπει ο ARFIT ανεξάρτητο οριακού σφάλματος	34
Σχήμα 13: Μέσο τετραγωνικό σφάλμα ανεξάρτητο οριακού σφάλματος	34
Σχήμα 14: False negative για διαφορετικά οριακά σχετικά σφάλματα	35
Σχήμα 15: false positive για διαφορετικά οριακά σχετικά σφάλματα.....	36
Σχήμα 16: Συμπίεση για διαφορετικά οριακά σχετικά σφάλματα	37
Σχήμα 17: Διαφορά προβλεπόμενης και πραγματικής τιμής (ARFIT).....	38
Σχήμα 18: Μέσο τετραγωνικό σφάλμα (ARFIT).....	38
Σχήμα 19: Διαφορά προβλεπόμενης και πραγματικής τιμής (προεκβολή)	39
Σχήμα 20: Μέσο τετραγωνικό σφάλμα (extrapolation).....	39
Σχήμα 21: Διαφορά προβλεπόμενης και πραγματικής τιμής (προεκβολή παρεμβολής).....	40

Σχήμα 22: Μέσο τετραγωνικό σφάλμα (προεκβολή παρεμβολής).....	40
Σχήμα 23: Διαφορά προβλεπόμενης και πραγματικής τιμής (αφελής προσέγγιση).....	41
Σχήμα 24: Μέσο τετραγωνικό σφάλμα (naive).....	41
Σχήμα 25: Σύγκριση False Positive διαφορετικών μεθόδων.....	43
Σχήμα 26: Σύγκριση False Negative διαφορετικών μεθόδων	43
Σχήμα 27: Σύγκριση συμπίεσης διαφορετικών μεθόδων	44
Σχήμα 28: Σύγκριση μέσου όρου τετραγωνικού σφάλματος διαφορετικών μεθόδων	45
Σχήμα 29: Διαφορά προβλεπόμενης και πραγματικής τιμής (ARFIT - SML2010)	46
Σχήμα 30: Μέσο τετραγωνικό σφάλμα (ARFIT - SML2010)	47
Σχήμα 31: Διαφορά προβλεπόμενης και πραγματικής τιμής (Προεκβολή - SML2010).47	
Σχήμα 32: Μέσο τετραγωνικό σφάλμα (Προεκβολή - SML2010).....	48
Σχήμα 33: Διαφορά προβλεπόμενης και πραγματικής τιμής (Προεκβολή παρεμβολής - SML2010).....	49
Σχήμα 34: Μέσο τετραγωνικό σφάλμα (Προεκβολή παρεμβολής - SML2010)	49
Σχήμα 35: Διαφορά προβλεπόμενης και πραγματικής τιμής (Naive - SML2010)	50
Σχήμα 36: Μέσο τετραγωνικό σφάλμα (Naive - SML2010).....	50
Σχήμα 37: Σύγκριση False Positive διαφορετικών μεθόδων (SML2010)	52
Σχήμα 38: Σύγκριση False Negative διαφορετικών μεθόδων (SML2010).....	53
Σχήμα 39: Σύγκριση συμπίεσης διαφορετικών μεθόδων (SML2010)	53
Σχήμα 40: Σύγκριση μέσου όρου τετραγωνικού σφάλματος διαφορετικών μεθόδων (SML2010).....	54
Σχήμα 41: Διαφορά προβλεπόμενης και πραγματικής τιμής (ARFIT - Gaussian)	55
Σχήμα 42: Μέσο τετραγωνικό σφάλμα (ARFIT - Gaussian).....	56
Σχήμα 43: Διαφορά προβλεπόμενης και πραγματικής τιμής (Προεκβολή - Gaussian) ..	56
Σχήμα 44: Μέσο τετραγωνικό σφάλμα (Προεκβολή - Gaussian)	57
Σχήμα 45: Διαφορά προβλεπόμενης και πραγματικής τιμής (Προεκβολή Παρεμβολής - Gaussian)	57
Σχήμα 46: Μέσο τετραγωνικό σφάλμα (Προεκβολή Παρεμβολής - Gaussian)	58

Σχήμα 47: Διαφορά προβλεπόμενης και πραγματικής τιμής (Naive - Gaussian).....	58
Σχήμα 48: Μέσο τετραγωνικό σφάλμα (Naive - Gaussian).....	59
Σχήμα 49: Σύγκριση μέσου όρου τετραγωνικού σφάλματος διαφορετικών μεθόδων (Gauss).....	60
Σχήμα 50: Ιστόγραμμα κανονικής κατανομής	61

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Εφαρμογή αλγορίθμου PCA	21
---	----

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Συνολικές μετρικές ARFIT	42
Πίνακας 2: Συνολικές μετρικές προεκβολής (προεκβολή).....	42
Πίνακας 3: Συνολικές μετρικές προεκβολής παρεμβολής (προεκβολή παρεμβολής)	42
Πίνακας 4: Συνολικές μετρικές αφελής προσέγγισης	42
Πίνακας 5: Συνολικές μετρικές ARFIT (SML2010)	51
Πίνακας 6: Συνολικές μετρικές προεκβολής (SML2010)	51
Πίνακας 7: Συνολικές μετρικές προεκβολής παρεμβολής (SML2010)	51
Πίνακας 8: Συνολικές μετρικές αφελής προσέγγισης (SML2010)	51
Πίνακας 9: Συνολικές μετρικές ARFIT (Gaussian)	59
Πίνακας 10: Συνολικές μετρικές προεκβολής (Gaussian)	59
Πίνακας 11: Συνολικές μετρικές προεκβολής παρεμβολής (Gaussian)	60
Πίνακας 12: Συνολικές μετρικές αφελής προσέγγισης (Gaussian).....	60

ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο του μεταπτυχιακού προγράμματος σπουδών στο τμήμα Πληροφορικής και Τηλεπικοινωνιών στο Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών στην κατεύθυνση "Προηγμένα Πληροφοριακά Συστήματα".

Θα ήθελα να ευχαριστήσω ιδιαίτερα τον Επίκουρο Καθηγητή κ. Ευστάθιο Χατζηευθυμιάδη, που μου έδωσε την ευκαιρία να ασχοληθώ με το συγκεκριμένο ερευνητικό θέμα και έδωσε τις βασικές κατευθύνσεις πάνω στους οποίους αναπτύχθηκε η διπλωματική εργασία.

Επίσης, τον υποψήφιο διδάκτορα κ. Βασίλη Παπαταξιάρχη για την πολύτιμη καθοδήγηση και υποστήριξη του, σε τεχνικά και θεωρητικά θέματα, καθ' όλη την διάρκεια μελέτης και εκπόνησης της διπλωματικής εργασίας.

1. ΕΙΣΑΓΩΓΗ

1.1 Διάχυτος Υπολογισμός

Ο κινητός υπολογισμός (mobile computing) μετεξελίχθηκε σε διάχυτο υπολογισμό (pervasive computing) με θεμελιωτή του τον Mark Weiser. Ο ίδιος έθεσε ως στόχο του διάχυτου υπολογισμού την χρησιμοποίηση πολλαπλών υπολογιστών με τις τεχνολογίες αυτές να υφίστανται μόνο στο παρασκήνιο με αποτέλεσμα ο χρήστης να μην γνωρίζει την χρήση αυτών των τεχνολογιών, αλλά οι ίδιες οι τεχνολογίες να εξατομικεύονται προς την υποστήριξη των χρηστών [1].

Καθώς ένα τέτοιο σύστημα απαρτίζεται από πληθώρα υπολογιστικών δυνατοτήτων πρέπει συν τοις άλλοις να χαρακτηρίζεται και από κινητικότητα (mobility) ώστε να ακολουθείται ο χρήστης/φορέας του συστήματος. Επιπλέον χαρακτηριστικά αυτών των συστημάτων είναι:

- Χρήση ευφύων χώρων: Η ενσωμάτωση υπολογιστικής δυνατότητας στην κτιριακή υποδομή χαρακτηρίζεται ευφύης χώρος και αποτελεί τον συγκερασμό των δύο εννοιών [2].
- Αορατότητα: Αποφυγή της επίγνωσης του χρήστη για το σύστημα διάχυτου υπολογισμού. Το περιβάλλον πρέπει να ικανοποιεί τις προσδοκίες των χρηστών και να τους επιτρέπει μία ξεκούραστη αλληλεπίδραση που πλησιάζει το υποσυνείδητο επίπεδο [3].
- Δυνατότητα κλιμάκωσης: Οι περισσότεροι χρήστες σε διαφορετικούς χώρους περιπλέκουν την χρήση.
- Κάλυψη διαφορετικών συνθηκών περιβάλλοντος: Διαφορετικός εξοπλισμός χρειάζεται για την κάλυψη διαφορετικών χώρων.

Δομικό στοιχείο των παραπάνω υπολογιστικών συστημάτων αποτελεί ο όρος πλαίσιο (context). Ο ορισμός των Dey και Abowd [4] αναφέρει πως ένα πλαίσιο είναι οποιαδήποτε πληροφορία μπορεί να χρησιμοποιηθεί για να περιγράψει την κατάσταση μιας οντότητας (πρόσωπο, αντικείμενο, τοποθεσία), η οποία σχετίζεται με την αλληλεπίδραση μεταξύ του χρήστη και της εφαρμογής, συμπεριλαμβανομένου του χρήστη και της εφαρμογής. Ως πληροφορία πλαισίου ορίζεται οποιαδήποτε πληροφορία μπορεί να περιγράψει την κατάσταση μιας οντότητας. Μία οντότητα μπορεί να είναι μια συσκευή, ένας άνθρωπος ή μία εφαρμογή [4].

Οι μορφές του πλαισίου μπορούν να κατηγοριοποιηθούν χωρίζοντας το πλαίσιο σε εξωτερικό (φυσικό) και εσωτερικό (λογικό) [5]. Το εξωτερικό πλαίσιο το οποίο είναι το πιο διαδεδομένο είναι αυτό που μετράται με συστήματα αισθητήρων (πχ φωτός, θερμοκρασίας, υγρασίας κτλ), πληροφορίες που είναι όλες άμεσα μετρήσιμες. Αντιθέτως, το εσωτερικό πλαίσιο ανιχνεύεται μέσω παρακολούθησης και συγκεκριμένα τους στόχους, το πλαίσιο εργασίας και την ψυχολογική κατάσταση.

1.2 Δίκτυα αισθητήρων

Τα τελευταία χρόνια έχουν επιτευχθεί μεγάλα άλματα στην τεχνολογία που αποτελεί τους ασύρματους αισθητήρες. Η κατάλληλη τεχνολογική πρόοδος στα μικροηλεκτρονικά συστήματα και στην ασύρματη επικοινωνία έδωσε την δυνατότητα για την παραγωγή αισθητήρων που λειτουργούν ως κόμβοι με χαμηλότερο κόστος, χαμηλή κατανάλωση ενέργειας και με περισσότερες λειτουργίες από ότι στο παρελθόν. Αναπτύχθηκαν πρωτόκολλα δικτύων τα οποία επιτρέπουν στους κόμβους την επικοινωνία χωρίς να χρειαστεί η ανθρώπινη παρέμβαση, όταν οι ίδιοι οι κόμβοι είναι σε κοντινή απόσταση. Κάθε τέτοιος κόμβος αισθητήρα συμπεριλαμβάνει ανιχνευτές (sensors) με τους οποίους ανιχνεύουν τις πληροφορίες που είναι χρήσιμες από το περιβάλλον, αλλά και εξαρτήματα επεξεργασίας δεδομένων και επικοινωνίας, και λειτουργεί ταυτόχρονα και παράλληλα με ένα δίκτυο αισθητήρων.

Η εγκατάσταση του δικτύου αισθητήρων γίνεται με δύο τρόπους [6]:

- Εγκατάσταση αισθητήρων οι οποίοι μόνο λαμβάνουν μετρήσεις, δηλαδή επιτελούν μόνο αισθητήρια εργασία. Σε αυτή την περίπτωση η θέση και η επικοινωνιακή τοπολογία σχεδιάζονται μετά από μελέτη. Το υπό παρακολούθηση γεγονός ανιχνεύεται και κατόπιν στέλνονται τα σήματα στους κεντρικούς κόμβους όπου εκτελούνται τυχόν υπολογισμοί και συγχωνεύονται τα δεδομένα.
- Εγκατάσταση αισθητήρων μακριά από το πραγματικό φαινόμενο το οποίο όμως σημαίνει πως απαιτούνται αισθητήρες που χρησιμοποιούν πολύπλοκες τεχνικές και συνεπώς περισσότερη ενέργεια.

Οι ασύρματοι αισθητήρες έχουν βρει πλήθος εφαρμογών είτε σε πεδία όπως ανίχνευση πυρκαγιάς, ανίχνευση ρύπων αλλά και σε πεδία όπως είναι η υγεία και συγκεκριμένα στις εφαρμογές ασύρματης υγείας. Οι αισθητήρες έχουν προτιμηθεί σε σχέση με άλλα συμβατικά μέσα λόγω του χαμηλού κόστους τους και της φορητότητας τους. Για την εφαρμογή τους δεν χρειάζεται ήδη υπάρχουσα υποδομή με υπολογιστικές μονάδες. Έτσι, οι αισθητήρες αυτοί συγκροτούν ένα ή παραπάνω δίκτυα και συλλέγουν δεδομένα τα οποία αποστέλλονται σε μία κεντρική υπολογιστική μονάδα η οποία αποστέλλει τα δεδομένα μέσω δορυφόρου, ή εναλλακτικά κάθε αισθητήρας ξεχωριστά μπορεί να στείλει τα δεδομένα που έχει συλλέξει μέσω μιας γραμμής δορυφορικής προς επεξεργασία.

Οι αισθητήρες επίσης χαρακτηρίζονται από συνεργατική λειτουργία των κόμβων, καθώς αυτοί περιλαμβάνουν και έναν επεξεργαστή ο οποίος χρησιμοποιείται έτσι ώστε αντί να στέλνονται ακατέργαστα δεδομένα στους κεντρικούς κόμβους, να εφαρμόζονται απλοί αλγόριθμοι και να εκπέμπουν μόνο τα επεξεργασμένα δεδομένα. Αυτό εξασφαλίζει ένα μεγάλο πλήθος εφαρμογών για τα δίκτυα αισθητήρων.

Για την υλοποίηση των παραπάνω απαιτούνται ad-hoc τεχνικές. Οι σημαντικότερες διαφορές όμως των δύο δικτύων είναι [7]:

- Το πλήθος των κόμβων μπορεί να είναι πολύ μεγαλύτερο σε ένα δίκτυο αισθητήρων από ότι σε ένα ad-hoc δίκτυο
- Η χωρική πυκνότητα των δικτύων αισθητήρων είναι μεγάλη
- Οι αισθητήριοι κόμβοι είναι εύκολο να καταστραφούν
- Η τοπολογία ενός δικτύου αισθητήρων συχνά αλλάζει
- Οι αισθητήριοι κόμβοι χρησιμοποιούν κυρίως επικοινωνία broadcast ενώ τα περισσότερα δίκτυα ad hoc βασίζονται στην επικοινωνία σημείου προς σημείο (point to point).

- Οι αισθητήριοι κόμβοι διακρίνονται από περιορισμούς στους τομείς της ενέργειας, της υπολογιστικής ισχύος και της μνήμης.

1.3 Δομή του προβλήματος

Ενώ όλα αυτά θεωρητικά είναι δυνατά, στην πράξη υπάρχει ένας παράγοντας που μπορεί να κάνει την χρήση των αισθητήρων απαγορευτική και αυτός δεν είναι άλλος από την κατανάλωση ενέργειας [8]. Οι αισθητήρες σε μερικές περιπτώσεις δεν συλλέγουν συνέχεια πληροφορίες, και μπορούν ενδιάμεσα από τις χρονικές στιγμές που είναι ενεργοί να μπαίνουν σε καταστάσεις χαμηλής ενέργειας (sleep, hibernate) και ξαναγίνονται ενεργοί μέσω του ρολογιού της κάθε πλακέτας πάνω στην οποία λειτουργεί ο αισθητήρας. Ακόμα και έτσι όμως η ενέργεια που τροφοδοτεί τον κάθε αισθητήρα από την μπαταρία είναι πεπερασμένη με αποτέλεσμα να κρατάει μόνο για μερικές μέρες στην καλύτερη των περιπτώσεων. Στις περιπτώσεις όπου η μπαταρία τελειώσει απαιτείται μία συνεχής επισκόπηση των αισθητήρων η οποία είναι ασύμφορη από οικονομικής πλευράς και επιπλέον σπαταλούνται πόροι.

Σε αυτό το σημείο έρχεται η αποτελεσματική συμβολή μίας έξυπνης διαχείρισης της ενέργειας μέσω προφανώς της έξυπνης διαχείρισης των δεδομένων. Τα δεδομένα είναι κρίσιμα οπότε σε κάθε περίπτωση δεν πρέπει να απορρίπτονται μετρήσεις οι οποίες είναι μεγάλης σημασίας ως προς την φύση των παραμέτρων των μετρήσεων. Για να μειωθεί η μεγάλη κατανάλωση ενέργειας από τους αυτόνομους αισθητήρες στο κάθε δίκτυο που είναι τοποθετημένοι, θα πρέπει να μειωθεί και η μεταγωγή των μετρήσεων μέσω εσωτερικών κριτηρίων. Με αυτόν τον τρόπο μπορεί να μειωθεί η κατανάλωση ενέργειας αλλά και το bandwidth της γραμμής που χρησιμοποιείται.

1.4 Παρουσίαση Εργασίας

Η εργασία αυτή προσανατολίζεται προς αυτή την κατεύθυνση της μείωσης της κατανάλωσης ενέργειας των μονάδων των αισθητήρων. Αυτό επιτυγχάνεται με μία προεπεξεργασία των δεδομένων ώστε να προωθούνται μόνο συγκεκριμένα διανύσματα μετρήσεων και όχι όλα. Κάθε πλαίσιο μετρήσεων αποτελείται από πολλαπλές μεταβλητές μετρήσεων και αφορά μία συγκεκριμένη χρονική στιγμή. Άρα σε κάθε διαφορετική χρονική στιγμή έχουμε και ένα διαφορετικό πλαίσιο μεταβλητών που περιλαμβάνει όλες τις μεταβλητές οι οποίες αποτελούν αξιοποιήσιμα μεγέθη.

Τα πλαίσια που αποστέλλονται μεταξύ των κόμβων στο δίκτυο αισθητήρων έχουν υποστεί μείωση διαστάσεων με αποτέλεσμα να έχει παραμείνει ένα πλαίσιο το οποίο αποτελείται από μεταβλητές οι οποίες είναι γραμμικοί συνδυασμοί των αρχικών μεταβλητών/μετρήσεων. Αυτές οι μεταβλητές επιλέγονται έτσι ώστε να μεγιστοποιούν την διακύμανση, και ο αλγόριθμος που χρησιμοποιείται περιλαμβάνει ένα συγκεκριμένο ποσοστό διακύμανσης το οποίο περιλαμβάνεται σαν σταθερά και κυμαίνεται από 60 μέχρι 90 τοις εκατό. Το πλήθος των επιλεγμένων μεταβλητών είναι σαφώς μικρότερο από το πλήθος των αρχικών μεταβλητών και έτσι με αυτόν τον τρόπο επιτυγχάνεται συμπίεση των δεδομένων που αποστέλλονται μεταξύ των κόμβων.

Στην περίπτωση όπου ένας κόμβος A ο οποίος πρέπει να στείλει τα δεδομένα σε έναν κόμβο B είτε αποφασίσει ότι δεν θα στείλει νέα πλαίσια για ένα χρονικό διάστημα για να εξοικονομήσει ενέργεια, είτε για τεχνικούς λόγους δεν καταστεί δυνατή η επικοινωνία προς τον κόμβο B θα πρέπει να προβλεφθούν τα πλαίσια. Σε αυτή την περίπτωση ο συγκεκριμένος αλγόριθμος χρησιμοποιείται από τον κόμβο B ώστε να δημιουργήσει με βάση τις προηγούμενες τιμές (πλαίσια) που είχε λάβει τα νέα πλαίσια με τις

αναμενόμενες τιμές τους. Έτσι χωρίς να μεταφερθούν καθόλου δεδομένα μεταξύ των δύο κόμβων ο κόμβος Β έχει δημιουργήσει με το μικρότερο δυνατό σφάλμα, το πλαίσιο το οποίο είναι το αναμενόμενο. Στο κεφάλαιο 4 υπάρχουν τα αποτελέσματα που χαρακτηρίζουν αυτές τις προβλέψεις και αντιπαραβάλλονται με άλλες μεθόδους.

Ο αλγόριθμος πρόβλεψης χρησιμοποιεί το αυτοπαλίνδρομο σύστημα πολλών μεταβλητών ώστε να μπορεί να λάβει πληροφορίες αλλαγής από διαφορετικές μεταβλητές και δεν χρησιμοποιεί ένα απλό μονομεταβλητό σύστημα το οποίο για να προβλέψει τις μελλοντικές τιμές, παράγει κάθε μία προβλεπόμενη μεταβλητή ξεχωριστά με βάση τις τελευταίες τιμές του.

Επιπλέον το αυτοπαλίνδρομο υπόδειγμα που χρησιμοποιείται δίνει την δυνατότητα περαιτέρω συμπίεσης αφού αν το πλαίσιο έχει σχετικό σφάλμα από το προηγούμενο κάτω από μία οριακή τιμή, αυτό δεν θα σταλεί ποτέ, και στην θέση του θα σταλεί ένα χαμηλής ενέργειας σήμα το οποίο θα σηματοδοτήσει ότι ο λαμβάνων κόμβος αρκεί να χρησιμοποιήσει το παραγόμενο από τον αλγόριθμο πλαίσιο.

2. ΜΕΙΩΣΗ ΔΙΑΣΤΑΣΕΩΝ

2.1 Αλγόριθμος SVD

Κάθε πλαίσιο από τους αισθητήρες περιέχει διαφορετικές μετρήσεις προερχόμενες από διαφορετικά όργανα μέτρησης. Το πλεονέκτημα που υπάρχει σε αυτή την περίπτωση είναι ότι έχουμε περισσότερη πληροφορία για την κάθε κατάσταση με αποτέλεσμα οι εξαγωγές των συμπερασμάτων να είναι πιο ασφαλείς αφού περισσότερη πληροφορία σημαίνει πιο πολλές σχέσεις μεταξύ των μεταβλητών. Από την άλλη όμως το μειονέκτημα που υπάρχει γνωστό και ως κατάρα των διαστάσεων (curse of dimensionality) είναι ότι επειδή υπάρχουν περισσότερες πληροφορίες, θα πρέπει να σπαταλήσουμε περισσότερη υπολογιστική ισχύ για την επεξεργασία όλων αυτών των μεταβλητών. Σε πολλές περιπτώσεις οι μεταβλητές αυτές είναι άμεσα συνδεδεμένες και μία μεταβολή στην μία (από ένα πλαίσιο στο επόμενο του) είναι πολύ πιθανό να σημαίνει και την μεταβολή της άλλης. Σε άλλες περιπτώσεις μπορεί απλά οι δύο ή περισσότερες αυτές μεταβλητές να περιγράφουν το ίδιο ακριβώς μέγεθος όπως είναι πχ η θερμοκρασία αλλά σε διαφορετικές μονάδες όπως πχ Κελσίου, Φαρενάιτ.

Ο αλγόριθμος Singular Value Decomposition (SVD) βοηθάει σε αυτές τις περιπτώσεις αφού χρησιμοποιώντας τον μπορούμε να εξαλείψουμε μερικές μεταβλητές οι οποίες είτε αποτελούν θόρυβο (noise) είτε είναι γραμμικά εξαρτώμενες από άλλες μεταβλητές με αποτέλεσμα να παραμένουν μόνο οι πιο σημαντικές για την ανάλυση του προβλήματος. Ο αλγόριθμος SVD δίνει την δυνατότητα να μελετηθεί μόνο ένα ποσοστό από τις πιο σημαντικές μεταβλητές χρησιμοποιώντας τις ιδιοτιμές και την παραγοντοποίηση του αρχικού πίνακα.

Αν υποθέσουμε ότι έχουμε έναν πίνακα A με n σειρές (διαφορετικά πλαίσια) και p στήλες (διαφορετικές μεταβλητές του ίδιου πλαισίου).

Με βάση τον SVD ο A μπορεί να γραφεί ως:

$$A_{n \times p} = U_{n \times n} S_{n \times p} V^T_{p \times p}$$

όπου

$$U^T U = I_{n \times n}$$

$$V^T V = I_{p \times p}$$

Για να βρούμε τον SVD ουσιαστικά πρέπει να βρούμε τις ιδιοτιμές και τα ιδιοδιανύσματα των πινάκων AA^T και $A^T A$. Ουσιαστικά τα ιδιοδιανύσματα του $A^T A$ είναι οι στήλες του V και τα ιδιοδιανύσματα του AA^T είναι οι στήλες του U . Ο πίνακας S είναι ένας διαγώνιος πίνακας όπου οι τιμές του είναι οι singular τιμές του πίνακα A . Ο αριθμός των τιμών αυτών είναι η τάξη του πίνακα, δηλαδή ο αριθμός των γραμμικά ανεξάρτητων στηλών ή γραμμών. Όλες αυτές οι τιμές είναι μεγαλύτερες από 0 και μειώνεται η τιμή τους σε κάθε επόμενη σειρά. Επιλέγοντας λιγότερες τέτοιες τιμές από τον μέγιστο αριθμό, εξάγεται μία προσέγγιση του αρχικού πίνακα όπου έχουν μειωθεί οι διαστάσεις του αρχικού πίνακα στις πιο σημαντικές.

Ουσιαστικά αυτό που εξάγεται από την τελευταία εξίσωση είναι ότι ένας οποιοσδήποτε πίνακας A μπορεί να μετασχηματιστεί σε έναν ορθογώνιο, έναν διαγώνιο και άλλον ένα ορθογώνιο πίνακα.

2.2 Αλγόριθμος PCA

Ο αλγόριθμος PCA ουσιαστικά ψάχνει μία διαφορετική από την υπάρχουσα βάση, η οποία μάλιστα θα είναι γραμμικός συνδυασμός της υπάρχουσας βάσης η οποία περιγράφει καλύτερα τα δεδομένα.

Έστω X ο $m \times n$ πίνακας δεδομένων, Y ένας άλλος πίνακας ιδίων διαστάσεων και έναν γραμμικό μετασχηματισμό P . Τότε θα ισχύει στην περίπτωση αυτή:

$$PX = Y$$

Επίσης αν θεωρήσουμε τα εξής:

x_i τις στήλες του X ,

y_i τις στήλες του Y ,

p_i τις σειρές του P ,

έχουμε ότι:

$$PX = \begin{bmatrix} p_1 \\ \vdots \\ p_m \end{bmatrix} \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}$$

$$Y = \begin{bmatrix} p_1 \cdot x_1 & \cdots & p_1 \cdot x_n \\ \vdots & \ddots & \vdots \\ p_m \cdot x_1 & \cdots & p_m \cdot x_n \end{bmatrix}$$

όπου τα στοιχεία του Y είναι ένα σύνολο δεδομένων που περιγράφουν τα δεδομένα του X σε μία νέα βάση. Τα στοιχεία του P αποτελούν τις κύριες συνιστώσες (principal components) του X .

Η συνδιακύμανση μετράει τον βαθμό της γραμμικής εξάρτησης μεταξύ δύο μεταβλητών. Μία μεγάλη θετική τιμή συνδιακύμανσης υποδεικνύει θετικά εξαρτώμενα δεδομένα. Παρομοίως μία μεγάλη αρνητική συνδιακύμανση υποδεικνύει αρνητικά σχετιζόμενα δεδομένα.

Η συνδιακύμανση του X βρίσκεται από τον τύπο:

$$C_X \equiv \frac{1}{n} XX^T.$$

Οι υποθέσεις του PCA είναι η γραμμικότητα και ότι κύριες συνιστώσες είναι ορθογώνιες. Η συνδιακύμανση του πίνακα Y είναι ένας διαγώνιος πίνακας αφού θέλουμε το Y να είναι μη εξαρτώμενο (decorrelated).

Έτσι μπορούμε να την ξαναγράψουμε ως

$$\begin{aligned}
\mathbf{C}_Y &= \frac{1}{n} \mathbf{Y} \mathbf{Y}^T \\
&= \frac{1}{n} (\mathbf{P} \mathbf{X}) (\mathbf{P} \mathbf{X})^T \\
&= \frac{1}{n} \mathbf{P} \mathbf{X} \mathbf{X}^T \mathbf{P}^T \\
&= \mathbf{P} \left(\frac{1}{n} \mathbf{X} \mathbf{X}^T \right) \mathbf{P}^T \\
\mathbf{C}_Y &= \mathbf{P} \mathbf{C}_X \mathbf{P}^T
\end{aligned}$$

όπου περιέχεται η συνδιακύμανση του δεδομένου πίνακα X.

Επιλέγουμε τον πίνακα P να είναι ένας πίνακας που κάθε γραμμή αυτού θα είναι ένα ιδιοδιάνυσμα της συνδιακύμανσης του X και λύνοντας την παραπάνω εξίσωση καταλήγουμε ότι η συνδιακύμανση του Y είναι ένας διαγώνιος πίνακας.

Το να υπολογίσουμε όμως την συνδιακύμανση του X θα είχε σαν αποτέλεσμα το να χάσουμε δεδομένα και ακρίβεια. Έτσι αφού λόγω SVD μπορούμε να γράψουμε τον πίνακα X ως:

$$\mathbf{X} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$$

η συνδιακύμανση του X γίνεται (για κατάλληλα δεδομένα με μέση τιμή το μηδέν ώστε να απαλείφεται ο παρανομαστής) :

$$\mathbf{X} \mathbf{X}^T = (\mathbf{U} \mathbf{\Sigma} \mathbf{V}^T) (\mathbf{U} \mathbf{\Sigma} \mathbf{V}^T)^T$$

$$\mathbf{X} \mathbf{X}^T = (\mathbf{U} \mathbf{\Sigma} \mathbf{V}^T) (\mathbf{V} \mathbf{\Sigma} \mathbf{U}^T)$$

και αφού ο V είναι ορθογώνιος $\mathbf{V}^T \mathbf{V} = \mathbf{I}$

$$\mathbf{X} \mathbf{X}^T = \mathbf{U} \mathbf{\Sigma}^2 \mathbf{U}^T$$

2.3 Κριτήρια εφαρμογής PCA

Για την εφαρμογή του αλγορίθμου Principal Component Analysis (PCA) χρησιμοποιήθηκε η εντολή PCA του MATLAB όπου εσωτερικά χρησιμοποιεί τον αλγόριθμο svd. Πριν γίνει η εισαγωγή του πίνακα στην παραπάνω συνάρτηση τα δεδομένα έχουν επεξεργαστεί ώστε να έχουν μέσο το 0 και απόκλιση 1.

Η συνάρτηση PCA δίνει τα εξής αποτελέσματα:

coeff: Είναι οι συντελεστές των αρχικών μεταβλητών.

Κάθε μία σειρά αποτελεί και μία κύρια συνιστώσα και έχει από αυτούς τους συντελεστές προκύπτουν οι γραμμικοί συνδυασμοί που περιγράφουν κάθε μία γραμμική συνιστώσα.

explained: το ποσοστό της συνολικής διακύμανσης που είναι περιλαμβάνει κάθε μία από τις κύριες συνιστώσες.

2.4 Αλγόριθμος εφαρμογής PCA

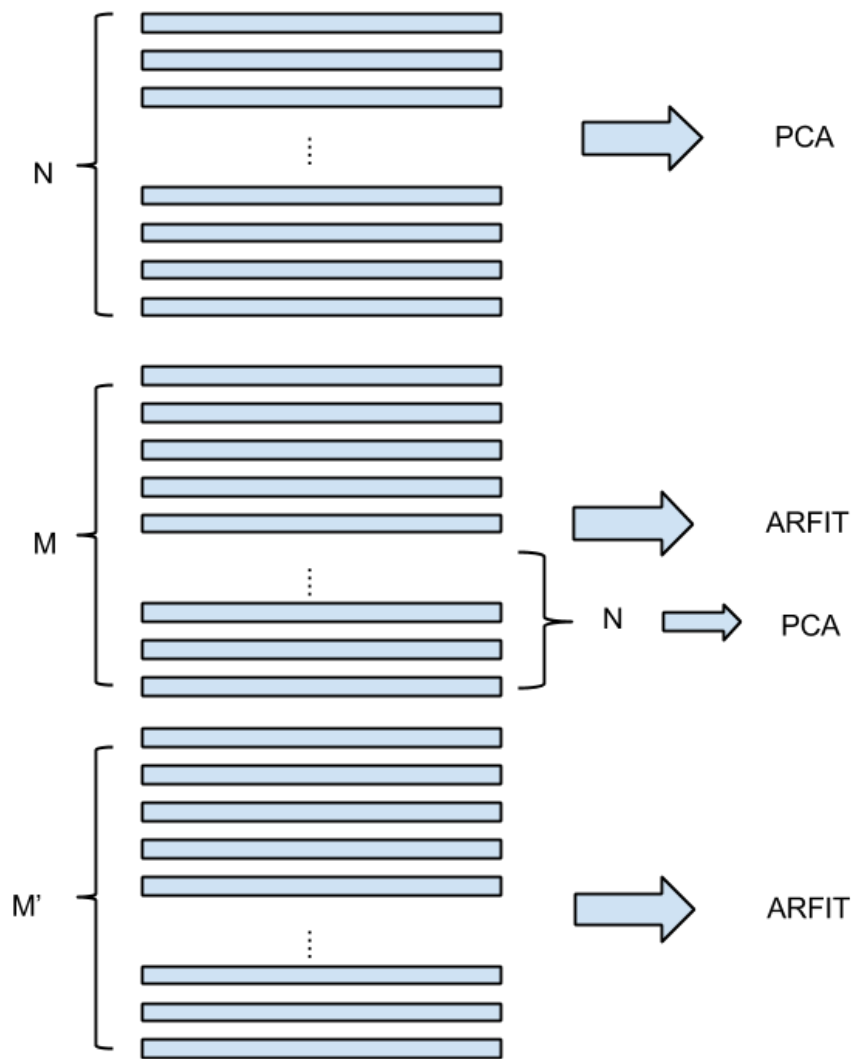
Για την εφαρμογή του PCA, ώστε να μειωθεί η διάσταση των δεδομένων πρέπει να επιλέξουμε ένα συγκεκριμένο όριο της συνολικής διακύμανσης που θέλουμε τα δεδομένα μας να περιγράφουν [9]. Προφανώς αν επιλεχθεί το 100% σε κάθε περίπτωση θα λάβουμε το σύνολο των αρχικών μεταβλητών μας σε πλήθος, μετατοπισμένες σε γραμμικούς συνδυασμούς αυτών. Συνήθεις επιλογές αυτού του ορίου είναι το 90-95% με αποτέλεσμα το πλήθος από τις κύριες συνιστώσες να είναι μικρότερο κατά ένα μεγάλο ποσοστό.

Επειδή τα πλαίσια λαμβάνονται από τους αισθητήρες ανά τακτά χρονικά διαστήματα ο τρόπος επεξεργασίας τους και ειδικότερα η εφαρμογή της μείωσης των διαστάσεων γίνεται σχεδόν σύγχρονα. Την πρώτη φορά εφαρμογής του αλγορίθμου, εφαρμόζεται ο PCA και αναγκαστικά θα υπάρχει μία αναμονή N δειγμάτων, όπου N είναι το παράθυρο στο οποίο εφαρμόζεται το PCA. Από αυτά τα N πλαίσια λαμβάνουμε τους συντελεστές για τις κύριες συνιστώσες.

Για τα επόμενα M πλαίσια χρησιμοποιούνται αυτοί οι συντελεστές. Επειδή τα πλαίσια που ανήκουν στο πλήθος M επεξεργάζονται μεμονωμένα, οι παραπάνω συντελεστές εφαρμόζονται μεμονωμένα σε κάθε πλαίσιο. Όταν παρέλθουν οι M τιμές, ο αλγόριθμος περνάει στην δεύτερη επανάληψη. Για να μην υπάρχει πάλι η αναμονή N τιμών χρησιμοποιούνται οι N τελευταίες τιμές οι οποίες επικαλύπτονται με τα M πλαίσια. Εφαρμόζοντας πάλι τον αλγόριθμο PCA σε αυτά τα N πλαίσια εξάγονται νέες κύριες συνιστώσες οι οποίες χρησιμοποιούνται για τα επόμενα M' πλαίσια.

Το πλήθος M' δεν είναι κατ' ανάγκη το ίδιο με το M . Το κριτήριο που χρησιμοποιήθηκε είναι ότι για το ίδιο ποσοστό εξασφαλισμένης διακύμανσης αν προκύψει ότι πάλι χρειάζεται το ίδιο πλήθος κύριων συνιστωσών τότε θα παραμείνει το ίδιο δηλαδή $M' = M$. Σε κάθε άλλη περίπτωση, δηλαδή αν χρειάζονται για το ίδιο ποσοστό εξασφαλισμένης διακύμανσης λιγότερες ή περισσότερες κύριες συνιστώσες, τότε το M' δεν θα είναι ίσο με το M , αλλά θα έχει μειωθεί κατά N .

Στο παρακάτω σχήμα φαίνονται οι δύο πρώτες επαναλήψεις του αλγορίθμου εφαρμογής του PCA.



Εικόνα 1: Εφαρμογή αλγορίθμου PCA

3. ΜΕΘΟΔΟΙ

3.1 Υπόδειγμα Αυτοπαλίνδρομου Διανύσματος

Η απόφαση για την αποστολή ή όχι του πλαισίου βασίζεται στην μέθοδο πρόβλεψης του επόμενου πλαισίου με βάση τα προηγούμενα. Κάθε πλαίσιο όμως έχει πολλές μεταβλητές οι οποίες μπορούν να αλληλοεπηρεάζονται.

Για να γίνει η πρόβλεψη του επόμενου πλαισίου βασικό στοιχείο είναι το αυτοπαλίνδρομο διάνυσμα. Το Υπόδειγμα Αυτοπαλίνδρομου Διανύσματος (Multivariate Auto-Regressive model - MAR) χρησιμοποιείται για την ανάλυση πολυμεταβλητών χρονοσειρών. Αποτελεί επέκταση του υποδείγματος αυτοπαλινδρόμησης μιας μεταβλητής και για αυτό υπερτερεί των αντίστοιχων μονομεταβλητών μοντέλων πρόβλεψης. Οι προβλέψεις στο MAR διακρίνονται από ευελιξία αφού μπορεί να υπάρχει εξάρτηση από πιθανή μελλοντική πορεία των μεταβλητών.

Ουσιαστικά μία σειρά από διαδοχικές μετρήσεις μιας σειράς περιέχει πληροφορία για την διεργασία που την δημιουργήσε. Για την προσπάθεια περιγραφής της σειράς απαιτείται να μοντελοποιήσουμε την τρέχουσα τιμή x_t ως ένα σταθμισμένο γραμμικό άθροισμα των προηγούμενων τιμών x_{t-1} , x_{t-2} κτλ

Αυτή είναι μία αυτοπαλίνδρομη διαδικασία και πολύ απλή αλλά και αποδοτική προσέγγιση των χρονοσειρών. Η τάξη του μοντέλου είναι ο αριθμός των παρατηρήσεων που προηγούνται και χρησιμοποιούνται. Τα βάρη είναι οι παράμετροι του μοντέλου που προσδιορίζονται από τα δεδομένα που χαρακτηρίζουν μοναδικά την χρονοσειρά. Τα πολυμεταβλητά αυτοπαλίνδρομα μοντέλα επεκτείνουν αυτή την προσέγγιση σε πολλαπλές χρονοσειρές ώστε το διάνυσμα των τρεχουσών τιμών $x_t = [x_{1t}, x_{2t}, \dots, x_{nt}]$ όλων των μεταβλητών να μοντελοποιείται σαν το γραμμικό άθροισμα των προηγούμενων διανυσμάτων. Έτσι το μοντέλο MAR περιγράφει την δυναμική συμπεριφορά της χρονοσειράς ως προς την πρόβλεψη αυτής.

Ας θεωρήσουμε

$$x_t = (x_{1t}, x_{2t}, \dots, x_{nt})^T$$

είναι ένα n -διάστατο διάνυσμα μιας χρονοσειράς. Το τάξης ϕ MAR μοντέλο παίρνει την μορφή:

$$x_t = c + \Pi_1 x_{t-1} + \Pi_2 x_{t-2} + \dots + \Pi_\phi x_{t-\phi} + \epsilon_t, \\ t = 1, \dots$$

Το c είναι ένα σταθερό διάνυσμα, τα Π είναι $(n \times n)$ πίνακες συντελεστών και το ϵ_t είναι λευκός θόρυβος με μέση τιμή το μηδέν και αμετάβλητο πίνακα συνδιακύμανσης. Η πρόβλεψη είναι ένας από τους κύριους στόχους του MAR. Όλες οι παραπάνω παράμετροι μπορούν να υπολογιστούν. Η καλύτερη γραμμική πρόβλεψη, από την άποψη του ελάχιστου μέσου τετραγωνικού σφάλματος, βάσει των διαθέσιμων πληροφοριών μέχρι εκείνο το σημείο είναι η εξής:

$$\tilde{x}_{t+1} = c + \Pi_1 x_t + \dots + \Pi_\phi x_{t-\phi+1}$$

Το εργαλείο που χρησιμοποιήθηκε ήταν το ARfit MATLAB, στο οποίο οι παραπάνω παράμετροι c , Π υπολογίστηκαν χρησιμοποιώντας τον αλγόριθμο ελαχίστων τετραγώνων [10].

Συγκεκριμένα για την επιλογή της τάξης ϕ , ενός αυτοπαλίνδρομου μοντέλου και τον υπολογισμό των παραμέτρων c και Π , ο κλιμακωτός αλγόριθμος ελαχίστων τετραγώνων που περιγράφεται από τους Neumaier και Schneider [11], υλοποιείται στο ARfit.

Λαμβάνοντας σαν είσοδο μια χρονοσειρά , αποτελούμενη από $N+\varphi$ διανύσματα κατάστασης u_n , ($n = 1- \varphi$, , N) και ενός κάτω $\varphi_{\min}$ και άνω $\varphi_{\max}$ ορίου, όσον αφορά την τάξη του μοντέλου , η συνάρτηση **arfit** αξιολογεί τα κριτήρια για την επιλογή της τάξης του μοντέλου, για μια ακολουθία αυτοπαλίνδρομων μοντέλων , διαδοχικής τάξης $\varphi_{\min}$ και $\varphi_{\max}$ και υπολογίζει τις παραμέτρους για ένα μοντέλο βέλτιστης τάξης φ_{opt} .

Σαν βέλτιστη τάξη φ_{opt} , ενός αυτοπαλίνδρομου μοντέλου επιλέγεται αυτή που βελτιστοποιεί έναν κριτήριο επιλογής τάξης [12]. Τα κριτήρια επιλογής τάξης που υλοποιούνται στο ARfit, είναι: κριτήριο τελευταίου λάθους πρόβλεψης (Final Prediction Error – FPE), [13] και Schwarz's - Bayesian κριτήριο (SBC) [14]. Ο Lutkepohl [15] συνέκρινε αυτά τα δύο και άλλα κριτήρια επιλογής τάξης σε μια μελέτη προσομοίωσης και διαπίστωσε ότι το SBC επιλέγει την σωστή τάξη για το μοντέλο συχνότερα και οδηγεί , κατά μέσο όρο, σε μικρότερα μέσης τιμής τετραγωνικά σφάλματα πρόβλεψης του εφαρμοζόμενου αυτοπαλίνδρομου μοντέλου. Συνεπώς το SBC , είναι το προεπιλεγμένο κριτήριο που χρησιμοποιείται στο ARfit και αυτό χρησιμοποιείται σε όλη την διπλωματική εργασία.

3.2 Προεκβολή

Μία άλλη μέθοδος που χρησιμοποιήθηκε για να συγκρίνουμε τα αποτελέσματα του αλγορίθμου ήταν η προεκβολή (extrapolation).

Αυτή η μέθοδος σε αντίθεση με της μέθοδο αυτοπαλινδρόμησης σε κάθε εφαρμογή της προεκβολής αυτή γίνεται σε κάθε μία μεμονωμένη μεταβλητή και όχι στο σύνολο αυτών. Άρα αντί για να προβλεφθεί ένα πλαίσιο σαν έξοδος αυτής της μεθόδου, προβλέπεται μία τιμή για κάθε μεταβλητή και συγκροτούμενες αυτές αποτελούν το προβλεπόμενο διάνυσμα τιμών.

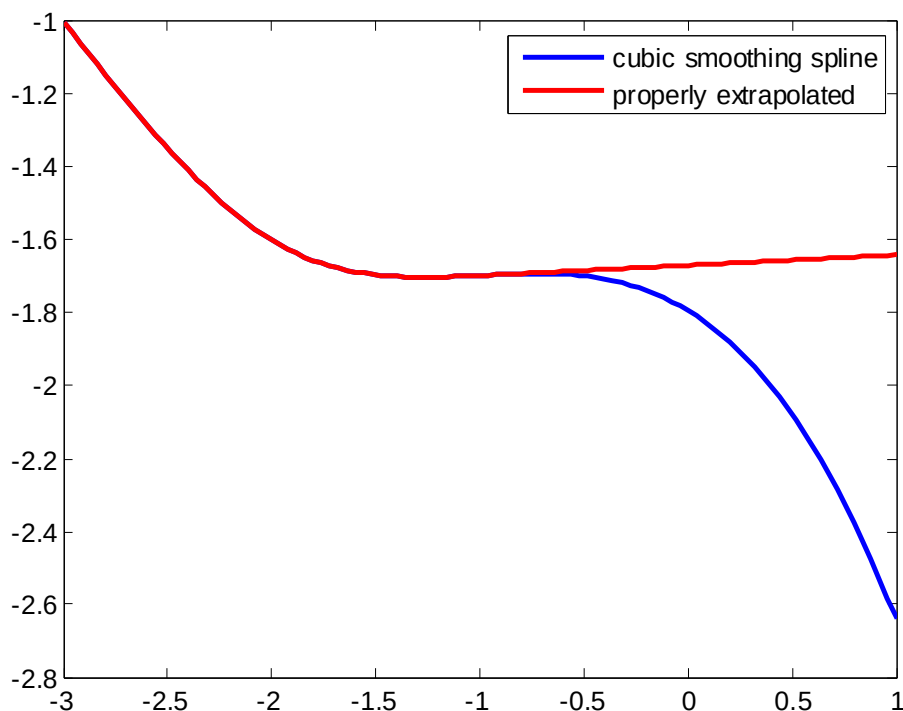
Η μέθοδος αυτή αρχικά βρίσκει μία κυβική καμπύλη - τμηματικά πολυώνυμη - με παράγοντα εξομάλυνσης (cubic smoothing spline) με βάση τα σημεία στους άξονες x,y που παίρνει ως παραμέτρους. Τα σημεία x είναι ακέραιοι και έχουν ως μέγιστο το -1. Τα σημεία y είναι οι ακριβείς μετρήσεις. Η προεκβολή γίνεται στο σημείο $x=0$ το οποίο χρονικά είναι η επόμενη τιμή.

Για παράδειγμα έστω ότι πρέπει να εφαρμοστεί προεκβολή για το επόμενο σημείο της μεταβλητής k , όταν οι τελευταίες τιμές είναι 5.1 , 10.3 , -4.4

Έτσι οι συντεταγμένες γίνονται:

$$\begin{aligned}x_{-3} &= 5.1 \Rightarrow (-3, 5.1) \\x_{-2} &= 10.3 \Rightarrow (-2, 10.3) \\x_{-1} &= -4.4 \Rightarrow (-1, -4.4)\end{aligned}$$

Χρησιμοποιώντας την συνάρτηση **csaps** από το MATLAB γίνεται λήψη της συνάρτησης που περιγράφει καλύτερα αυτά τα σημεία. Υπάρχει παράγοντας εξομάλυνσης p , ο οποίος για $p=0$, δίνει μία ευθεία αντί για καμπύλη και για $p=1$, η καμπύλη πάει στο άλλο άκρο. Με βάση αυτή την καμπύλη μετέπειτα δίνεται ως παράμετρος στην συνάρτηση του MATLAB **fnxtr** η οποία και εφαρμόζει την προεκβολή έξω από το διάστημα της καμπύλης. Έτσι διατηρείται μία ομαλότητα δεύτερης τάξης έξω από το διάστημα της καμπύλης. Αυτό συμβαίνει για να αποφευχθεί το λάθος αποτέλεσμα που μπορεί να βγάλει η μία μη καμπύλη η οποία δεν έχει εξομαλυνθεί. Αυτό φαίνεται και στο παρακάτω σχήμα.



Σχήμα 1: Προεκβολή συνάρτησης

3.3 Προεκβολή παρεμβολής

Η επόμενη μέθοδος που χρησιμοποιήθηκε ήταν η εφαρμογή μονομεταβλητής προεκβολής αφότου έχει υπάρξει συνάρτηση παρεμβολής. Έτσι αρχικά χρησιμοποιούνται δύο διαφορετικές συναρτήσεις του MATLAB για την εύρεση συνάρτησης προεκβολής, η κάθε μία με τα δικά της χαρακτηριστικά:

spline - Cubic spline data interpolation

pchip - Piecewise Cubic Hermite Interpolating Polynomial (PCHIP)

Η κύρια μέθοδος που χρησιμοποιείται είναι η συνάρτηση `interp1` η οποία κάνει την γραμμική παρεμβολή μεταξύ των δεδομένων τα οποία έχουμε δώσει σαν είσοδο και χρησιμοποιεί όποια από τις δύο παραπάνω συναρτήσεις έχουμε ορίσει ώστε να κάνει προεκβολή στα σημεία εξωτερικά του πεδίου που έχουμε ορίσει. Χρησιμοποιείται η `pchip`.

3.4 Αφελής προσέγγιση

Η αφελής προσέγγιση (Naive Approach) χρησιμοποιήθηκε καθαρά για σύγκριση με τους υπόλοιπους αλγορίθμους και ουσιαστικά σε κάθε πλαίσιο που λαμβάνεται, θεωρεί ότι το ίδιο πλαίσιο θα επαναληφθεί. Προφανώς αυτή η μέθοδος δεν μπορεί να προβλέψει μεγάλες αποκλίσεις αλλά από την άλλη έχει το πλεονέκτημα ότι δεν χρειάζεται υπολογιστικούς πόρους.

4. ΑΠΟΤΕΛΕΣΜΑΤΑ

4.1 Εισαγωγή

Τα αποτελέσματα χωρίζονται με βάση τα δεδομένα που χρειάστηκαν ώστε να παρθούν οι μετρήσεις. Τα μεγέθη που εξετάζονται είναι το μέσο τετραγωνικό σφάλμα, ο αριθμός των Ψευδώς Θετικά(False Positive) και Ψευδώς Αρνητικά(False Negative) καθώς και η συμπίεση. Σε κάθε περίπτωση συγκρίνεται το πλαίσιο που προβλέφτηκε σε αντιδιαστολή με το πλαίσιο το οποίο πραγματικά ήταν το επόμενο.

4.1.1 Μέσο τετραγωνικό σφάλμα

Υπολογίζοντας πρώτα την διαφορά των δύο πλαισίων, δηλαδή του πλαισίου που ο αλγόριθμος προέβλεψε ως το πιο πιθανό να είναι το επόμενο πλαίσιο αλλά και του πλαισίου που όντως ήταν το επόμενο, μετέπειτα η διαφορά αυτή υψώνεται στο τετράγωνο. Το μέσο τετραγωνικό σφάλμα (MSE) είναι ο μέσος όρος της υψωμένης στο τετράγωνο διαφοράς των δύο πλαισίων.

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2.$$

Εν συνεχεία, βρίσκεται ο μέσος όρος του τετραγωνικού σφάλματος για κάθε μεταβλητή και τέλος ο μέσος όρος όλων των προηγούμενων που αντικατοπτρίζει το μέσο τετραγωνικό σφάλμα όλων των μεταβλητών.

4.1.2 Ψευδώς αρνητικά

Ορίζεται κάθε φορά μία σταθερά η οποία προσδιορίζει πότε θεωρούμε ότι η προβλεπόμενη τιμή μίας μεταβλητής στο προβλεπόμενο πλαίσιο έχει παρουσιάσει διαφορά σε σχέση με την αντίστοιχη τιμή αυτής της μεταβλητής από το τελευταίο πλαίσιο που πραγματικά ελήφθη. Αυτή η σταθερά είναι συνήθως στο 20-40% αλλά εξαρτάται και από το πείραμα και την εφαρμογή. Αφού υπολογιστεί το σχετικό σφάλμα τότε αυτό συγκρίνεται με την παραπάνω σταθερά και αν είναι μεγαλύτερο τότε θέτουμε 1 σε έναν δυαδικό πίνακα και αν όχι τότε εισάγεται το 0. Σε κάθε περίπτωση το σχετικό σφάλμα υπολογίζεται σε όλο το πλαίσιο και όχι σε κάθε επιμέρους μεταβλητή. Αν το σχετικό σφάλμα είναι μικρότερο από την σταθερά που έχει οριστεί, τότε το πλαίσιο δεν θα αποσταλεί. Αν όμως όταν έρθει το επόμενο πλαίσιο και επαναυπολογιστεί το σχετικό σφάλμα, αυτό είναι μεγαλύτερο από την σταθερά, τότε θα έπρεπε αυτό το πλαίσιο να είχε αποσταλεί άρα εκεί έχουμε ένα ψευδώς αρνητικό (false negative). Τα false negative έχουν μεγάλη σημασία καθώς αποτελούν χάσιμο δεδομένων που ποτέ δεν αποστάλθηκαν.

4.1.3 Ψευδώς θετικά

Είναι η αντίθετη περίπτωση από το false negative και συμβαίνει όταν ενώ έχουμε υπολογίσει το σχετικό σφάλμα ως 1, δηλαδή μεγαλύτερο από την σταθερά που έχει οριστεί ως το όριο του σφάλματος και όταν έρθει το επόμενο πλαίσιο και επαναυπολογιστεί το σχετικό σφάλμα, αυτό είναι μικρότερο από την σταθερά, τότε θα έπρεπε αυτό το πλαίσιο να μην είχε αποσταλεί άρα εκεί έχουμε ένα ψευδώς θετικό (false positive).

4.1.4 Συμπίεση

Ως συμπίεση ορίζεται ο συνολικός αριθμός των πλαισίων προς τον αριθμό των πλαισίων που αποστάλθηκαν από τον αλγόριθμο. Έτσι αν για παράδειγμα από τα 10 πλαίσια αποστάλθηκαν τα 5, υπάρχει συμπίεση $2 = 200\%$. Δεν λαμβάνεται υπόψη η συμπίεση λόγω της μείωσης της διάστασης των πλαισίων.

4.2 Δεδομένα από τον τομέα ναυσιπλοΐας

Τα πρώτα δεδομένα που χρησιμοποιήθηκαν είναι παρμένα από αισθητήρες οργάνων στον τομέα της ναυσιπλοΐας (maritime field). Τα πραγματικά αυτά δεδομένα αποτελούν μία χρονοσειρά όπου το κάθε πλαίσιο αυτής έχει διαφορά χρονική πέντε δευτερόλεπτα από το επόμενο. Οι μεταβλητές που χρησιμοποιήθηκαν ήταν 29 ενώ ο χρονικός προσδιορισμός δεν χρησιμοποιήθηκε σαν μεταβλητή. Οι περισσότερες τιμές αφορούν το μηχανικό σκέλος του πλοίου και έχουν να κάνουν με την ροπή, το βάθος, την ταχύτητα του αέρα και λοιπές παρόμοιες περιπτώσεις.

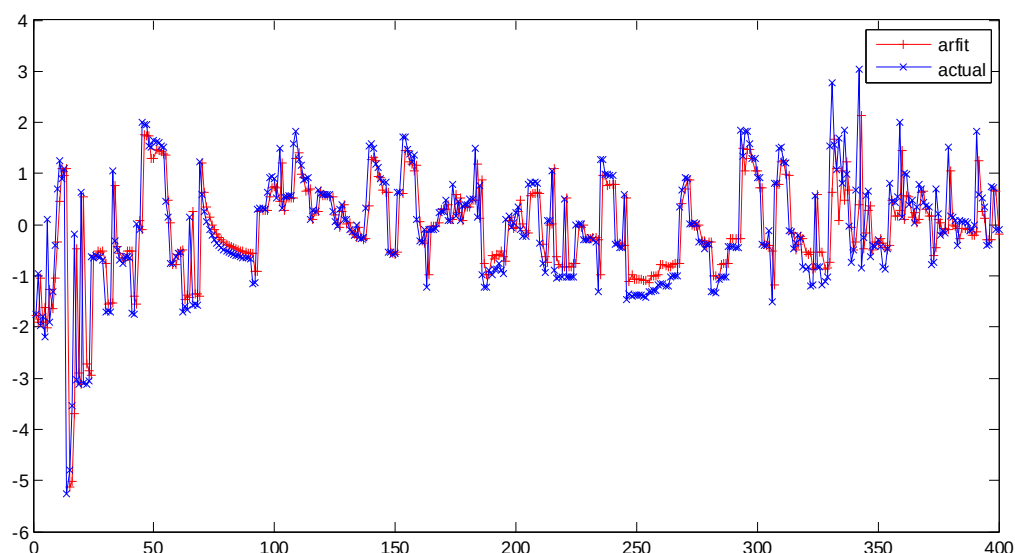
Τα δεδομένα αυτά αποθηκεύτηκαν σε έναν πίνακα $n \times 29$ όπου n είναι τα διαφορετικά πλαίσια και τροφοδοτήθηκαν στον αλγόριθμο.

Σε όλα τα σχήματα ο άξονας των x απεικονίζει τα δείγματα των μετρήσεων, κάθε ένα από τα οποία έρχεται σε διαδοχική χρονική στιγμή.

4.2.1 Συνολικό ποσοστό διακύμανσης

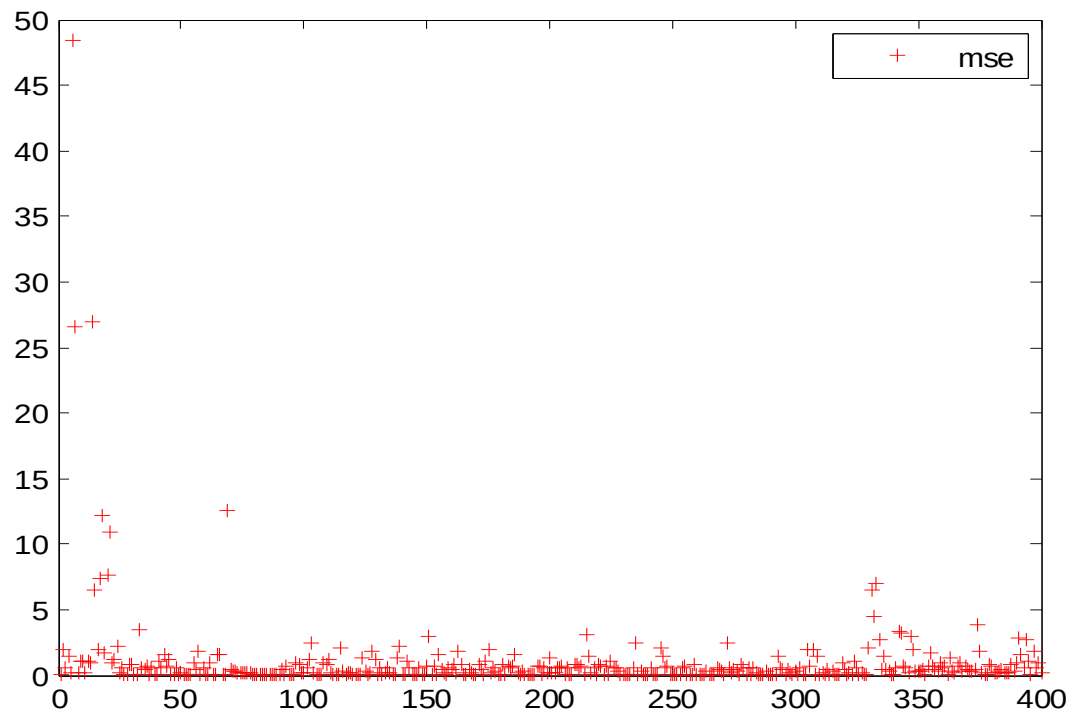
Εφαρμόζοντας τον αλγόριθμο PCA επιλέγονται m μεταβλητές οι οποίες εξηγούν την συνολική διακύμανση των αρχικών μεταβλητών σε φθίνουσα σειρά. Έτσι αν επιλεγθεί το 100% της διακύμανσης, θα επιλεγθούν και το 100% των μεταβλητών, δηλαδή στην παρούσα περίπτωση 29 μεταβλητές. Για διαφορετικά ποσοστά ολικής διακύμανσης εξάγεται διαφορετικός αριθμός m . Στα συγκεκριμένα δεδομένα για 60% της συνολικής διακύμανσης χρειάζονται 8 μεταβλητές, για 75% χρειάζονται 12 μεταβλητές και για 90% χρειάζονται 17 μεταβλητές.

Εφαρμόζουμε τον αλγόριθμο PCA σε 400 πλαίσια και εν συνεχεία προβλέπονται τα επόμενα πλαίσια με βάση τον αλγόριθμο ARfit για τα επόμενα 400 πλαίσια.



Σχήμα 2: Διαφορά μεταξύ προβλεπόμενης τιμής και επόμενης τιμής χρησιμοποιώντας τον ARFIT και 60% της διακύμανσης (πρώτη μεταβλητή)

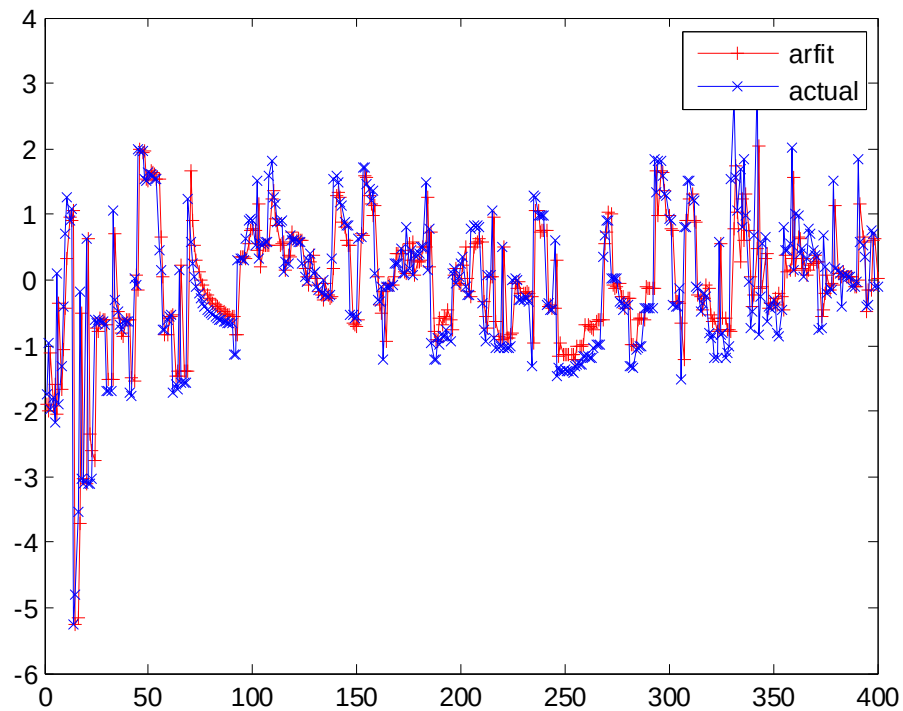
Στο παραπάνω σχήμα φαίνεται ότι για την πρώτη μεταβλητή από τις 8 συνολικά, η οποία είναι αυτή η οποία προσφέρει την μεγαλύτερη επεξήγηση της συνολικής διακύμανσης, η τιμή που προβλέπεται έχει μικρή διαφορά από τις προηγούμενες και έχει μικρή διαφορά από την τιμή που είναι η επόμενη. Η σύγκριση αυτή προφανώς γίνεται αφότου υπάρξει η πληροφορία του επόμενου πλαισίου. Με βάση την διαφορά όλων των μεταβλητών του πλαισίου βρίσκεται και το μέσο τετραγωνικό σφάλμα και παρουσιάζεται ο μέσος όρος αυτού από όλες τις μεταβλητές. Ο συνολικός μέσος όρος του σφάλματος είναι 0.96626.



**Σχήμα 3: Μέσος όρος του μέσου τετραγωνικού σφάλματος
χρησιμοποιώντας τον ARFIT και 60% της διακύμανσης**

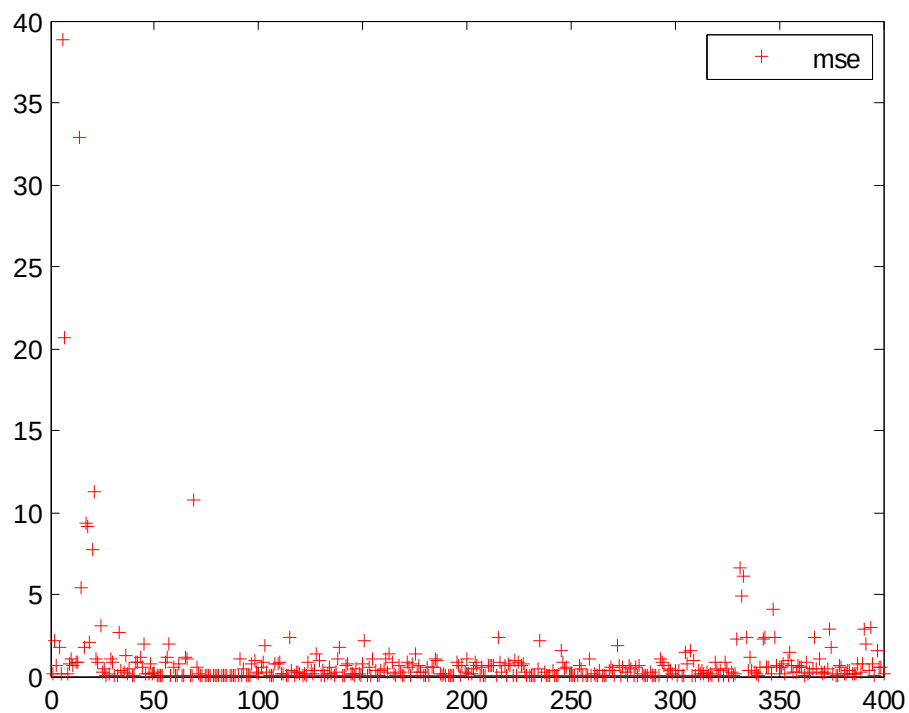
Φαίνεται ότι σε γενικά επίπεδα ο μέσος όρος του μέσου τετραγωνικού σφάλματος είναι μικρός εκτός από τις περιπτώσεις όπου έχουμε ακραίες εναλλαγές τιμών όπου είναι λογικό να έχουμε μεγάλο σφάλμα μεταξύ της προβλεπόμενης τιμής και της τιμής που ήταν πραγματικά η επόμενη.

Εφαρμόζοντας τον αλγόριθμο PCA σε 400 πλαίσια με το 75% της συνολικής διακύμανσης χρειάζονται 12 μεταβλητές.



Σχήμα 4: Διαφορά μεταξύ προβλεπόμενης τιμής και επόμενης τιμής χρησιμοποιώντας τον ARFIT και 75% της διακύμανσης (πρώτη μεταβλητή)

Ο συνολικός μέσος όρος του σφάλματος είναι 0.89617.

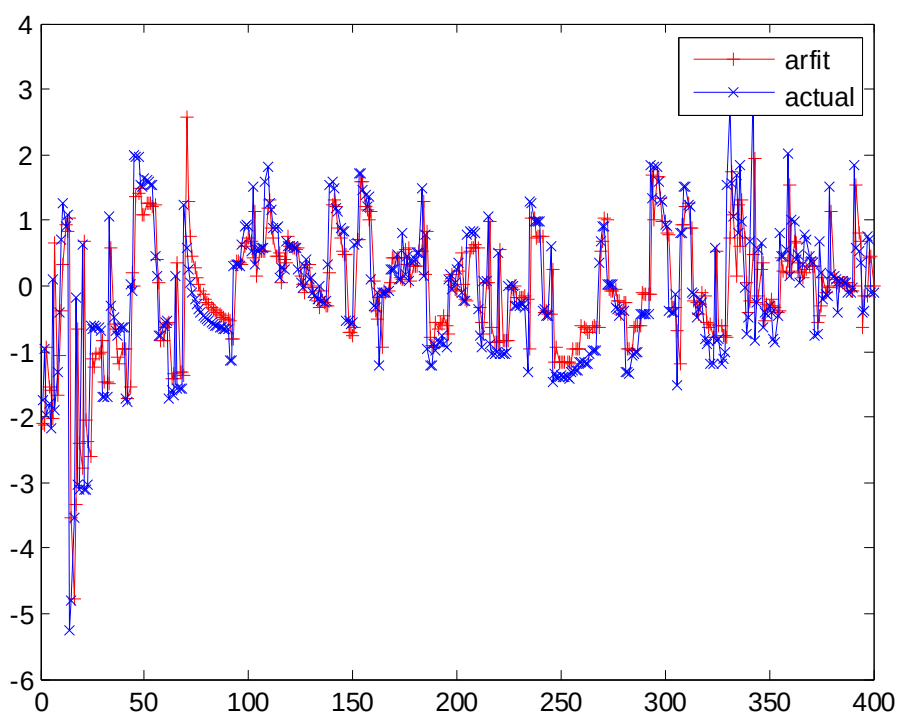


Σχήμα 5: Μέσος όρος του μέσου τετραγωνικού σφάλματος χρησιμοποιώντας τον ARFIT και 75% της διακύμανσης

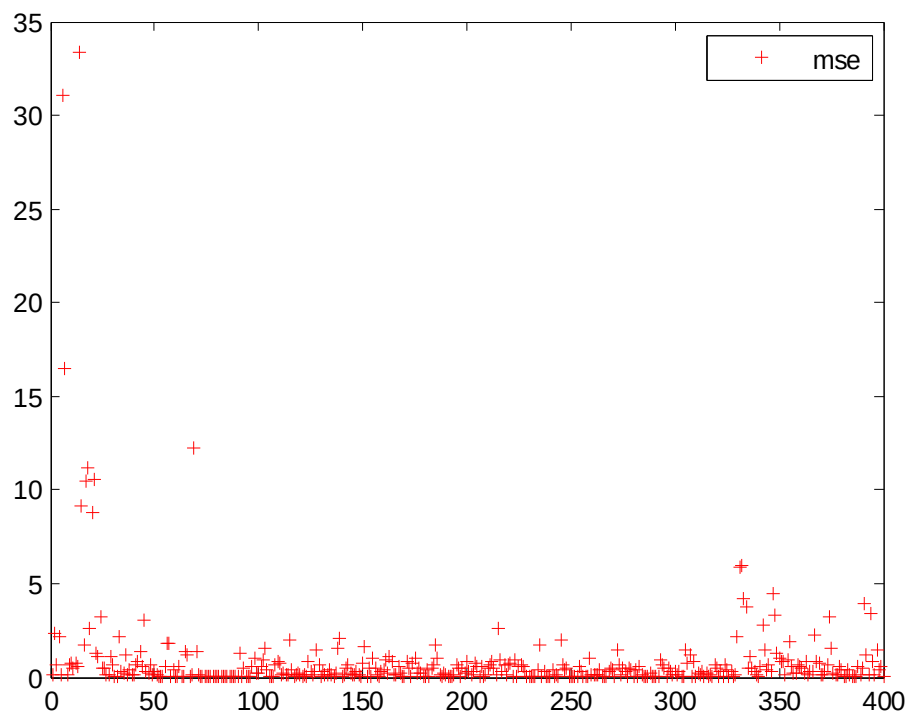
Οι διαφορές είναι μικρές από τα αποτελέσματα που υπήρχαν για το 60% της διακύμανσης και αυτό συμβαίνει γιατί πρόκειται για την ίδια μεταβλητή (Σχήμα 2 και Σχήμα 4). Επίσης ο μέσος όρος των σφαλμάτων είναι μικρότερος γιατί επειδή υπάρχει περισσότερη πληροφορία, ο αλγόριθμος ARfit είναι σε θέση να προβλέψει με μεγαλύτερη ακρίβεια τις επόμενες τιμές.

Εφαρμόζοντας τον αλγόριθμο PCA σε 400 πλαίσια με το 90% της συνολικής διακύμανσης χρειάζονται 17 μεταβλητές.

Ο συνολικός μέσος όρος του σφάλματος είναι 0.88854.



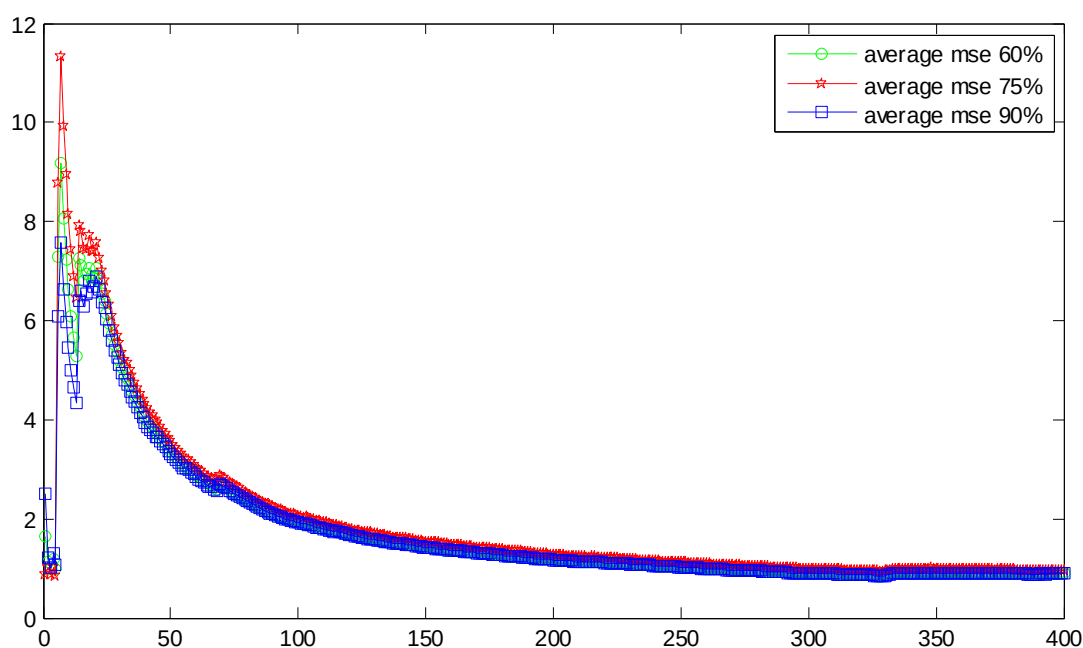
Σχήμα 6: Διαφορά μεταξύ προβλεπόμενης τιμής και επόμενης τιμής χρησιμοποιώντας τον ARFIT και 90% της διακύμανσης (πρώτη μεταβλητή)



Σχήμα 7: Μέσος όρος του μέσου τετραγωνικού σφάλματος χρησιμοποιώντας τον ARFIT και 90% της διακύμανσης

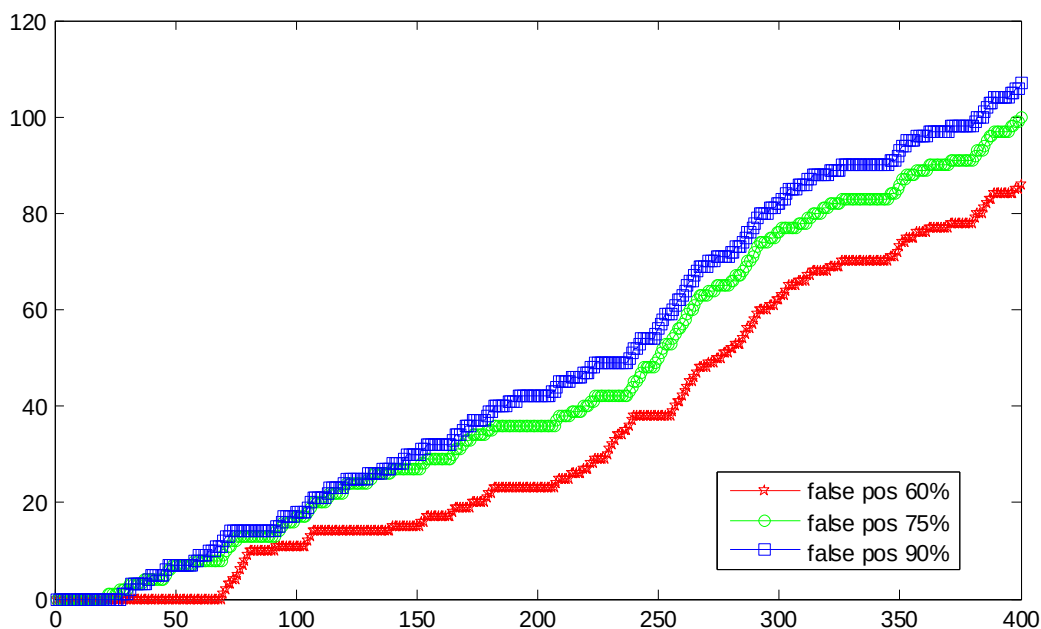
Ο συνολικός μέσος όρος του σφάλματος είναι μικρότερος από τις δύο άλλες περιπτώσεις (60% και 75% της συνολικής διακύμανσης) και αυτό εξηγείται όπως και στα σχήματα 4 και 5.

Οι υπόλοιπες μετρικές, φαίνονται αθροιστικά στα παρακάτω σχήματα.

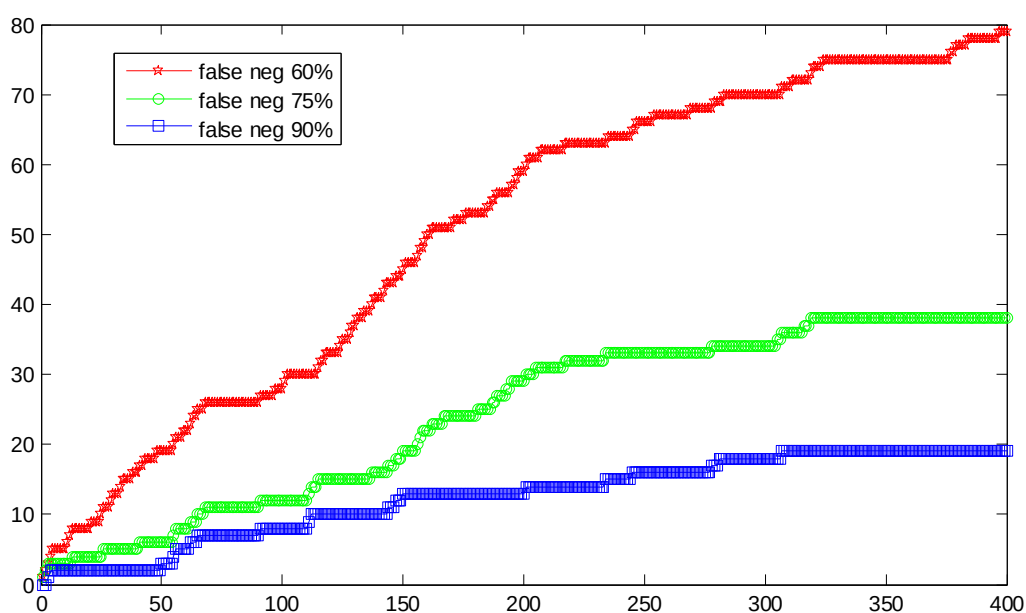


Σχήμα 8: Μέσος όρος για το μέσο τετραγωνικό σφάλμα

Στο Σχήμα 8 γίνεται εμφανές πως το χαμηλότερο μέσο τετραγωνικό σφάλμα υπάρχει στον αλγόριθμο ARfit που χρησιμοποιεί το 90% της διακύμανσης και αυτό είναι λογικό με βάση τις παραπάνω επεξηγήσεις. Επίσης φαίνεται πως στην αρχή υπάρχει υψηλός μέσος όρος λόγω συνεχών διακυμάνσεων ενώ στην συνέχεια φαίνεται πως ο μέσος όρος για το μέσο τετραγωνικό σφάλμα συγκλίνει προς μία συγκεκριμένη τιμή.

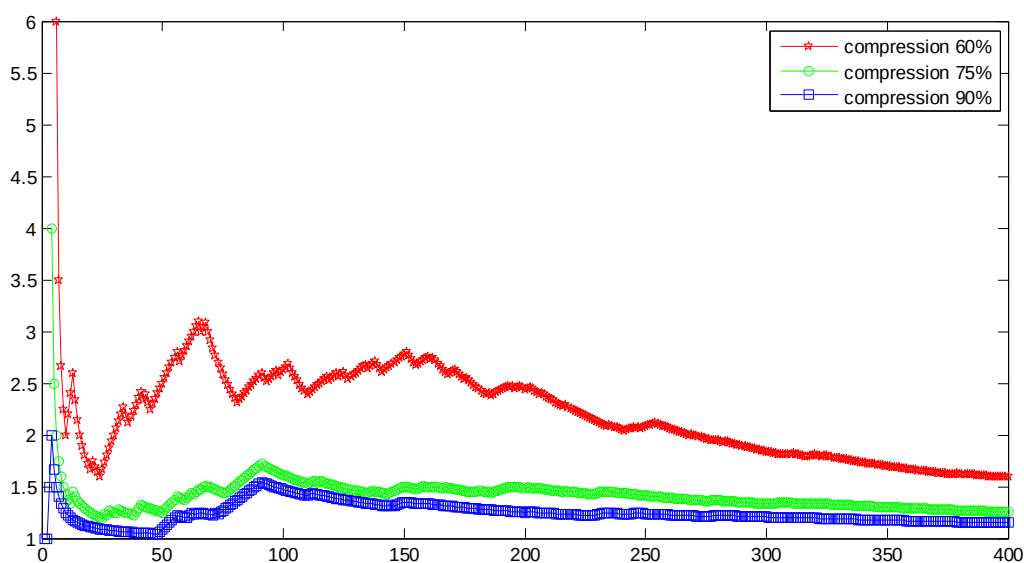


Σχήμα 9: False positive



Σχήμα 10: False negative

Στα παραπάνω δύο σχήματα γίνεται εμφανές πως καθώς αυξάνεται το ποσοστό της διακύμανσης αυξάνονται τα false positive και μειώνονται τα false negative. Αυτό συμβαίνει γιατί καθώς αυξάνεται το ποσοστό της διακύμανσης που διαχειρίζεται ο αλγόριθμος ARfit παράλληλα αυξάνονται και οι μεταβλητές, άρα σε πλαίσια τα οποία εμφάνιζαν απόκλιση σε μερικές από τις μεταβλητές, έχοντας πιο πολλές μεταβλητές το συνολικό σφάλμα αυξάνεται επηρεασμένο και από τις υπόλοιπες μεταβλητές. Έτσι υπάρχουν περισσότερες φορές που ο αλγόριθμος αποφασίζει να στείλει το αντίστοιχο πλαίσιο σε σχέση με τον ίδιο αλγόριθμο με μειωμένη διακύμανση. Συνεπώς αφού αποστέλλονται περισσότερα πλαίσια θα υπάρχουν και περισσότερα false positive και λιγότερα false negative.

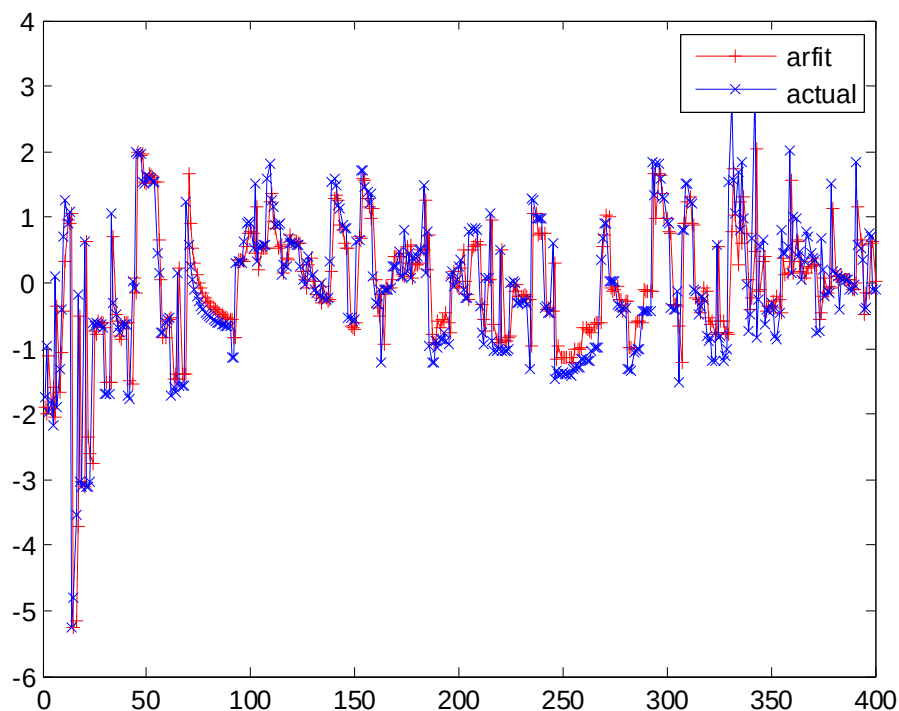


Σχήμα 11: Συμπίεση

Συνεπώς αφού περισσότερα πλαίσια αποστέλλονται, είναι λογικό ότι για μεγαλύτερη διακύμανση θα έχουμε μικρότερη συμπίεση αφού στο ίδιο χρονικό διάστημα θα έχουν αποσταλεί πιο πολλά δεδομένα.

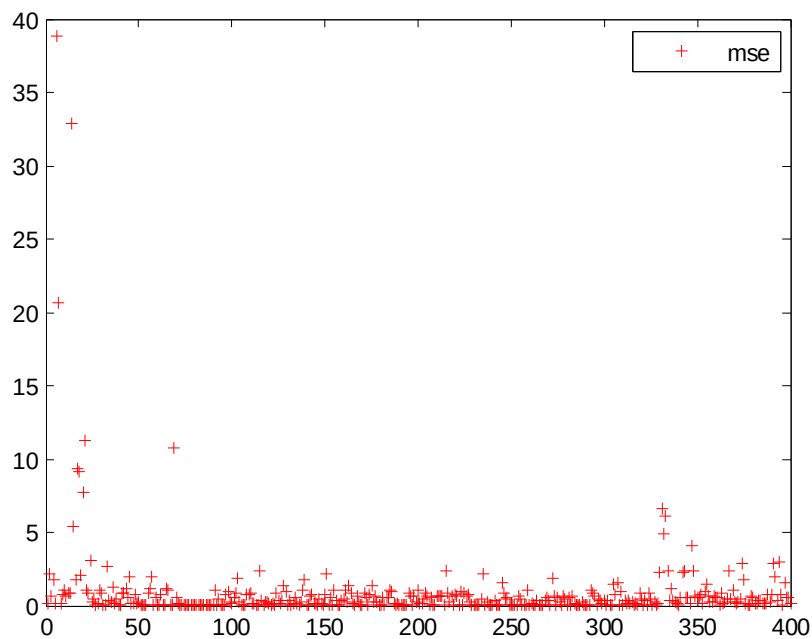
4.2.2 Ποσοστό σφάλματος

Η συνολική διακύμανση που θα εφαρμοστεί θα είναι 75% άρα στα παραπάνω δεδομένα αυτό ισοδυναμεί με 12 μεταβλητές. Για 12 μεταβλητές θα εξεταστούν οι προαναφερόμενες μετρικές για διαφορετικά όρια σφάλματος. Το όριο σφάλματος είναι το ποσοστό που καθορίζει πόσο μεγάλο θα είναι το σχετικό σφάλμα για να αποφασίσει ο αλγόριθμος ότι πρέπει να σταλθεί ένα πλαίσιο. Τα παρακάτω αποτελέσματα παρουσιάζονται για σχετικό σφάλμα 20%, 30% και 40%. Το σχετικό σφάλμα υπολογίζεται με βάση το προηγούμενο πραγματικό πλαίσιο και το πλαίσιο το οποίο προβλέπεται από τον εκάστοτε αλγόριθμο ως το πιο πιθανό για να ληφθεί την αμέσως επόμενη χρονική στιγμή.



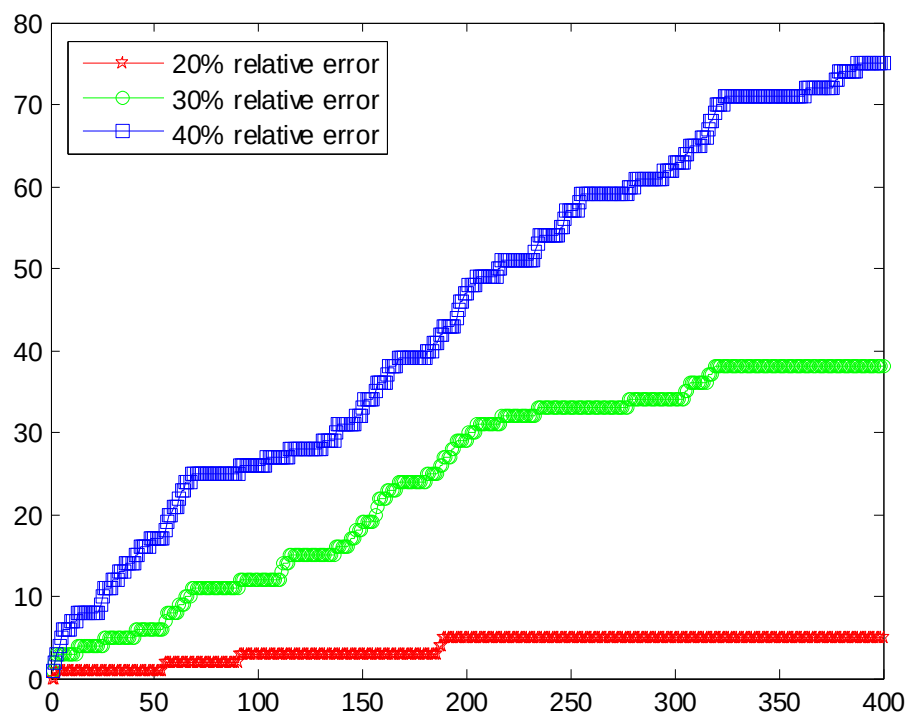
Σχήμα 12: Διαφορά μεταξύ των πραγματικών τιμών και των τιμών που προβλέπει ο ARFIT ανεξάρτητο οριακού σφάλματος

Ανεξάρτητα από το ποιο είναι το οριακό ποσοστό σφάλματος, η διαφορά μεταξύ των πραγματικών τιμών και των τιμών που προβλέπει ο ARfit είναι η ίδια και στις τρεις περιπτώσεις αφού δεν επηρεάζει καθόλου ούτε το ποια τιμή προβλέπεται ούτε το μέσο τετραγωνικό σφάλμα.



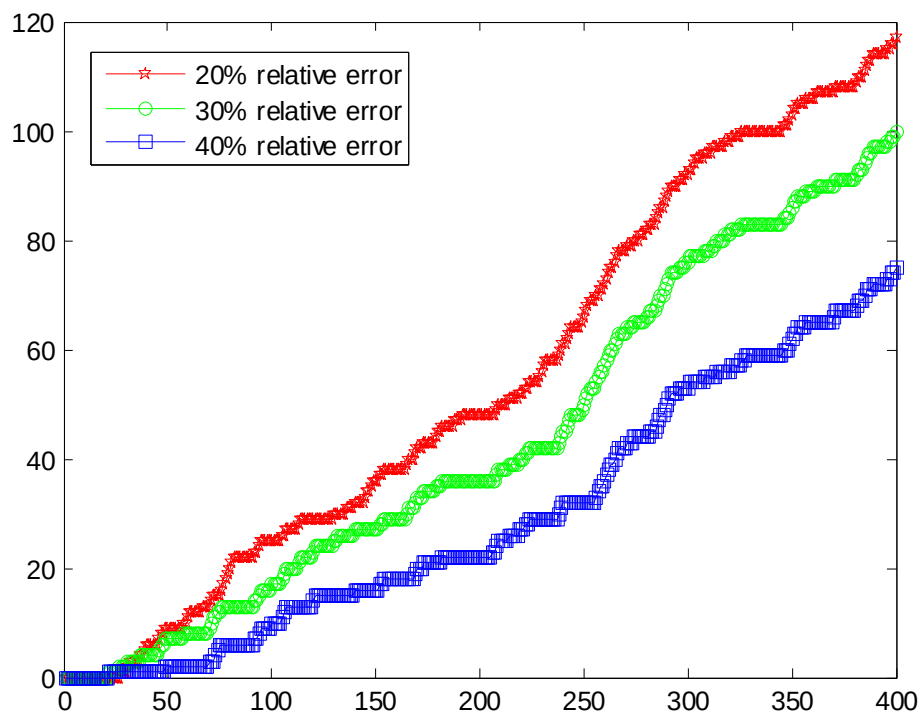
Σχήμα 13: Μέσο τετραγωνικό σφάλμα ανεξάρτητο οριακού σφάλματος

Αν και στα δύο παραπάνω σχήματα γίνεται εμφανές πως δεν παρουσιάζονται διαφορές για διαφορετικά οριακά σφάλματα, εκεί που παρουσιάζονται οι διαφορές είναι στα σχήματα των false positive, false negative καθώς και συμπίεσης.



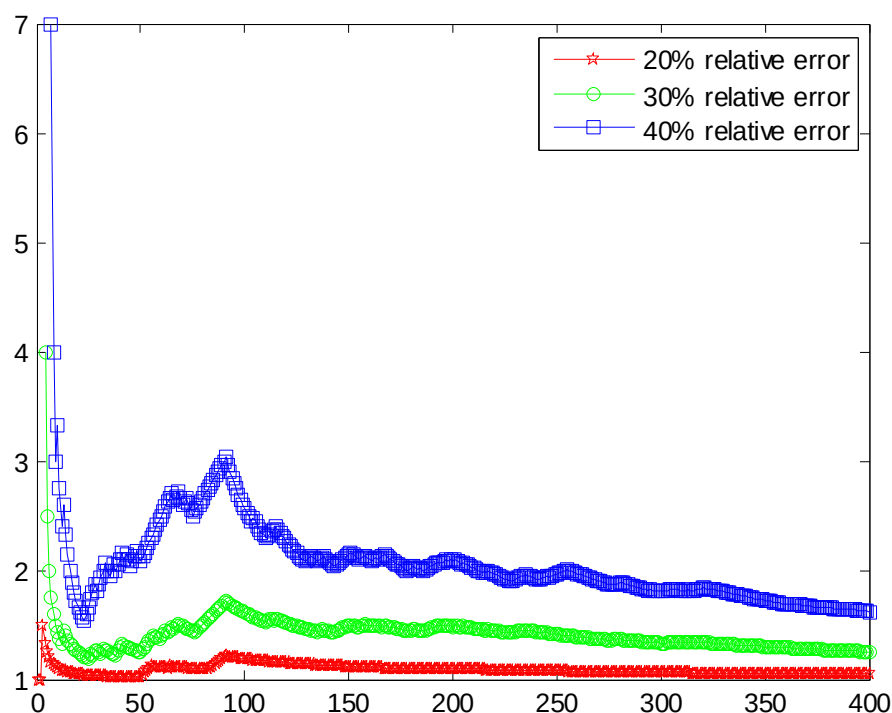
Σχήμα 14: False negative για διαφορετικά οριακά σχετικά σφάλματα

Όσο αυξάνεται το παράθυρο του σχετικού σφάλματος τόσο πιο πολλά πλαίσια θα υπάρχουν στα οποία ο αλγόριθμος θα αποφανθεί ότι δεν υπάρχει μεταβολή μεταξύ του προβλεπόμενου πλαισίου και του αμέσως προηγούμενου. Στο παραπάνω σχήμα φαίνεται πως όσο αυξάνουμε το οριακό σχετικό ποσοστιαίο σφάλμα, τόσο αυξάνονται και τα false negative αφού αυξάνονται οι περιπτώσεις όπου ο αλγόριθμος αποφασίζει ότι δεν υπάρχει μεταβολή μεταξύ των δύο συγκρινόμενων πλαισίων.



Σχήμα 15: false positive για διαφορετικά οριακά σχετικά σφάλματα

Όσο μειώνεται το παράθυρο του σχετικού σφάλματος τόσο πιο πολλά πλαίσια θα υπάρχουν στα οποία ο αλγόριθμος θα αποφανθεί ότι υπάρχει μεταβολή μεταξύ του προβλεπόμενου πλαισίου και του αμέσως προηγούμενου. Στο παραπάνω σχήμα φαίνεται πως όσο μειώνουμε το οριακό σχετικό ποσοστιαίο σφάλμα, τόσο αυξάνονται και τα false positive αφού αυξάνονται οι περιπτώσεις όπου ο αλγόριθμος αποφασίζει ότι υπάρχει μεταβολή μεταξύ των δύο συγκρινόμενων πλαισίων ενώ στην πραγματικότητα εκεί μπορεί να κάνει λάθος άρα να εμφανίζεται ένα false positive.

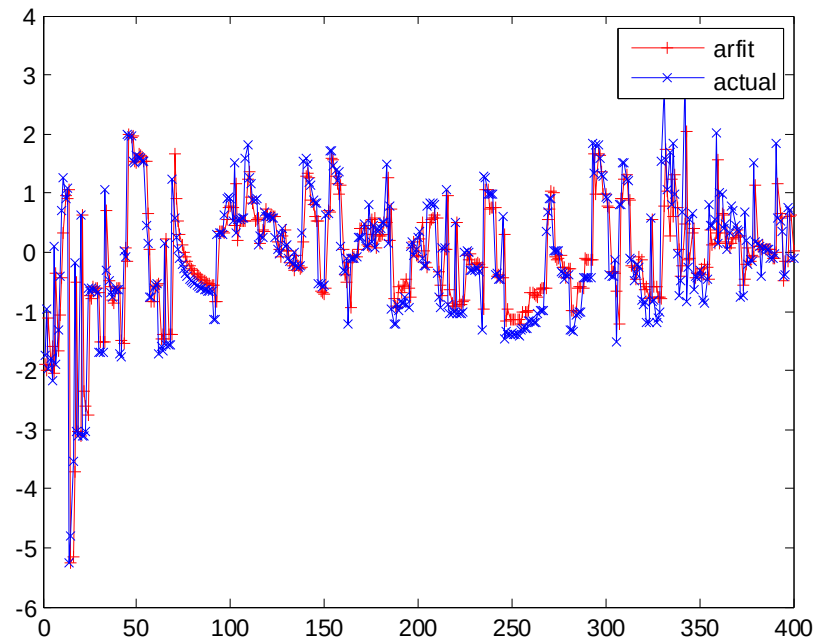


Σχήμα 16: Συμπίεση για διαφορετικά οριακά σχετικά σφάλματα

Παρομοίως, όσο πιο μεγάλο είναι το οριακό σχετικό ποσοστιαίο σφάλμα, τόσες πιο πολλές φορές ο αλγόριθμος αποφασίζει πως δεν υπάρχει μεταβολή μεταξύ των συγκρινόμενων πλαισίων άρα αποφασίζει να μην αποστείλει το πλαίσιο, με αποτέλεσμα έστω και αν αυτό αυξάνει όπως ήταν αναμενόμενο τα false negative, να έχουμε μεγαλύτερη συμπίεση αφού ο μεγαλύτερος παράγοντας στην συμπίεση είναι τα πλαίσια που αποφάσισε ο αλγόριθμος ότι δεν πρέπει να αποσταλούν.

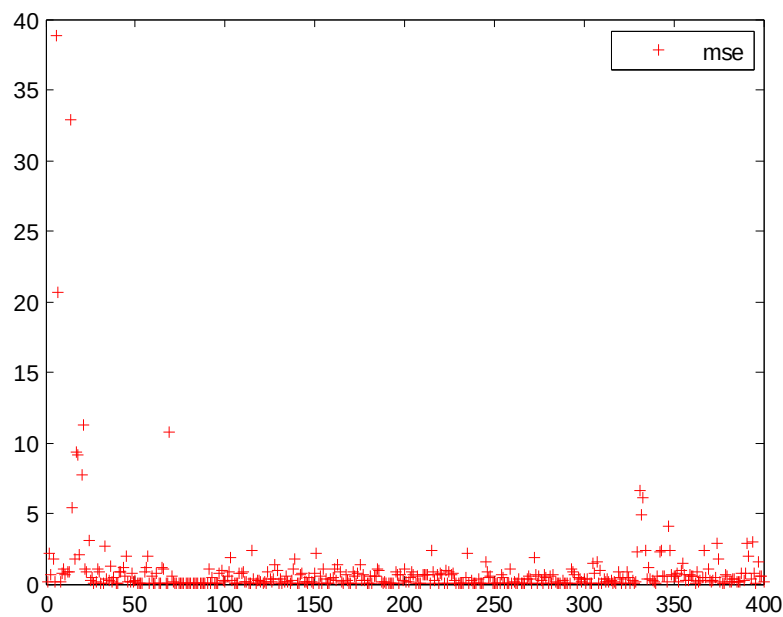
4.2.3 Σύγκριση με άλλες μεθόδους

Η συνολική διακύμανση που εφαρμόζεται είναι 75% άρα στα παραπάνω δεδομένα το οποίο ισοδυναμεί με 12 μεταβλητές. Για 12 μεταβλητές εξετάζονται οι προαναφερόμενες μετρικές για συγκεκριμένο ποσοστιαίο όριο σφάλματος το οποίο είναι 30%, σε σύγκριση με τις υπόλοιπες μεθόδους.

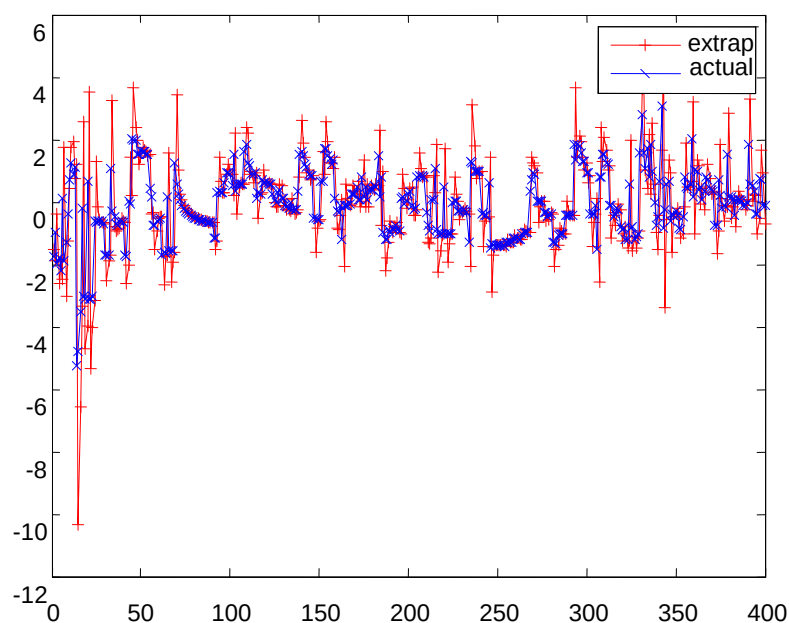


Σχήμα 17: Διαφορά προβλεπόμενης και πραγματικής τιμής (ARFIT)

Ο αλγόριθμος ARfit φαίνεται από το παραπάνω σχήμα ότι προβλέπει αρκετά καλά τα επόμενα πλαίσια καθώς οι διαφορές από τις πραγματικές τιμές δεν εμφανίζουν μεγάλες αποκλίσεις. Αυτό φαίνεται και από το σχήμα 18 όπου απεικονίζεται το μέσο τετραγωνικό σφάλμα και η μέγιστη τιμή είναι 39 ενώ ελάχιστες περιπτώσεις έχουν τόσο υψηλό σφάλμα (περιπτώσεις απότομης αλλαγής).

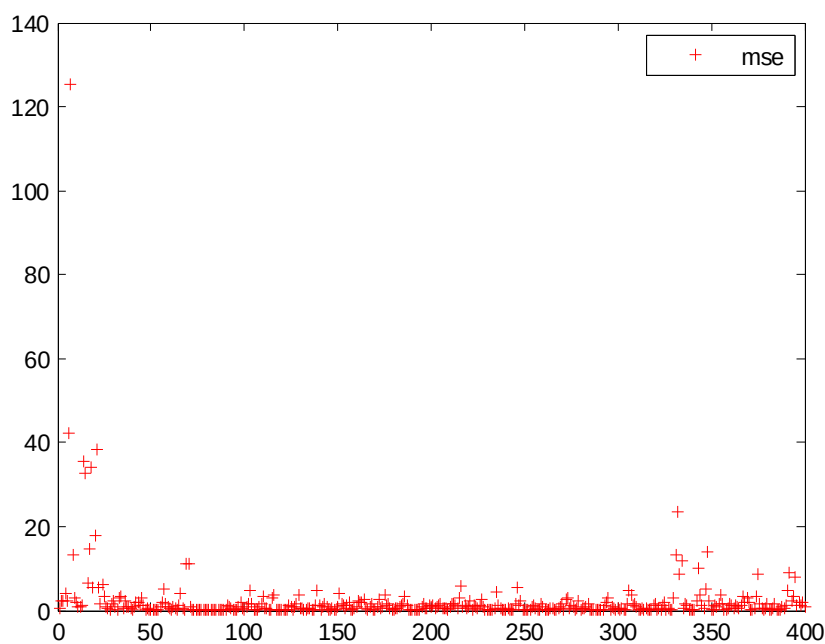


Σχήμα 18: Μέσο τετραγωνικό σφάλμα (ARFIT)

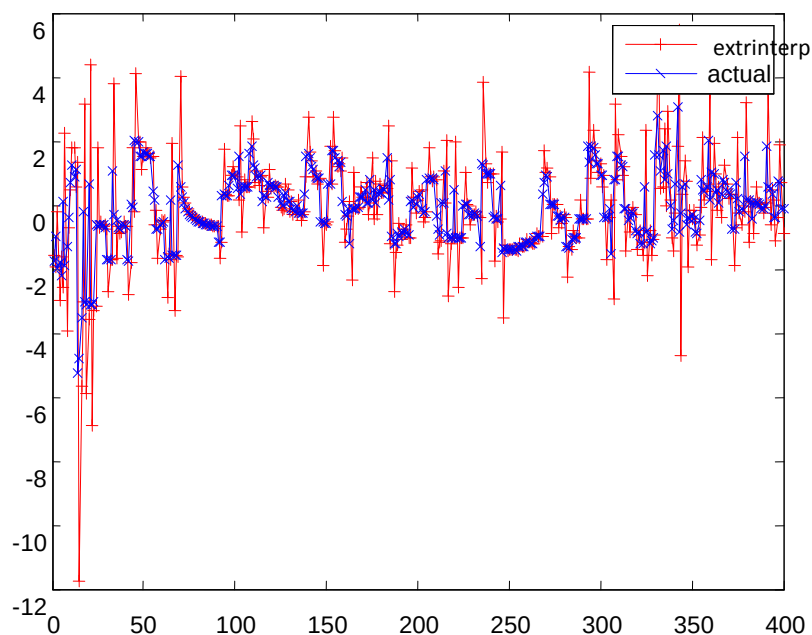


Σχήμα 19: Διαφορά προβλεπόμενης και πραγματικής τιμής (προεκβολή)

Ο αλγόριθμος προεκβολής φαίνεται από το παραπάνω σχήμα ότι δεν προβλέπει αρκετά καλά τα επόμενα πλαίσια καθώς οι διαφορές από τις πραγματικές τιμές σε μερικές περιπτώσεις εμφανίζουν μεγάλες αποκλίσεις. Αυτό φαίνεται και από το σχήμα 20 όπου απεικονίζεται το μέσο τετραγωνικό σφάλμα και η μέγιστη τιμή είναι 127 δηλαδή τρεις φορές χειρότερα αποτελέσματα από τον αλγόριθμο ARfit.

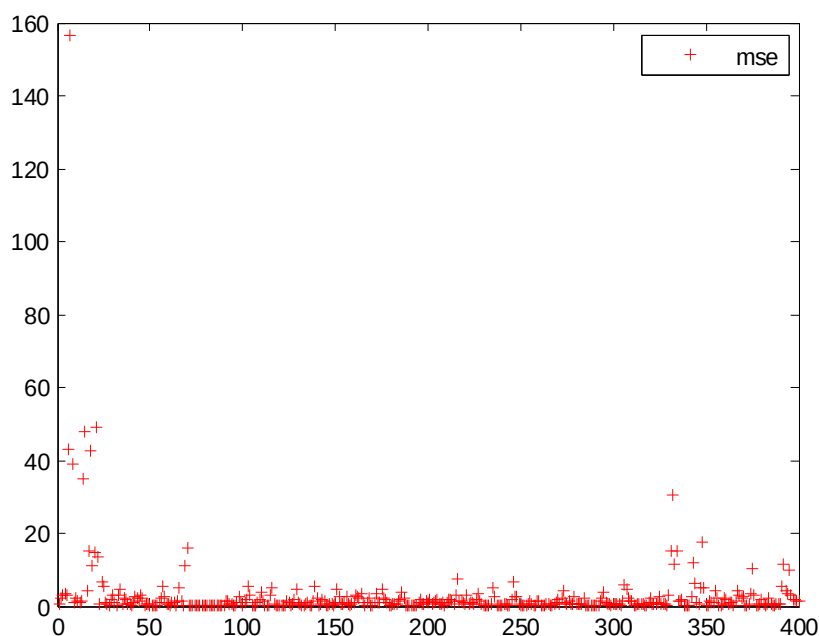


Σχήμα 20: Μέσο τετραγωνικό σφάλμα (extrapolation)

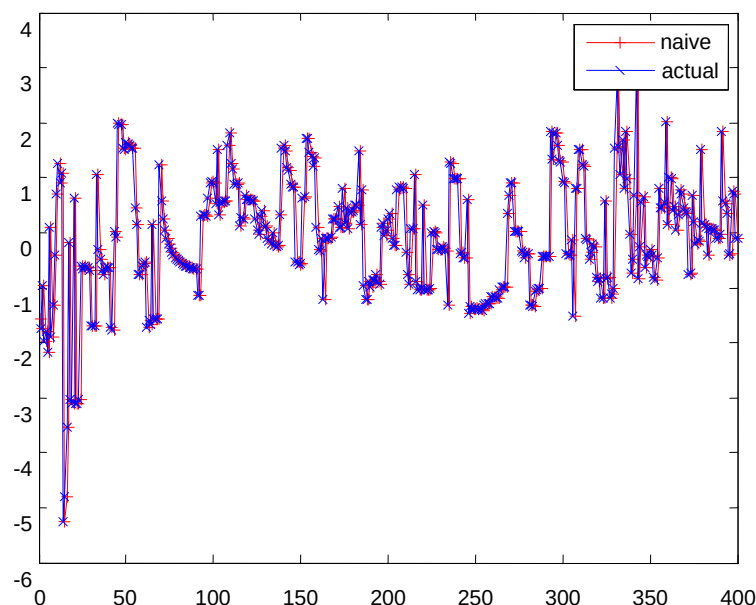


Σχήμα 21: Διαφορά προβλεπόμενης και πραγματικής τιμής (προεκβολή παρεμβολής)

Τα αποτελέσματα της προεκβολής παρεμβολής είναι όπως ήταν αναμενόμενο παρόμοια με αυτά των σχημάτων 19 και 20 και φαίνεται η αδυναμία των δύο αυτών μονομεταβλητών μεθόδων να μπορέσουν να αξιοποιήσουν τις πληροφορίες από όλες τις μεταβλητές για να προβλέψουν το επόμενο πλαίσιο.

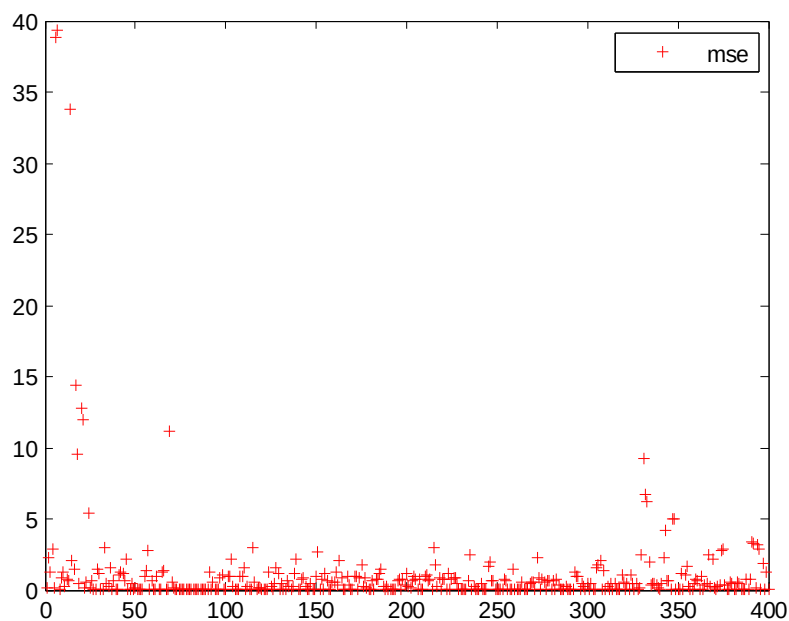


Σχήμα 22: Μέσο τετραγωνικό σφάλμα (προεκβολή παρεμβολής)



Σχήμα 23: Διαφορά προβλεπόμενης και πραγματικής τιμής (αφελής προσέγγιση)

Ο αλγόριθμος αφελοούς προσέγγισης φαίνεται από το παραπάνω σχήμα ότι προβλέπει αρκετά καλά τα επόμενα πλαίσια καθώς οι διαφορές από τις πραγματικές τιμές δεν εμφανίζουν μεγάλες αποκλίσεις. Αυτό φαίνεται και από το σχήμα 24 όπου απεικονίζεται το μέσο τετραγωνικό σφάλμα και η μέγιστη τιμή είναι 39 ενώ ελάχιστες περιπτώσεις έχουν τόσο υψηλό σφάλμα (περιπτώσεις απότομης αλλαγής).



Σχήμα 24: Μέσο τετραγωνικό σφάλμα (naive)

Πίνακας 1: Συνολικές μετρικές ARFIT

Μέσο σφάλμα	0.89617
Συμπύεση	1.2539
False positive	100
False negative	38

Πίνακας 2: Συνολικές μετρικές προεκβολής (προεκβολή)

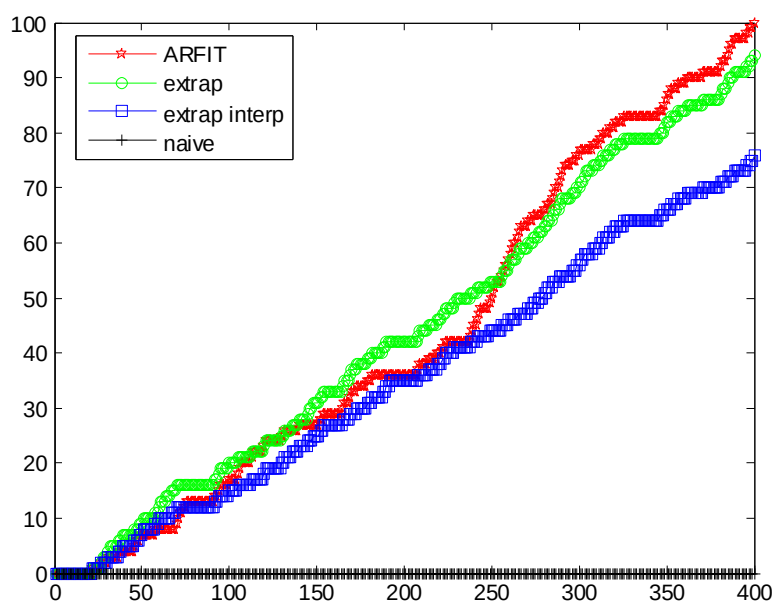
Μέσο σφάλμα	2.1983
Συμπύεση	1.3158
False positive	94
False negative	47

Πίνακας 3: Συνολικές μετρικές προεκβολής παρεμβολής (προεκβολή παρεμβολής)

Μέσο σφάλμα	2.7293
Συμπύεση	1.5625
False positive	76
False negative	77

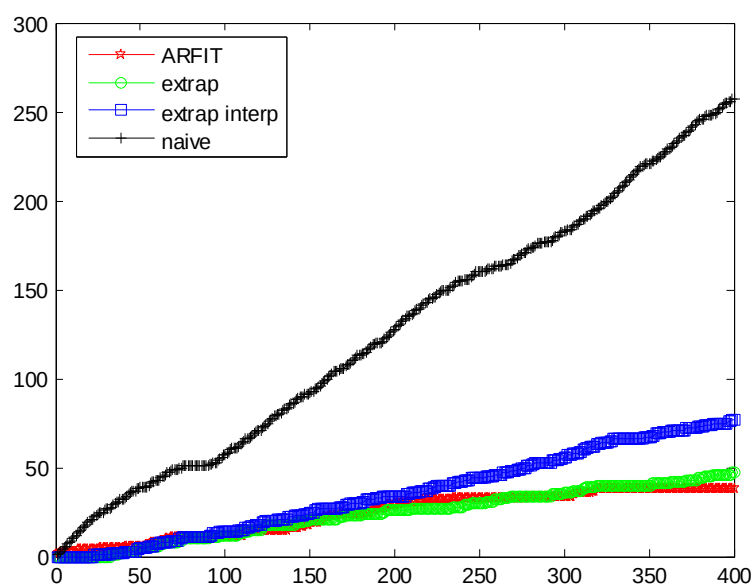
Πίνακας 4: Συνολικές μετρικές αφελής προσέγγισης

Μέσο σφάλμα	1.0443
Συμπύεση	-
False positive	0
False negative	257



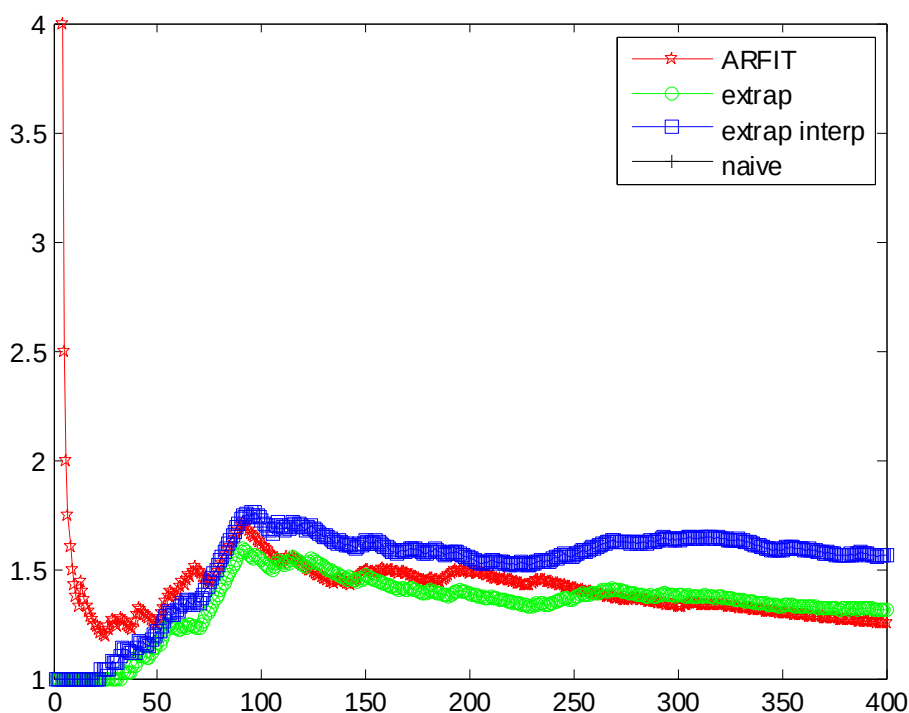
Σχήμα 25: Σύγκριση False Positive διαφορετικών μεθόδων

Στο Σχήμα 25 φαίνεται πως ο αλγόριθμος προεκβολής και ARfit παρουσιάζουν πολύ κοντινά αποτελέσματα ως προς τον αριθμό των false positive. Ο αλγόριθμος αφελής προσέγγισης έχει μηδενικό αποτέλεσμα αφού επειδή θεωρεί ότι η επόμενη τιμή θα είναι ίδια με την προηγούμενη, άρα έχει μηδενική διαφορά και συνεπώς σε καμία περίπτωση δεν αποφασίζει να στείλει το προβλεπόμενο πλαίσιο, το οποίο είναι ο λόγος για τα τόσα πολλά false negative (Σχήμα 26).



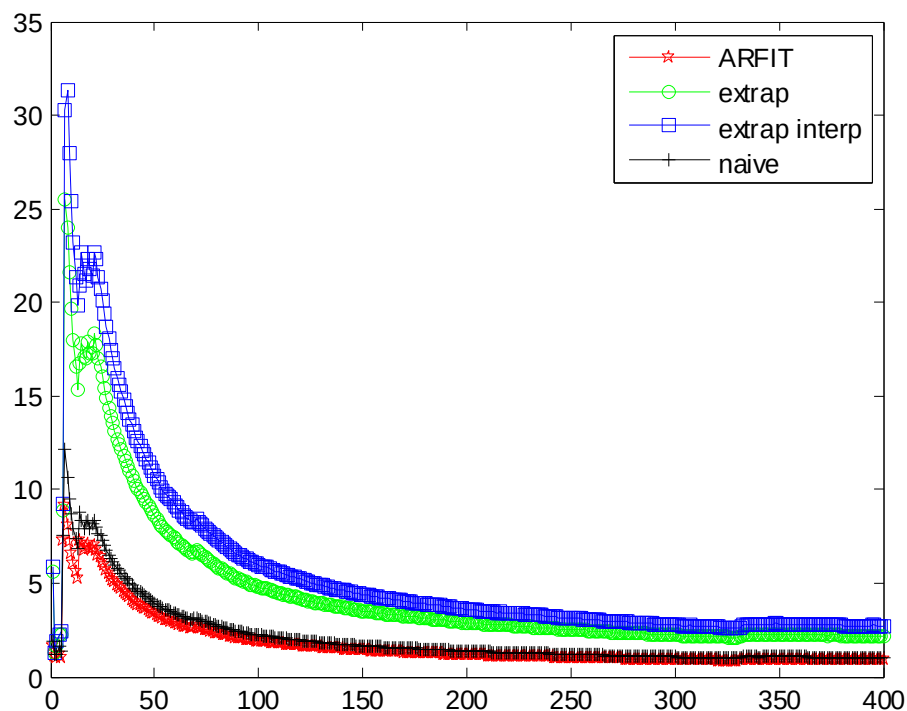
Σχήμα 26: Σύγκριση False Negative διαφορετικών μεθόδων

Επιπλέον στο σχήμα 26 φαίνεται και πως ο ARfit χρησιμοποιώντας τις πληροφορίες από πολλές μεταβλητές αντί για μόνο μία, καταφέρνει να μειώσει τα false negative σε σχέση με τους υπόλοιπους αλγόριθμους και έτσι έχουμε την μικρότερη απώλεια πληροφορίας που είναι από τα σημαντικότερα μεγέθη.



Σχήμα 27: Σύγκριση συμπίεσης διαφορετικών μεθόδων

Στο σχήμα 27 βλέπουμε πως η συμπίεση έχει κοντινά αποτελέσματα για τον ARfit και τον αλγόριθμο προεκβολής καθώς έχουν ίδιο αριθμό false negative.



Σχήμα 28: Σύγκριση μέσου όρου τετραγωνικού σφάλματος διαφορετικών μεθόδων

Λόγω της σωστής αξιοποίησης της πληροφορίας από διαφορετικές μεταβλητές ο αλγόριθμος ARfit επιτυγχάνει να μειώσει τον μέσο όρο του μέσου τετραγωνικού σφάλματος μεταξύ των πλαισίων που προβλέφθηκαν και των πλαισίων που πραγματικά ήταν τα επόμενα στην χρονοσειρά των δεδομένων.

4.3 SML2010

Η δεύτερη σειρά δεδομένων που χρησιμοποιήθηκαν είναι παρμένη από αισθητήρες οργάνων σε ένα σπίτι. Τα πραγματικά αυτά δεδομένα αποτελούν μία χρονοσειρά όπου το κάθε πλαίσιο αυτής έχει διαφορά χρονική 1 λεπτό από το επόμενο.

Τα δεδομένα αυτά είναι διαθέσιμα στην ιστοσελίδα:

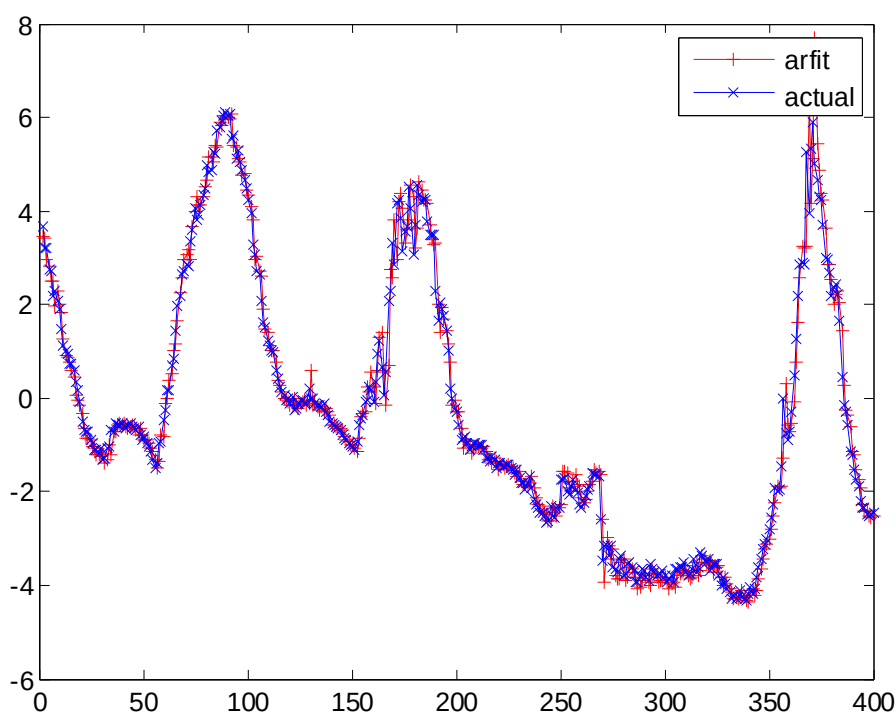
<https://archive.ics.uci.edu/ml/datasets/SML2010>

Οι μεταβλητές που χρησιμοποιήθηκαν ήταν 24 ενώ οι χρονικοί προσδιορισμοί δεν χρησιμοποιήθηκαν σαν μεταβλητές. Οι τιμές αφορούν επίπεδα θερμοκρασίας σε διαφορετικά δωμάτια, επίπεδα διοξειδίου του άνθρακα, επίπεδα εξωτερικής θερμοκρασίας και ηλιοφάνειας όπως και την ταχύτητα του αέρα. Άρα λόγω αυτών των μεταβλητών δεν υπάρχουν απότομες αλλαγές στις τιμές.

Τα δεδομένα αυτά αποθηκεύτηκαν σε έναν πίνακα $n \times 24$ όπου n είναι τα διαφορετικά πλαίσια και τροφοδοτήθηκαν στον αλγόριθμο.

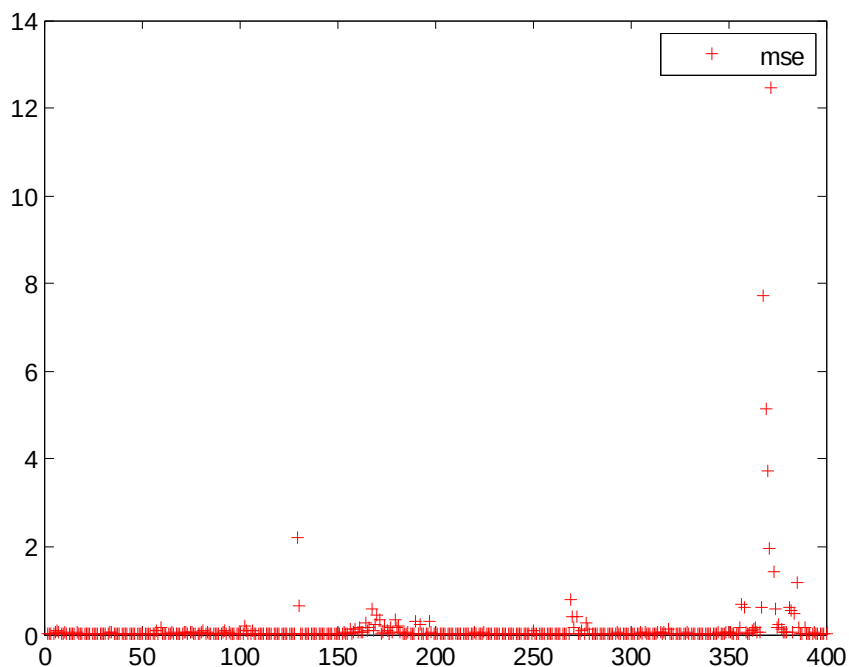
4.3.1 Σύγκριση μεθόδων

Η συνολική διακύμανση που εφαρμόζεται είναι 95% στα παραπάνω δεδομένα το οποίο ισοδυναμεί με 8 μεταβλητές. Αυτό συμβαίνει γιατί πολλές μεταβλητές εξαρτώνται άμεσα από άλλες. Για 8 μεταβλητές εξετάζονται οι προαναφερόμενες μετρικές για συγκεκριμένο ποσοστιαίο όριο σφάλματος το οποίο είναι 30%, σε σύγκριση με τις υπόλοιπες μεθόδους.

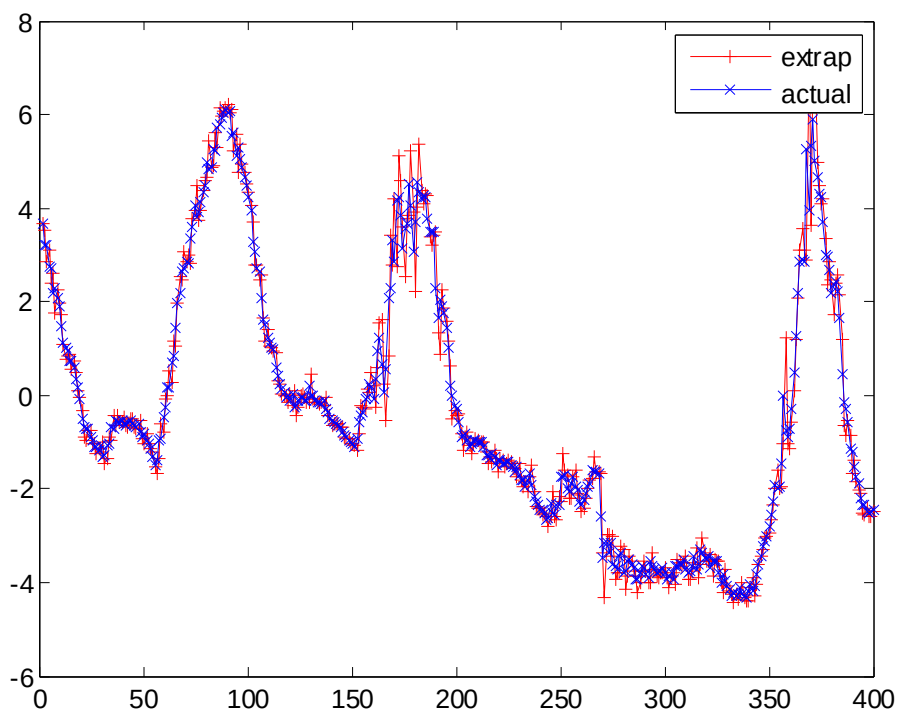


Σχήμα 29: Διαφορά προβλεπόμενης και πραγματικής τιμής (ARFIT - SML2010)

Ο αλγόριθμος ARfit φαίνεται από το παραπάνω σχήμα ότι προβλέπει αρκετά καλά τα επόμενα πλαίσια καθώς οι διαφορές από τις πραγματικές τιμές δεν εμφανίζουν μεγάλες αποκλίσεις. Αυτό φαίνεται και από το σχήμα 30 όπου απεικονίζεται το μέσο τετραγωνικό σφάλμα.



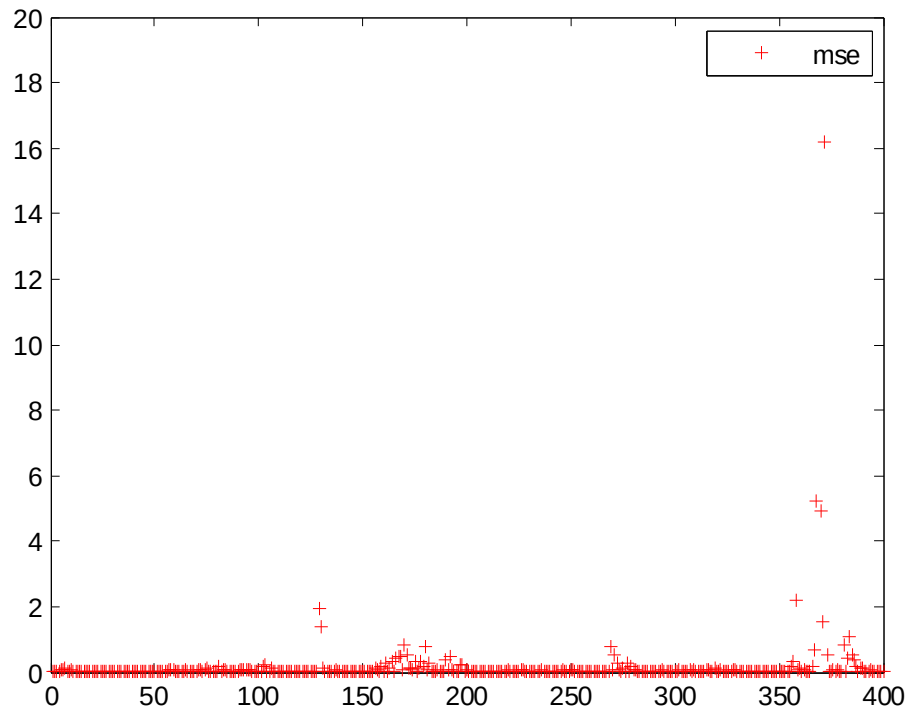
Σχήμα 30: Μέσο τετραγωνικό σφάλμα (ARFIT - SML2010)



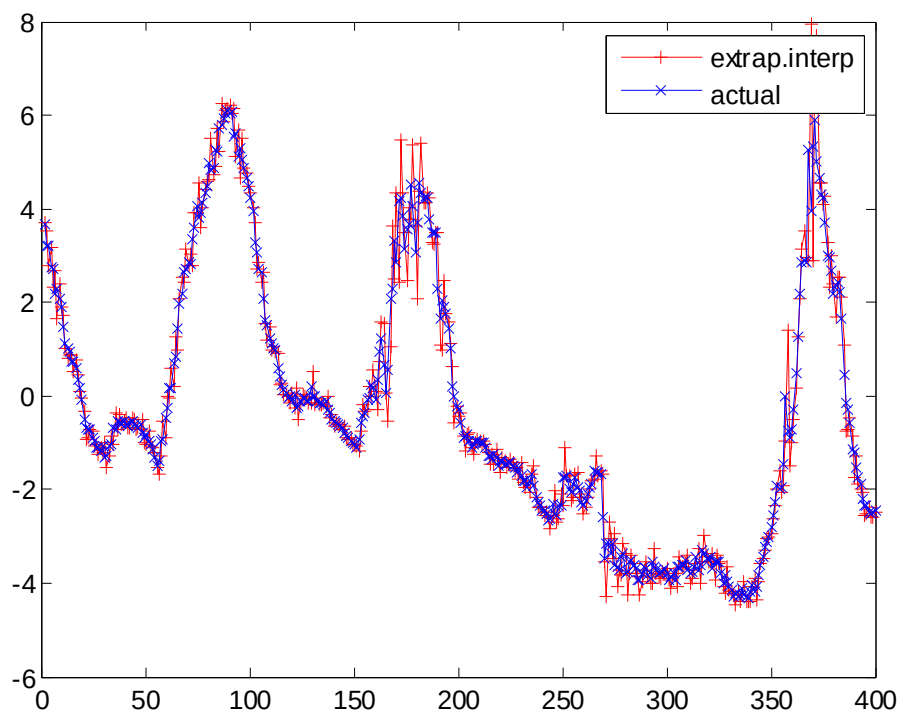
Σχήμα 31: Διαφορά προβλεπόμενης και πραγματικής τιμής (Προεκβολή - SML2010)

Ο αλγόριθμος προεκβολής φαίνεται από το παραπάνω σχήμα 31 ότι δεν προβλέπει αρκετά καλά τα επόμενα πλαίσια καθώς οι διαφορές από τις πραγματικές τιμές σε μερικές περιπτώσεις εμφανίζουν μεγάλες αποκλίσεις. Αυτό γίνεται σε περιπτώσεις όπου υπάρχει πχ ανοδικές τιμές και ο αλγόριθμος θεωρεί μέσω της προεκβολής ότι θα συνεχιστούν αυτές οι τιμές, ενώ η επόμενη τιμή είναι ίδιου μεγέθους ή μικρότερη με την προηγούμενη.

Αυτό φαίνεται και από το σχήμα 32 όπου απεικονίζεται το μέσο τετραγωνικό σφάλμα.

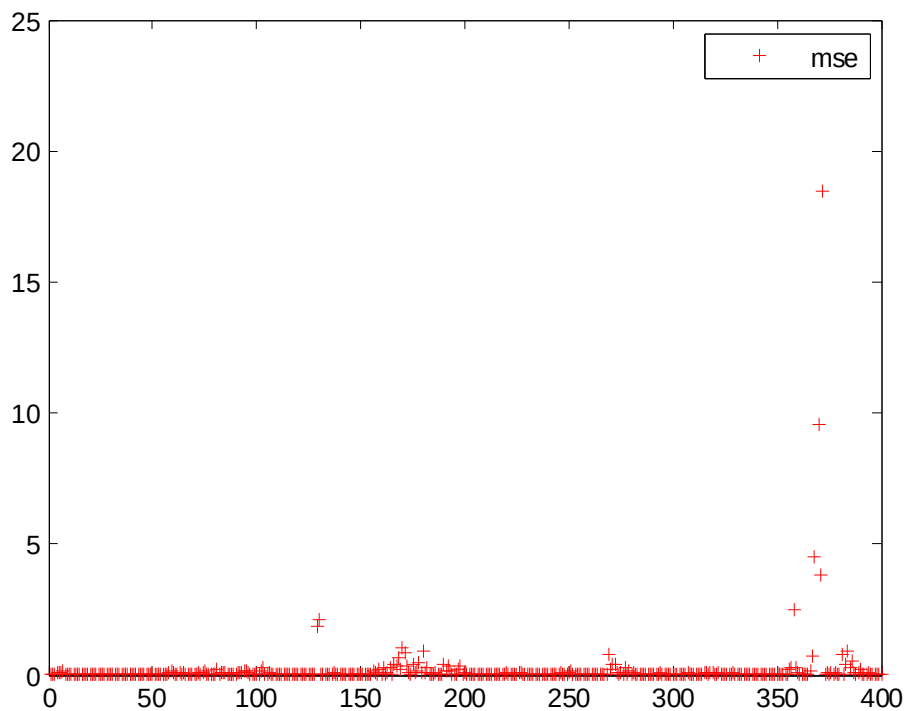


Σχήμα 32: Μέσο τετραγωνικό σφάλμα (Προεκβολή - SML2010)

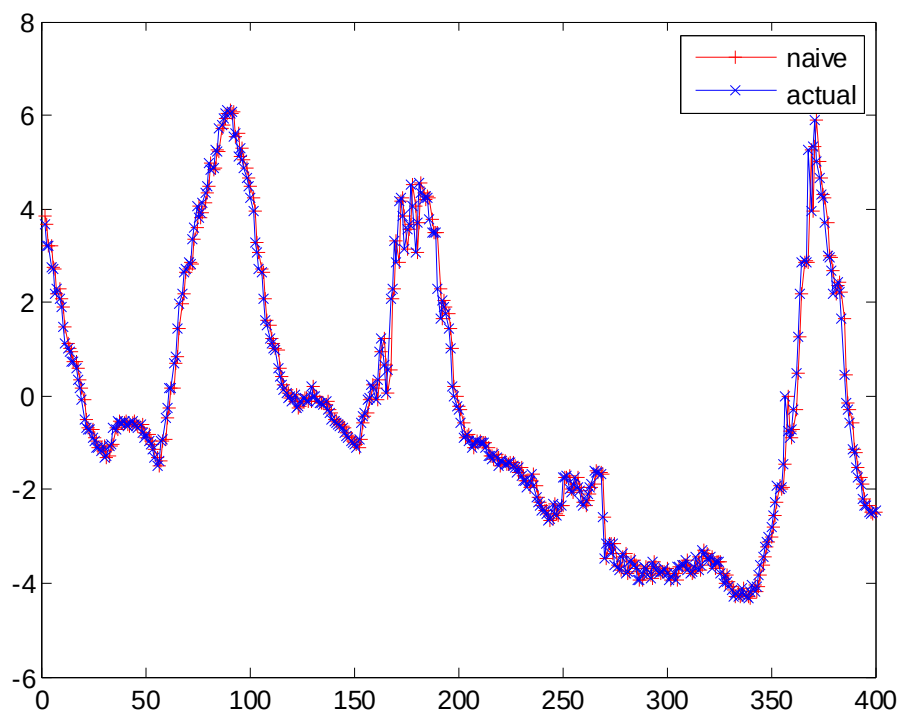


Σχήμα 33: Διαφορά προβλεπόμενης και πραγματικής τιμής (Προεκβολή παρεμβολής - SML2010)

Τα αποτελέσματα της προεκβολής παρεμβολής είναι όπως ήταν αναμενόμενο παρόμοια με αυτά των σχημάτων 31 και 32 και φαίνεται η αδυναμία των δύο αυτών μονομεταβλητών μεθόδων να μπορέσουν να αξιοποιήσουν τις πληροφορίες από όλες τις μεταβλητές για να προβλέψουν το επόμενο πλαίσιο με μικρό σφάλμα.

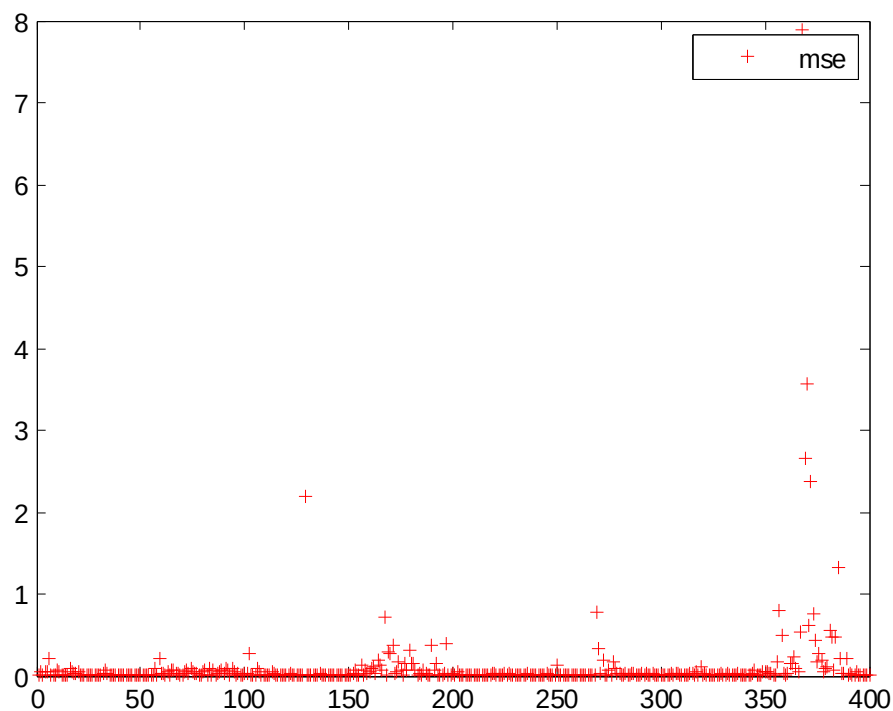


Σχήμα 34: Μέσο τετραγωνικό σφάλμα (Προεκβολή παρεμβολής - SML2010)



Σχήμα 35: Διαφορά προβλεπόμενης και πραγματικής τιμής (Naive - SML2010)

Ο αλγόριθμος αφελούς προσέγγισης φαίνεται από το παραπάνω σχήμα ότι προβλέπει αρκετά καλά τα επόμενα πλαίσια καθώς οι διαφορές από τις πραγματικές τιμές δεν εμφανίζουν μεγάλες αποκλίσεις. Αυτό φαίνεται και από το σχήμα 36 όπου απεικονίζεται το μέσο τετραγωνικό σφάλμα και η μέγιστη τιμή είναι 8 ενώ ελάχιστες περιπτώσεις έχουν τόσο υψηλό σφάλμα (περιπτώσεις απότομης αλλαγής).



Σχήμα 36: Μέσο τετραγωνικό σφάλμα (Naive - SML2010)

Πίνακας 5: Συνολικές μετρικές ARFIT (SML2010)

Μέσο σφάλμα	0.13895
Συμπίεση	6.25
False positive	14
False negative	66

Πίνακας 6: Συνολικές μετρικές προεκβολής (SML2010)

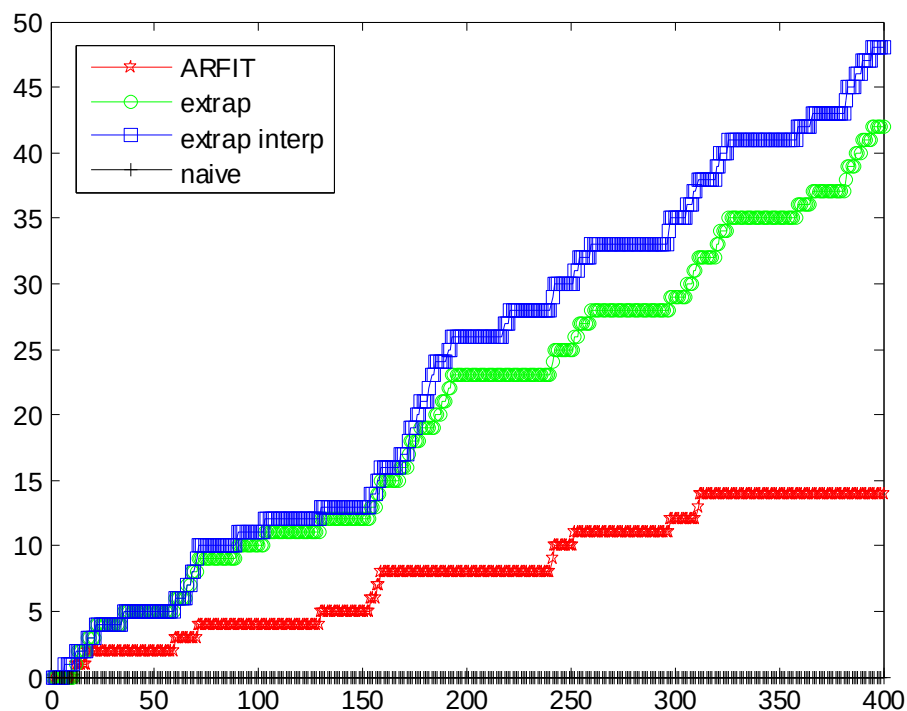
Μέσο σφάλμα	0.19271
Συμπίεση	3.6
False positive	42
False negative	47

Πίνακας 7: Συνολικές μετρικές προεκβολής παρεμβολής (SML2010)

Μέσο σφάλμα	0.23575
Συμπίεση	3.2258
False positive	48
False negative	40

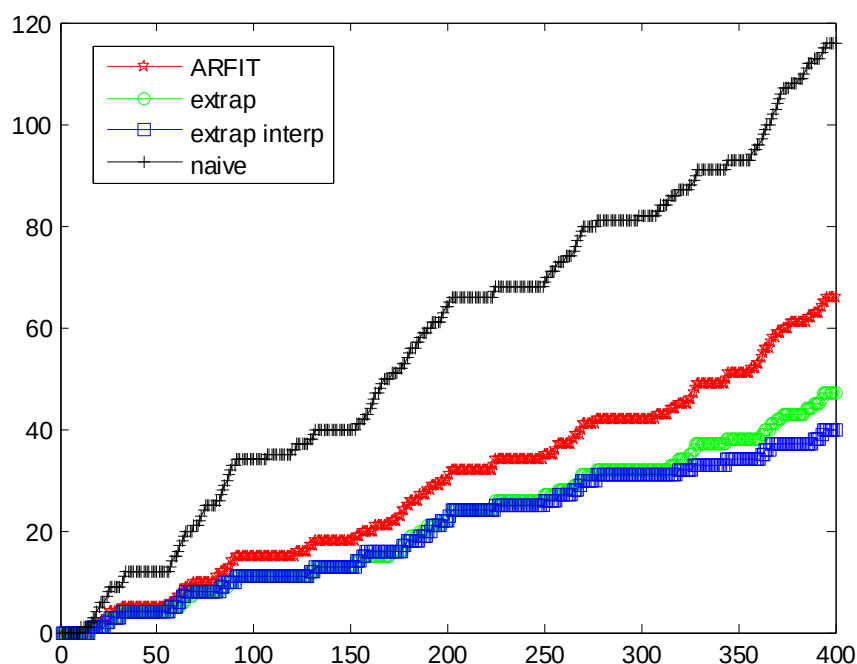
Πίνακας 8: Συνολικές μετρικές αφελής προσέγγισης (SML2010)

Μέσο σφάλμα	0.10227
Συμπίεση	-
False positive	0
False negative	116



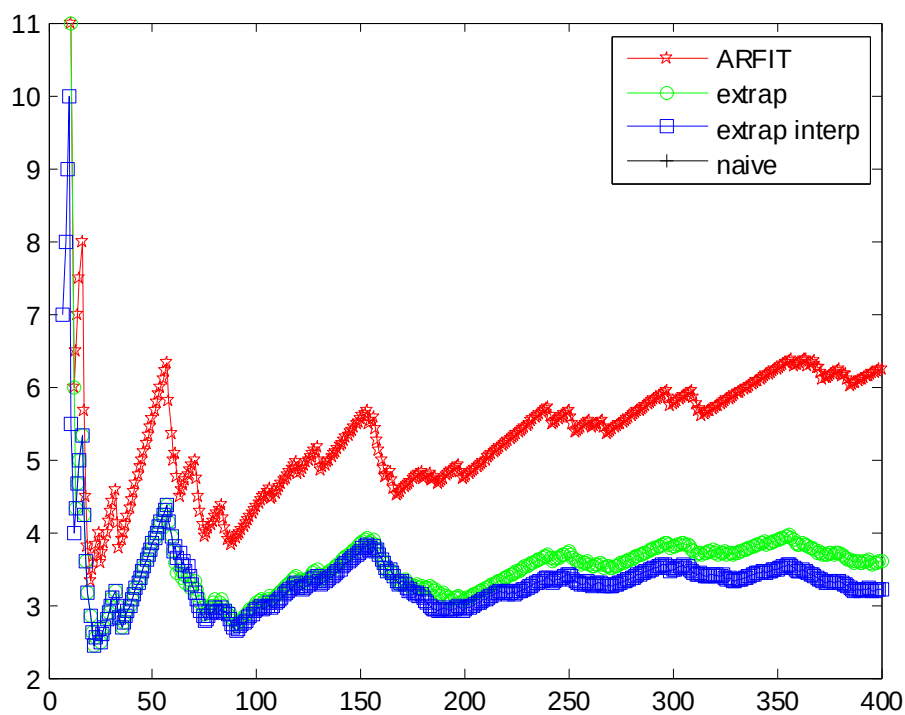
Σχήμα 37: Σύγκριση False Positive διαφορετικών μεθόδων (SML2010)

Στο Σχήμα 37 φαίνεται πως ο αλγόριθμος ARfit παρουσιάζει χαμηλά αποτελέσματα ως προς τον αριθμό των false positive. Ο αλγόριθμος αφελής προσέγγισης έχει μηδενικό αποτέλεσμα αφού επειδή θεωρεί ότι η επόμενη τιμή θα είναι ίδια με την προηγούμενη, άρα έχει μηδενική διαφορά και συνεπώς σε καμία περίπτωση δεν αποφασίζει να στείλει το προβλεπόμενο πλαίσιο, το οποίο είναι ο λόγος για τα τόσα πολλά false negative (Σχήμα 38).



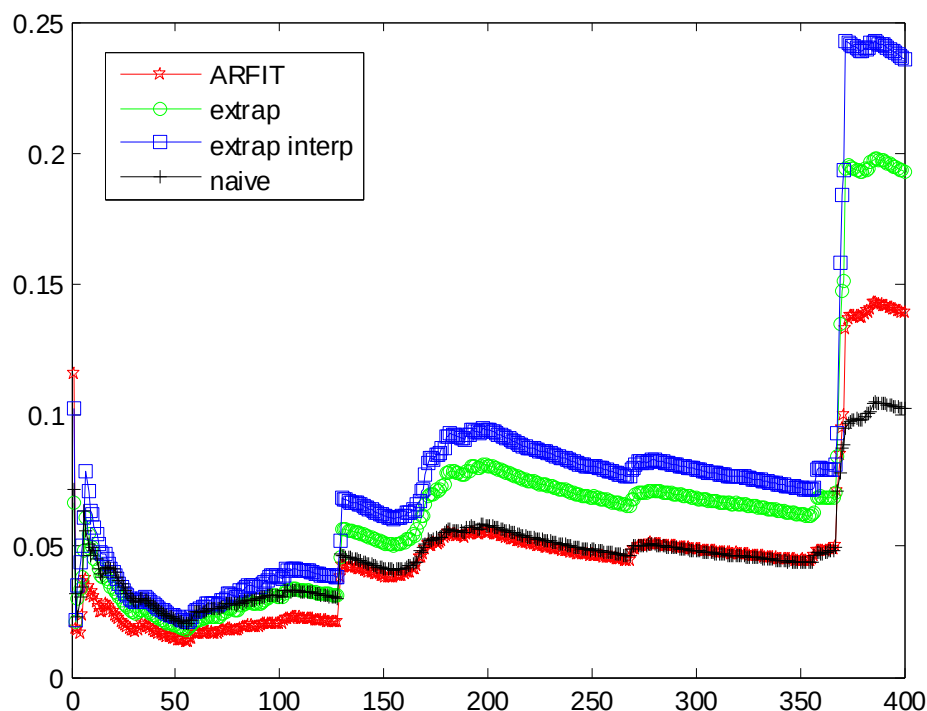
Σχήμα 38: Σύγκριση False Negative διαφορετικών μεθόδων (SML2010)

Επιπλέον φαίνεται και πως ο ARfit χρησιμοποιώντας τις πληροφορίες από πολλές μεταβλητές αντί για μόνο μία, καταφέρνει να μειώσει τα false positive σε σχέση με τους υπόλοιπους αλγόριθμους.



Σχήμα 39: Σύγκριση συμπίεσης διαφορετικών μεθόδων (SML2010)

Στο σχήμα 39 βλέπουμε πως η συμπίεση έχει καλύτερα αποτελέσματα για τον ARfit καθώς έχει πιο πολλά false negative.



Σχήμα 40: Σύγκριση μέσου όρου τετραγωνικού σφάλματος διαφορετικών μεθόδων (SML2010)

Λόγω της σωστής αξιοποίησης της πληροφορίας από διαφορετικές μεταβλητές ο αλγόριθμος ARfit επιτυγχάνει να μειώσει τον μέσο όρο του μέσου τετραγωνικού σφάλματος μεταξύ των πλαισίων που προβλέφθηκαν και των πλαισίων που πραγματικά ήταν τα επόμενα στην χρονοσειρά των δεδομένων.

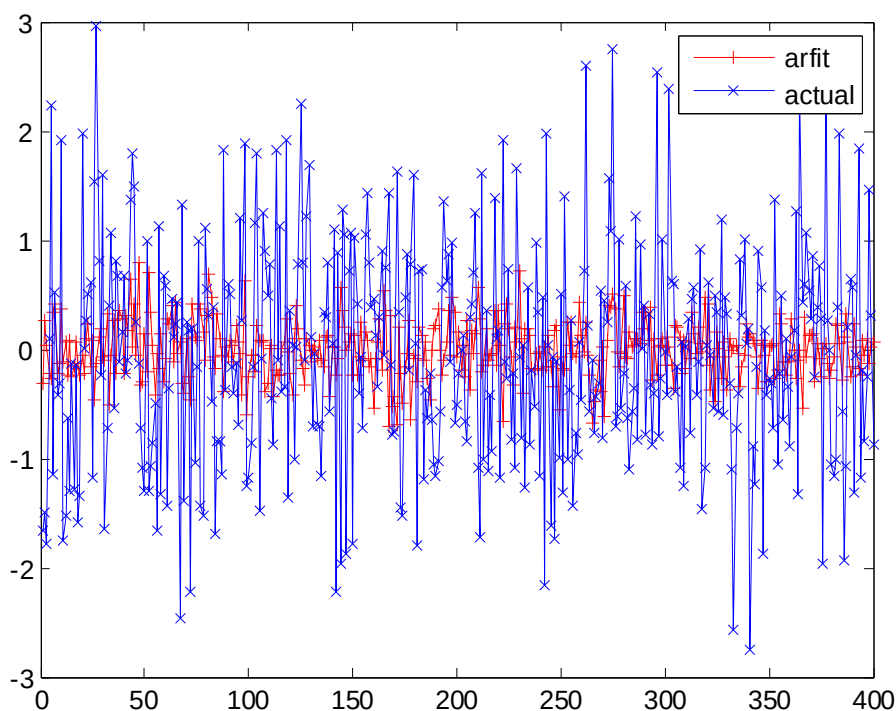
4.4 Κανονική κατανομή

Το τρίτο dataset που χρησιμοποιήθηκε είναι μία κανονική κατανομή αριθμών. Χρησιμοποιήθηκε η συνάρτηση randn από το MATLAB με αποτέλεσμα έναν πίνακα 29 μεταβλητών οι οποίες όμως σε αντίθεση με τα προηγούμενα δεδομένα που χρησιμοποιήθηκαν δεν είναι υψηλά συσχετισμένες αφού πρόκειται περί τυχαίων τιμών και όχι κάποια χρονοσειρά.

Τα δεδομένα αυτά αποθηκεύτηκαν σε έναν πίνακα $n \times 29$ όπου n είναι τα διαφορετικά πλαίσια και τροφοδοτήθηκαν στον αλγόριθμο

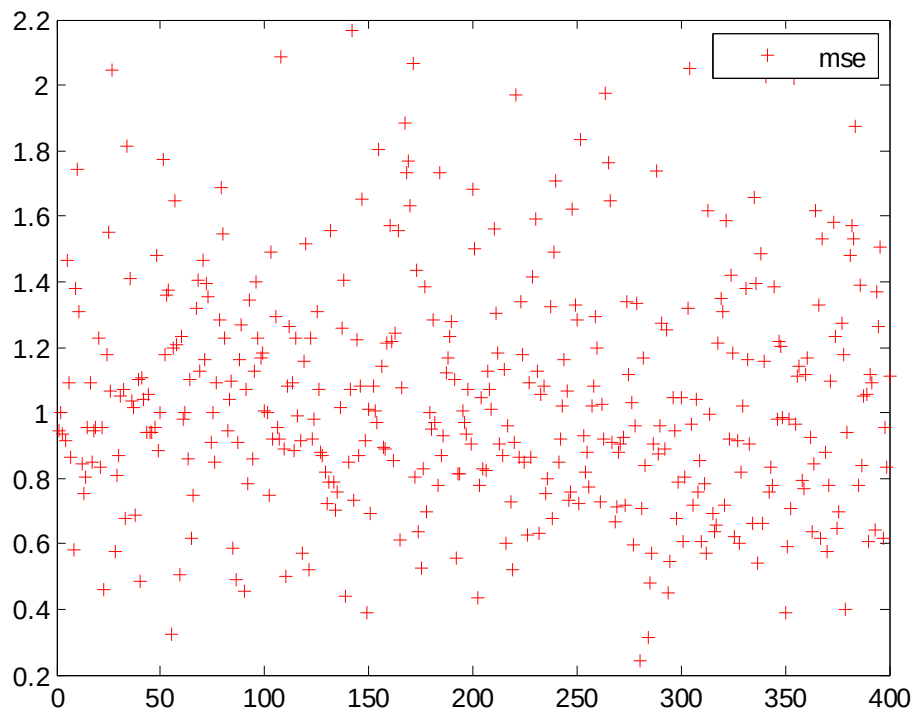
4.4.1 Σύγκριση μεθόδων

Η συνολική διακύμανση που εφαρμόζεται είναι 70% άρα στα παραπάνω δεδομένα το οποίο ισοδυναμεί με 18 μεταβλητές. Για 18 μεταβλητές εξετάζονται οι προαναφερόμενες μετρικές για συγκεκριμένο ποσοστιαίο όριο σφάλματος το οποίο είναι 30%, σε σύγκριση με τις υπόλοιπες μεθόδους.

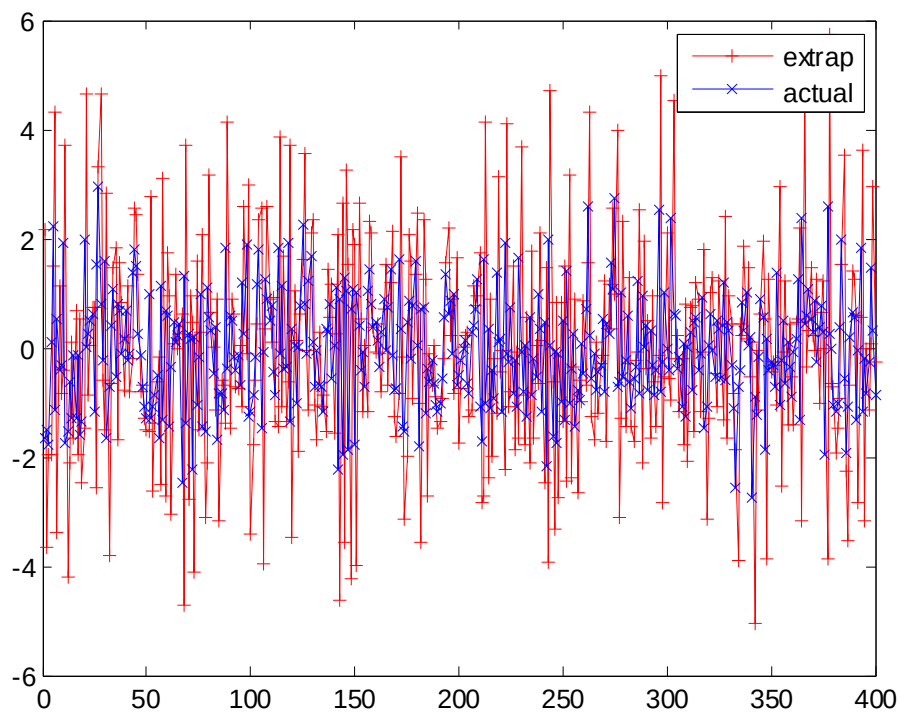


Σχήμα 41: Διαφορά προβλεπόμενης και πραγματικής τιμής (ARFIT - Gaussian)

Ο αλγόριθμος ARfit φαίνεται από το παραπάνω σχήμα ότι προβλέπει τιμές αρκετά μικρότερες από αυτές που θα ήταν το βέλτιστο να προβλέψει. Αυτό φαίνεται και από το σχήμα 42 όπου απεικονίζεται το μέσο τετραγωνικό σφάλμα και είναι εμφανής η τυχαιότητα των αριθμών.

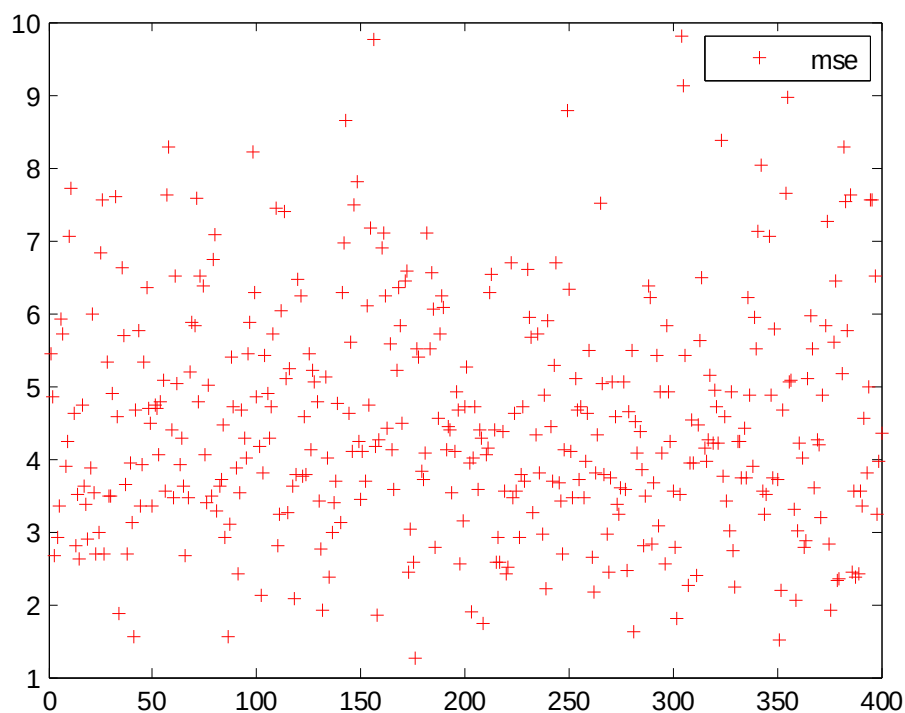


Σχήμα 42: Μέσο τετραγωνικό σφάλμα (ARFIT - Gaussian)

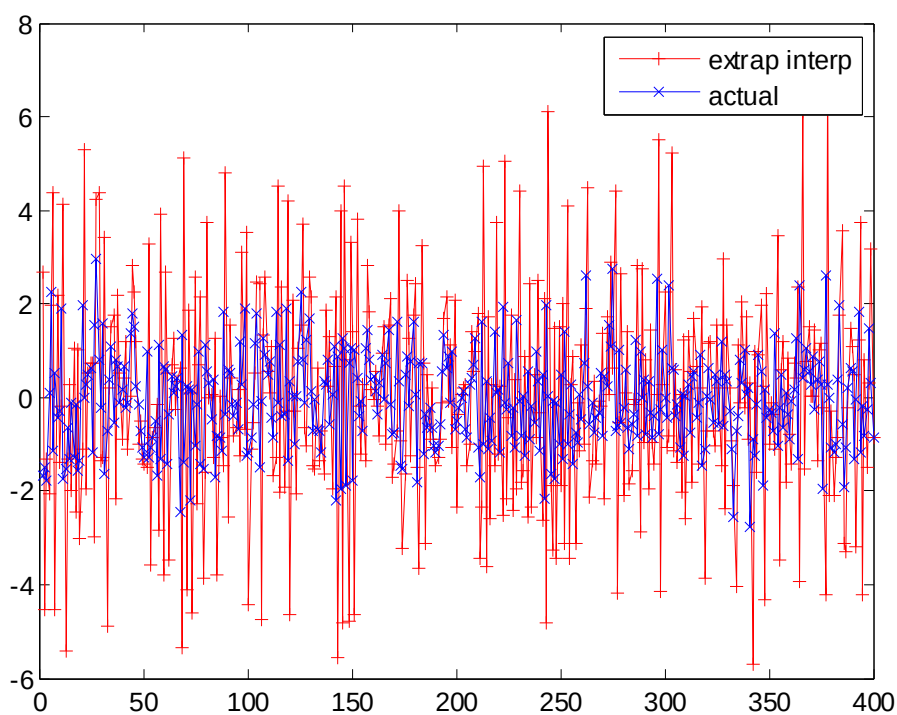


Σχήμα 43: Διαφορά προβλεπόμενης και πραγματικής τιμής (Προεκβολή - Gaussian)

Ο αλγόριθμος προεκβολής φαίνεται από το παραπάνω σχήμα ότι δεν προβλέπει αρκετά καλά τα επόμενα πλαίσια καθώς οι διαφορές από τις πραγματικές τιμές σε μερικές περιπτώσεις εμφανίζουν μεγάλες αποκλίσεις. Αυτό φαίνεται και από το σχήμα 44.

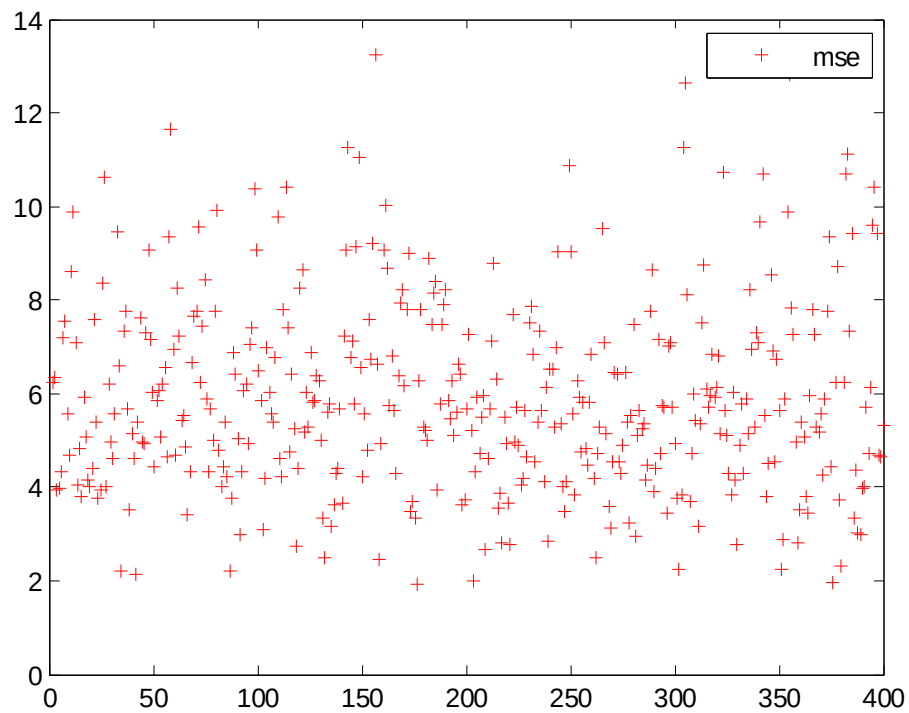


Σχήμα 44: Μέσο τετραγωνικό σφάλμα (Προεκβολή - Gaussian)

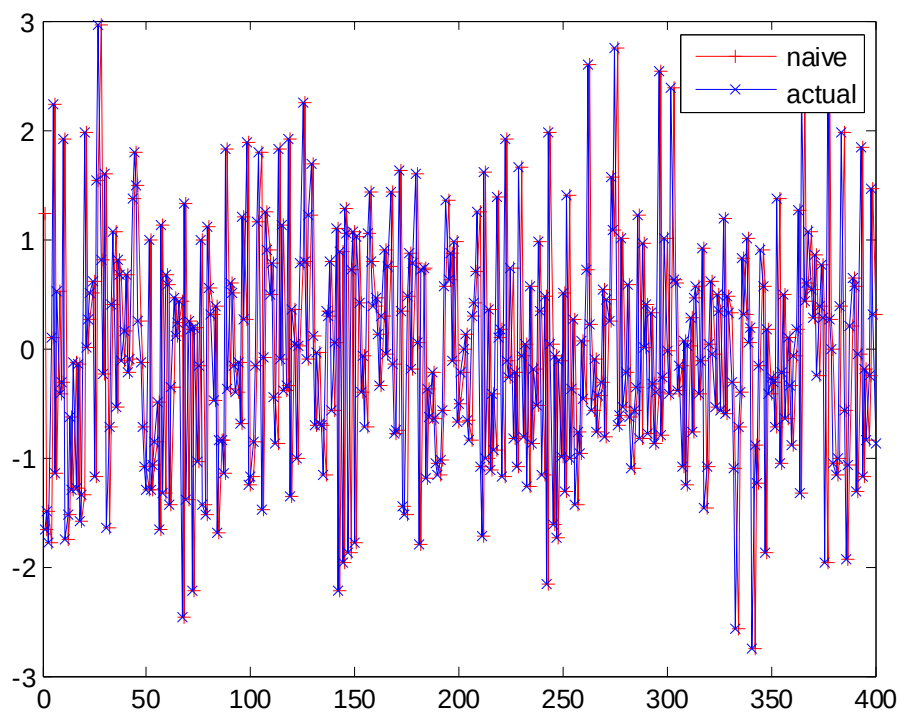


Σχήμα 45: Διαφορά προβλεπόμενης και πραγματικής τιμής (Προεκβολή Παρεμβολής - Gaussian)

Τα αποτελέσματα της προεκβολής παρεμβολής είναι όπως ήταν αναμενόμενο παρόμοια με αυτά των σχημάτων 43 και 44.

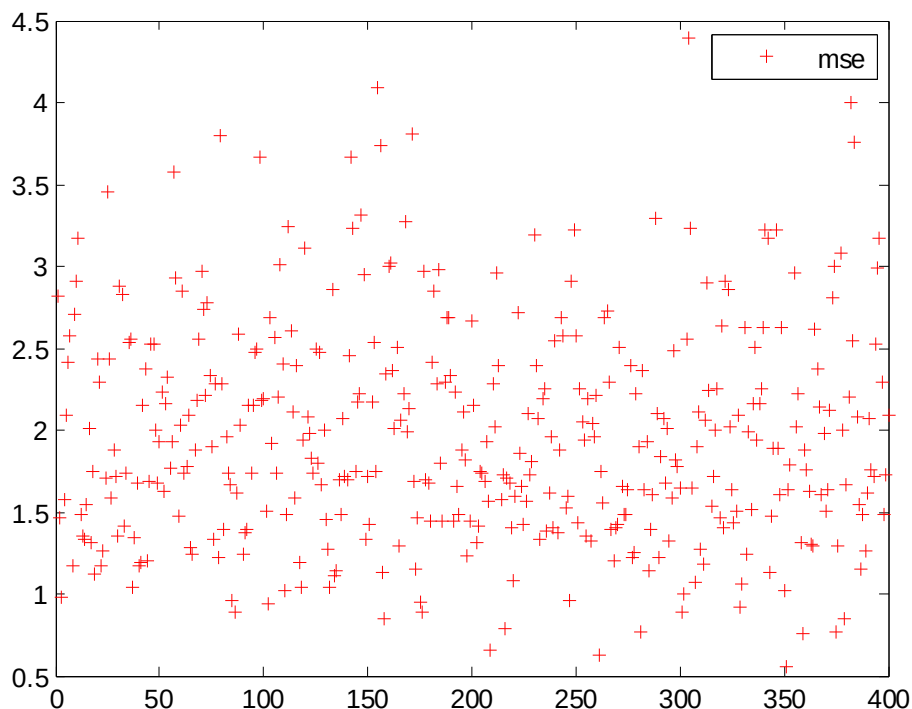


Σχήμα 46: Μέσο τετραγωνικό σφάλμα (Προεκβολή Παρεμβολής - Gaussian)



Σχήμα 47: Διαφορά προβλεπόμενης και πραγματικής τιμής (Naive - Gaussian)

Ο αλγόριθμος αφελούς προσέγγισης φαίνεται από το παραπάνω σχήμα ότι προβλέπει πάντα ότι η επόμενη τιμή θα είναι η τιμή που μόλις υπήρχε ενώ λόγω της τυχαιότητας των αριθμών αυτό δεν ισχύει παρά μόνο σε λίγες περιπτώσεις. Βέβαια καθώς οι περισσότερες τιμές κυμαίνονται γύρω από έναν μέσο είναι πιθανόν οι προβλέψεις αυτές να ευσταθούν.



Σχήμα 48: Μέσο τετραγωνικό σφάλμα (Naive - Gaussian)

Πίνακας 9: Συνολικές μετρικές ARFIT (Gaussian)

Μέσο σφάλμα	1.048
Συμπύεση	1
False positive	0
False negative	0

Πίνακας 10: Συνολικές μετρικές προεκβολής (Gaussian)

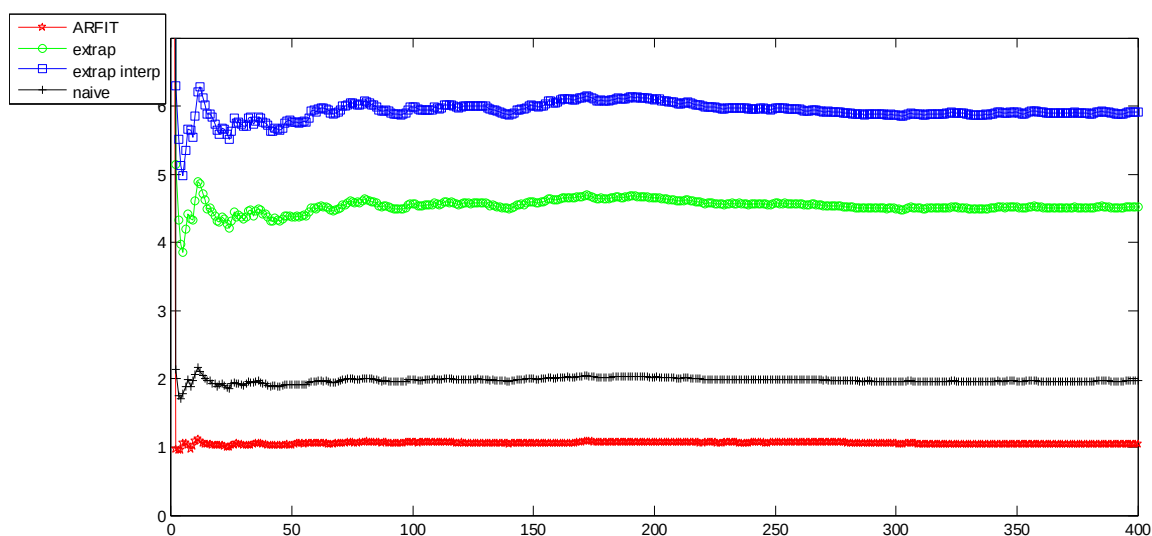
Μέσο σφάλμα	4.5226
Συμπύεση	1
False positive	0
False negative	0

Πίνακας 11: Συνολικές μετρικές προεκβολής παρεμβολής (Gaussian)

Μέσο σφάλμα	5.9152
Συμπίεση	1
False positive	0
False negative	0

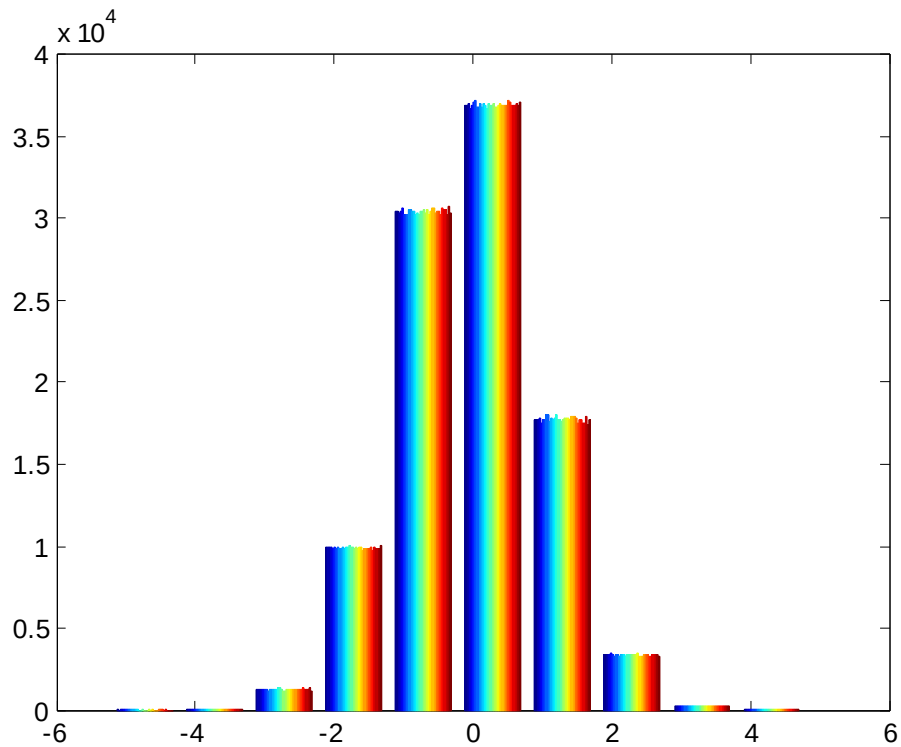
Πίνακας 12: Συνολικές μετρικές αφελής προσέγγισης (Gaussian)

Μέσο σφάλμα	1.9726
Συμπίεση	-
False positive	0
False negative	400

**Σχήμα 49: Σύγκριση μέσου όρου τετραγωνικού σφάλματος διαφορετικών μεθόδων (Gauss)**

Λόγω της σωστής αξιοποίησης της πληροφορίας από διαφορετικές μεταβλητές ο αλγόριθμος ARfit επιτυγχάνει να μειώσει τον μέσο όρο του μέσου τετραγωνικού σφάλματος μεταξύ των πλαισίων που προβλέφθηκαν και των πλαισίων που πραγματικά ήταν τα επόμενα στην χρονοσειρά των δεδομένων.

Το ιστόγραμμα των δεδομένων της συγκεκριμένης μετατοπισμένης κανονικής κατανομής παρουσιάζεται στο σχήμα 50.



Σχήμα 50: Ιστόγραμμα κανονικής κατανομής

5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Τα αποτελέσματα από τα διαφορετικά δεδομένα εισόδου με βάση τα οποία εξήχθησαν συγκεκριμένες μετρικές συνδράμουν ώστε να μπορέσουν να καταγραφούν τα συμπεράσματα. Ο αλγόριθμος ARfit χρησιμοποιεί ένα πολυμεταβλητό μοντέλο περιγραφής της εκάστοτε χρονοσειράς. Οι αλγόριθμοι προεκβολής και προεκβολής παρεμβολής είναι και οι δύο αλγόριθμοι μονομεταβλητοί δηλαδή χρησιμοποιούν μόνο την πληροφορία από μία συγκεκριμένη μεταβλητή για να προβλέψουν την αμέσως επόμενη τιμή της συγκεκριμένης. Ο αλγόριθμος αφελούς προσέγγισης σε κάθε περίπτωση έχει εισαχθεί μόνο για να υπάρχει ένα μέτρο σύγκρισης με την πιο απλή προσέγγιση των πιο σύνθετων αλγορίθμων, και δεν μπορεί να χρησιμοποιηθεί σε πραγματικές εγκαταστάσεις σε συστήματα αισθητήρων όπου παρουσιάζονται μεταβολές και δη σημαντικές μεταβολές.

Οι δύο μεταβλητοί αλγόριθμοι όπως ήταν αναμενόμενο παρουσιάζουν παρόμοια αποτελέσματα με τον αλγόριθμο προεκβολής να παρουσιάζει μικρότερο μέσο όρο του μέσου τετραγωνικού σφάλματος σε όλες τις περιπτώσεις. Παρατηρείται μία ισορροπία των δύο αλγορίθμων μεταξύ false positive και false negative, δηλαδή στις περιπτώσεις όπου ο ένας αλγόριθμος από τους δύο εμφανίζει περισσότερα false negative, ο άλλος αλγόριθμος εμφανίζει περισσότερα false negative.

Ο αλγόριθμος ARfit και στα τρία διαφορετικά δεδομένα εισόδου παρουσιάζει χαμηλότερο μέσο τετραγωνικό σφάλμα σε κάθε μία εκ των μεταβλητών άρα και στον μέσο όσο τους που εικονίζεται στους πίνακες 1, 5 και 9.

Αυτό οφείλεται στο ότι το μοντέλο MAR που εφαρμόζεται από τον αλγόριθμο είναι πιο κοντά στα δεδομένα της χρονοσειράς και λαμβάνει πληροφορίες που οι άλλοι μονομεταβλητοί αλγόριθμοι δεν μπορούν να επεξεργαστούν και να αξιολογήσουν. Αυτές οι πληροφορίες οι οποίες είναι χρήσιμες για μία όσο τον δυνατόν έγκυρη πρόβλεψη του επόμενου πλαισίου και προφανώς προέρχονται από τις υπόλοιπες μεταβλητές καθώς μεταβολές σε αυτές μπορεί να σημαίνουν μία πιθανή μεταβολή σε κάποια άλλη μεταβλητή η οποία δεν είχε μεγάλες μεταβολές, και αυτό είναι σε θέση να το προβλέψει ο αλγόριθμος ARfit μέσω του μοντέλου του.

Σε δεδομένα που δεν αποτελούσαν ουσιαστικά συγκεκριμένη χρονοσειρά, όπως της τρίτης ομάδας δεδομένων που άνηκαν στην κανονική κατανομή φαίνεται ότι πάλι ο αλγόριθμος ARfit εμφάνισε καλύτερα αποτελέσματα με την μορφή χαμηλότερου μέσου τετραγωνικού σφάλματος σε σύγκριση με τις άλλες μεθόδους. Αυτό συμβαίνει λόγω της φύσης του αλγορίθμου ARfit που ευνοείται από πλαίσια αριθμών που ανήκουν στην κανονική κατανομή.

Φαίνεται από τα αποτελέσματα τα πλέον καλύτερα σε σχέση με τους άλλους αλγορίθμους, ο ARfit τα εμφανίζει στην πρώτη ομάδα δεδομένων όπου εμφανίζονται μεγάλες αλλαγές μεταξύ διαφορετικών χρονικών στιγμών. Οι μεγάλες αυτές αλλαγές λαμβάνονται από τους μονομεταβλητούς αλγορίθμους ως εφελκυστικά για ακόμα μεγαλύτερες εκτιμήσεις οι οποίες όμως είναι λανθασμένες.

Έτσι με τον αλγόριθμο ARfit επιτυγχάνεται ακόμα μεγαλύτερη συμπίεση των αρχικών δεδομένων.

6. ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Τα αποτελέσματα είναι ενθαρρυντικά και ο αλγόριθμος επιδέχεται περαιτέρω εξερεύνησης και βελτιστοποιήσεων.

Μία από τις μελλοντικές επεκτάσεις που μπορούν να γίνουν αλλαχθούν κάποιες από τις παραμέτρους του ARfit όπως το P_{min} & P_{max} , όπως και το πλήθος των δεδομένων στα οποία βασίζεται το μοντέλο των μεταβλητών, τα οποία είναι σταθερά σε όλα τα παραπάνω πειράματα.

Σε δεδομένα τα οποία έχουν εξαχθεί από κανονική κατανομή μπορούν να δοκιμαστούν οι μετρικές των false positive και false negative με μεγαλύτερα οριακά σχετικά σφάλματα, γιατί αυτά που δοκιμάστηκαν (30%) έχουν μεν πρακτικό ενδιαφέρον αλλά οι αποκλίσεις είναι πολύ μεγάλες ώστε να υπάρξει κάποια διαφορά στα αποτελέσματα.

Επίσης μπορούν να αναλυθούν οι μεταβολές και χωρίς PCA ώστε να υπάρχει μία σύγκριση μεταξύ των μεταβολών που συμβαίνουν στα μετασχηματισμένα δεδομένα σε σχέση και σε σύγκριση με τα αρχικά δεδομένα.

Αυτές οι αναλύσεις των μεταβολών μπορούν να γίνουν για μία εφαρμογή του PCA ώστε να βρεθεί πόσο αποκλίνουν από τα πραγματικά δεδομένα καθώς απομακρυνόμαστε χρονικά από τα πλαίσια τα οποία χρησιμοποιήθηκαν για να εξαχθούν τα διανύσματα των μεταβλητών που απαρτίζουν τις συνιστώσες. Αυτή η ανάλυση μπορεί να επεκταθεί και για παραπάνω από ένα κύκλο εφαρμογής του PCA αλγορίθμου και να συγκριθούν τα αποτελέσματα.

Τέλος, στον παρόντα αλγόριθμο εφαρμογής του PCA και αλλαγής των ενδιάμεσων διαστημάτων, διατηρείται σταθερό το ποσοστό διακύμανσης που πρέπει να επεξηγείται από τις μεταβλητές με την μεγαλύτερη διακύμανση και αν πλήθος των μεταβλητών δεν είναι το ίδιο με αυτό που είχε βρεθεί στην τελευταία εφαρμογή του αλγορίθμου τότε το αντίστοιχο διάστημα μειώνεται. Σαν επέκταση αυτού, αντί για να μειώνεται το διάστημα με βάση το πλήθος των μεταβλητών, μπορεί να γίνει μία υλοποίηση μίας μεθόδου που ελέγχει πόσο απέχουν τα νέα διανύσματα των συνιστωσών από τα παλιά διανύσματα. Έτσι μία αλλαγή στο πλήθος των μεταβλητών η οποία παράλληλα είναι σχετικά μικρή αλλαγή ως προς τα διανύσματα δεν θα επιφέρει αλλαγές στα διαστήματα στα οποία εφαρμόζουμε τον αλγόριθμο ARfit.

Εν κατακλείδι, θα μπορούσε να γίνει και μία μελέτη για το ενεργειακό πλεονέκτημα ενός τέτοιου αλγορίθμου αφού υπολογίζει όλο το επόμενο πλαίσιο μαζί και όχι μεμονωμένα τις τιμές των μεταβλητών όπως οι μονομεταβλητοί αλγόριθμοι προεκβολής.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός Όρος
Extrapolation	Προεκβολή
Mobile computing	Κινητός υπολογισμός
Pervasive computing	Διάχυτος υπολογισμός
Mobility	Κινητικότητα
Context	Πλαίσιο
Multivariate Auto-Regressive model	Υπόδειγμα Αυτοπαλίνδρομου Διανύσματος
Mean squared error	Μέσο τετραγωνικό σφάλμα
False negative	Ψευδώς αρνητικό
False positive	Ψευδώς θετικό

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

PCA	Principal Component Analysis
ARfit	Multivariate AutoRegressive Model Fitting
SVD	Singular Value Decomposition
MAR	Multivariate Auto-Regressive model
MSE	Mean Squared Error
FN	False Negative
FP	False Positive

ΠΑΡΑΡΤΗΜΑ Ι

Ο κώδικας των υλοποιημένων συναρτήσεων PCAARFIT.m και των υπολοίπων συναρτήσεων που υλοποιήθηκαν. Σημειώνεται ότι χρησιμοποιήθηκε η βιβλιοθήκη ARfit του MATLAB.

PCAARFIT.m

```
%orio allaghs katastashs. 0.3 = 30%
errorBoundary = 0.3;
%example: using 80 for 80% - sets num of principal components
minimumExplainedValue = 70;

%set 1 for active and anything else for inactive
setNumberOfPrincipalComponentsBasedOnExplainedPercentage = 1;

%ari8mos twm principal components pou kratame = 9 IF AND ONLY IF
%setNumberOfPrincipalComponentsBasedOnExplainedPercentage != 1
numOfPrincipalComponents = 12;

%set dataset
data = data_mar;

pca_length = 400;

numberOfFramesToCheck = 400;
%to mege8os tou window to opoio trofodotei to arfit kai ginetai kai to
%zscore (standardisation) %htan 100 kai 100
chunk = 400;
chunkZscore = 400;

%methods: -0:arfit, -1:naive, -2:extrapolation, -3:extrap interpolation
method = 0;

%to min kai to max tou arfit
arfitFiMin = 1;
arfitFiMax = 4;

%there are 9 (=Num of principal components) columnVariables
columnVariable = 1;
%legend1 = strcat('arfit_ 1-4:', num2str(columnVariable));
legend1 = 'naive';
legend2 = 'actual';

siglevTotal=0;
globalCounter =1;

%ayto pou kanoume pleon einai pca se ena kommati sthn arxh me mhkos
%pca_length kai kratame tis metablhtes (toys syndyasmous aytwn)
%Etsi gia to epomeno diasthma non_pca_length trexoume ton arfit ekei
%xwris na kanoume pca. Apla trexoume ton arfit gia ayto to diasthma to
%opoio einai pollaplasio tou pca_length kata ena compression_rate
%pca_length = 400;

compression_rate = 8;
```

```

min_compression_rate = 3;

non_pca_length = pca_length*compression_rate;

%to compression rate den mporei na einai mikrotero apo 3 kai meiwnetai an
%sto epomeno pca pou paragetai to plh8os ton metavlhtwn einai diaforetiko
%apo thn prohghoumenh fora enw ayksanetai an einai to idio plh8os (px 9
%metavlhtes)

current_start = 1;
current_end = pca_length+1;
index_iteration = -1;

%init stuff
cccc12_30_naive_mseAverageArray = [];
cccc12_30_naive_compressionFinalArray = [];
cccc12_30_naive_totalFPVectorArray = [];
cccc12_30_naive_totalFNVectorArray = [];
cccc12_30_naive_meanMseArray = [];

while 1

    disp(current_start);
    disp(current_end);

    index_iteration = index_iteration +1;
    %an einai mon'o kano pca, an einai zygo kanw arfit
    if (mod(index_iteration,2) == 0)
        disp('do pca');

        %currently this is not dynamic
        cleanDataBeforePca = data(current_start:current_end, :);

        %if we use 'Centered', false we must not use zscore(cleanDataBeforePca)
        %in order for the
        %result to be correct - this is standartization
        %[coeff,~,latent,~,explained] =
pca((cleanDataBeforePca),'NumComponents',numOfPrincipalComponents,'Centered',false);

        if (setNumberOfPrincipalComponentsBasedOnExplainedPercentage == 1)
            previousNumOfPrincipalComponents = numOfPrincipalComponents;
            [~,~,~,~,explained] = pca(zscore(cleanDataBeforePca));
            numOfPrincipalComponents = getNumberOfPCForMinimumExplainedValue(explained,
minimumExplainedValue);
            if (numOfPrincipalComponents ~= previousNumOfPrincipalComponents)
                %decrease compression rate
                if (compression_rate > min_compression_rate)
                    compression_rate = compression_rate -1;
                end
            else
                %increase compression rate
                compression_rate = compression_rate +1;

            end
        end
        %numOfPrincipalComponents
        %return;

        [coeff,~,latent,~,explained] =
pca(zscore(cleanDataBeforePca),'NumComponents',numOfPrincipalComponents);
    
```

```

current_start = current_end+1;
current_end = current_end + pca_length*compression_rate;

else
disp('do mar');

%binTable holds comparison between the expectedValue and the actualValue
mse = [];
binTable = [];
binVectorTable = [];
%actBinTable holds comparison between the actualValue and the previous one
actBinTable = [];
actBinVectorTable = [];
expectedValuesToPlot = [];
arrivedValuesToPlot = [];
xValuesToPlot = [];
binVectorOriginalTable = [];

%we must calculate a new matrix of actually pca data using the coeff
%from the previous pca done with matlab

%create transpose of coeff so that it can be used for multiplication
coeffTruncated = coeff(:,1:numOfPrincipalComponents);

%get the data and perform mar
m=0;
%globalCounter = 0;

for indexForMar = current_start:current_end
    %globalCounter = m;

    indexForMar;

    scoreRaw = data(indexForMar-chunkZscore:indexForMar-1,:);
    %scoreRawAfterZscore = zscore(scoreRaw);
    %simulate pca
    score = zscore(scoreRaw)*coeffTruncated;
    %score = (scoreRaw)*coeffTruncated;

    %epeidh exw krathsei chunkZscore kai oxi chunk dedomena 8a
    %prepei krathsw mono ta chunk teleytaia
    chunkIndex = size(score,1) - chunk + 1;
    score = score(chunkIndex:end,:);

    %prepei na yparxei xExpected value gia
    %na thn sygkrinw ARA m > 1
    if m > 1
        %xJustCame einai h teleytaia timh tou pinaka pou 8ewroume oti
        %einai h trexousa
        xJustCame = score(end, :);

        %retrieve original data
        originalDataPrevious = scoreRaw(end-1,:);
        originalDataLast = scoreRaw(end,:);
        %disp('original data vector');
        binVectorOriginalTable = [binVectorOriginalTable ;
calBinTableVector(originalDataPrevious', originalDataLast', errorBoundary)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        %disp('binary line:');

```

```

%disp(calBinTable(score(end-1, :)', xExpected));

%disp('actual binary line:');
%disp(calBinTable(score(end-1, :)', xJustCame));

%add expected and arrived values to appropriate arrays:
expectedValuesToPlot = vertcat(expectedValuesToPlot, xExpected');
arrivedValuesToPlot = vertcat(arrivedValuesToPlot, xJustCame');
xValuesToPlot = [xValuesToPlot; globalCounter];

globalCounter = globalCounter + 1;

%vres tis diafores kai ypologise ton binary pinaka
binTable = [binTable ; calBinTable(lastVectorUsed', xExpected, errorBoundary)];

binVectorTable = [binVectorTable ; calBinTableVector(lastVectorUsed', xExpected,
errorBoundary)];

%oi actual times einai aytes gia tis opoies mporoume na vroume
%ek tou apotelesmatos an 8a eprepe na stalei to vector 'h oxi

actBinVectorTable = [actBinVectorTable ; calBinTableVector(lastVectorUsed',
xJustCame, errorBoundary)];

actBinTable = [actBinTable ; calBinTable(lastVectorUsed', xJustCame, errorBoundary)];

mse = [mse ; calMseTable(xExpected, xJustCame)'];

%just for showing
binReversed = binTable';

if globalCounter > numberOfFramesToCheck

    %plots of expected values and values arrived
    figure;
    plot(xValuesToPlot, expectedValuesToPlot(:, columnVariable), '-+r');
    hold;
    plot(xValuesToPlot, arrivedValuesToPlot(:, columnVariable), '-xb');
    legend(legend1, legend2);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %false positive and false negative
    tableOfFPandFN = actBinTable - binTable;
    %all 1 are false negative - 1 per vector is enough???
    totalFN = sum(tableOfFPandFN == 1);
    %disp('total false negative:');
    %disp(totalFN);

    %all -1 are false positive
    totalFP = sum(tableOfFPandFN == -1);
    %disp('total false positive:');
    %disp(totalFP);

    %false positive and false negative per vector
    totalFPperVector = 0;
    totalFNperVector = 0;
    for indexFP = 1:size(tableOfFPandFN,1)
        vectorBinTable = binTable(indexFP,:);
        vectorActualTable = actBinTable(indexFP,:);
        anyBinTable = any(vectorBinTable);
        anyActualTable = any(vectorActualTable);
    end
end

```

```

    if ((anyBinTable == 1) && (anyActualTable == 0))

        totalFPperVector = totalFPperVector +1;
    elseif ((anyBinTable == 0) && (anyActualTable == 1))

        totalFNperVector = totalFNperVector +1;
    end
end

disp('total false negative per vector:');
disp(totalFNperVector);

disp('total false positive per vector:');
disp(totalFPperVector);

%false positive and false negative per vector
%second version: average or vector error
tableOfFPandFNVector = actBinVectorTable - binVectorTable;
%all 1 are false negative - 1 per vector is enough???
totalFNVector = sum(tableOfFPandFNVector == 1);
disp('total false negative vector (Average):');
disp(totalFNVector);

%all -1 are false positive
totalFPVector = sum(tableOfFPandFNVector == -1);
disp('total false positive vector (Average):');
disp(totalFPVector);

cccc12_30_naive_totalFNVectorArray =
[cccc12_30_naive_totalFNVectorArray;totalFNVector];
cccc12_30_naive_totalFPVectorArray =
[cccc12_30_naive_totalFPVectorArray;totalFPVector];
figure;
plot(xValuesToPlot, cccc12_30_naive_totalFPVectorArray, '-+g');
legend('false positive');
figure;
plot(xValuesToPlot, cccc12_30_naive_totalFNVectorArray, '-+r');
legend('false negative');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('compression:');
numOfVectorsSent = sum(binVectorTable==1);
compressionFinal = size(binVectorTable,1) / numOfVectorsSent;
cccc12_30_naive_compressionFinalArray =
[cccc12_30_naive_compressionFinalArray;compressionFinal];
disp(compressionFinal);
figure;
plot(xValuesToPlot, cccc12_30_naive_compressionFinalArray, '-+g');
legend('compression');

disp('optimal compression:');
actualNumOfVectorsSent = sum(actBinVectorTable==1);
compressionFinalMaximum = size(actBinVectorTable,1) / actualNumOfVectorsSent;
disp(compressionFinalMaximum);

disp('optimal ORIGINAL compression:');
actualNumOfVectorsSentOriginal = sum(binVectorOriginalTable==1);
compressionFinalMaximumOriginal = size(binVectorOriginalTable,1) /
actualNumOfVectorsSentOriginal;
disp(compressionFinalMaximumOriginal);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%find errors based on mse
mseForEachColumn = sum(mse);
%size(mse)
%mseForEachColumn

%get average error per column
mseForEachColumnAverage =
bsxfun(@rdivide,mseForEachColumn,size(mse,1)*ones(1,numOfPrincipalComponents));
%mseForEachColumnAverage

totalMeanError = mean(mseForEachColumnAverage);
cccc12_30_naive_mseAverageArray =
[cccc12_30_naive_mseAverageArray;totalMeanError];
totalMeanError
figure;
plot(xValuesToPlot, cccc12_30_naive_mseAverageArray, '-+r');
legend('average mse');

cccc12_30_naive_meanMseArray = mean(mse,2);
figure;
plot(xValuesToPlot, mean(mse,2), '+r');
legend('mse');

return
end

%get arrays of criteria
%compression
numOfVectorsSent = sum(binVectorTable==1);
compressionFinal = size(binVectorTable,1) / numOfVectorsSent;
cccc12_30_naive_compressionFinalArray =
[cccc12_30_naive_compressionFinalArray;compressionFinal];

%mse
mseForEachColumn = sum(mse);
mseForEachColumnAverage =
bsxfun(@rdivide,mseForEachColumn,size(mse,1)*ones(1,numOfPrincipalComponents));
totalMeanError = mean(mseForEachColumnAverage);
cccc12_30_naive_mseAverageArray =
[cccc12_30_naive_mseAverageArray;totalMeanError];

%FN
tableOfFPandFNVector = actBinVectorTable - binVectorTable;
%all 1 are false negative - 1 per vector is enough???
totalFNVector = sum(tableOfFPandFNVector == 1);
cccc12_30_naive_totalFNVectorArray =
[cccc12_30_naive_totalFNVectorArray;totalFNVector];

%FP
%all -1 are false positive
totalFPVector = sum(tableOfFPandFNVector == -1);
cccc12_30_naive_totalFPVectorArray =
[cccc12_30_naive_totalFPVectorArray;totalFPVector];

end

m = m+1;

%ARFIT related stuff
[w, A, C, sbc, fpe, th]=arfit(score, arfitFiMin, arfitFiMax, 'sbc');
[siglev,res] = arres(w,A,score);

```

```

%disp('Siglev for upcoming counter');
%disp(siglev);
%apo8hkeyoume thn metrikh pou mas deixnei an mporei na
%xrhsimopoi8ei arfit
siglevTotal = [siglevTotal;siglev];
%to A einai ola ta A1|A2|A3... concatenated
%construct
%find arfitOPT
[d1,d2] = size(A);
arfitFiOpt = d2/numOfPrincipalComponents;
aMatrixNumMax = (arfitFiOpt)*(numOfPrincipalComponents-1);
pointerForX = 0;

lim = chunk +1;

%PREDICT NEXT VALUE

%to swnoume edw giati meta 8a prepei na ksanakanoume zscore ara mporei
%ayth h timh na allaksei ligo, opote prepei na to apo8hkeysoume edw ayto to
%vector gia na mporoume na to sygkrinoume me to actual
%vector ( to expected 8a sygkri8ei me to neo)
lastVectorUsed = score(end,:);

if method == 0
    %first way
    %regular prediction with arfit
    xExpected = predict(score,arfitFiOpt,A,w,numOfPrincipalComponents,lim);

    %show original data used to produce the prediction
    %    disp('show original data used to produce the prediction');
    %    pastValuesToDisplay2 = [];
    %    for(i=1:arfitFiOpt)
    %        %first accessed is newer
    %        %disp(v(lim-i,:));
    %        pastValuesToDisplay2 = vertcat( scoreRaw(lim-i,:),pastValuesToDisplay2);
    %    end
    %disp(pastValuesToDisplay2);

elseif method == 1

    %second way
    %naive prediction
    xExpected = score(end, :);

elseif method == 2

    %third way
    %extrapolation
    tempExpected = [];
    numOfPastValues = 3;
    for indexForVariable = 1:size(score(end, :))

        %to pio aristero ston pinaka einai to pio palio
        yValues = score(end-numOfPastValues+1:end,indexForVariable);
        xValues = -numOfPastValues:-1;
        %kai ta dyo dianysmata einai orizontia
        %to xValues ksekinaei ousiastika apo to -numOfPastValues wste
        %panta h timh poy 8a toy zhthsoyme na mas dwsei 8a einai h timh
        %opou to x einai 0 (h opoia 8a einai h expected)
        %disp(yValues');
        %disp(xValues');
    end

```



```

    %plot(xValues, yValues);
    %pause;

    s = csaps(xValues,yValues);
    sn = fnxtr(s);
    %           fnplt(s, [-numOfPastValues 1]), hold on, fnplt(sn,[-numOfPastValues 1], 'r')
    %           legend('cubic smoothing spline','properly extrapolated')
    %           hold off
    %           pause;

    expectedZero = fnval(sn,0);
    %disp('expected');
    %disp(expectedZero);

    tempExpected = horzcat(tempExpected, expectedZero);

end

elseif method == 3
    %fourth way
    %extrapolate interpolation using two methods: pchip and spline
    tempExpected = [];
    numOfPastValues = 2;
    for indexForVariable = 1:size(score(end, :))

        %to pio aristero ston pinaka einai to pio palio
        yValues = score(end-numOfPastValues+1:end,indexForVariable)';
        xValues = -numOfPastValues:-1;
        %kai ta dyo dianysmata einai orizontia
        %to xValues ksekinaei ousiastika apo to -numOfPastValues wste
        %panta h timh poy 8a toy zhthsoyme na mas dwsei 8a einai h timh
        %opou to x einai 0 (h opoia 8a einai h expected)
        %disp(yValues');
        %disp(xValues');

        %plot(xValues, yValues);
        %pause;
        %Define the query points to be a finer sampling over the range of x.
        xq = -numOfPastValues:0;

        %we can use pchip or spline...
        figure
        vq2 = interp1(xValues,yValues,xq,'pchip','extrap');
        %vq2 = interp1(xValues,yValues,xq,'spline','extrap');

        expectedZero = vq2(end);

        tempExpected = horzcat(tempExpected, expectedZero);
    end

end

if (method == 2) || (method == 3)
    %use this line when extrapolating/interpolating
    xExpected = tempExpected';
end

end

%we can get the dataAfterPseudoPca even for one line by standartizing
%it and multiplying with coeffs from PCA: = zscore(dataBeforePca)*coeffTruncated;

```

```

%dataBeforePca = data(current_start:current_end, :);
%dataAfterPseudoPca = zscore(dataBeforePca)*coeffTruncated;

%create new start and end for next iteration
current_start = current_end+1;
current_end = current_end + pca_length;
end

end

```

calBinTable.m

```

function [Bin] = calBinTable(xOld, xExpected, errorBoundary)

%vazoume 1 an einai h diafora megalyterh apo N% enw 0 an den einai
tempSubtract = ( xOld - xExpected);
tempBin = [];

for indexError = 1:size(tempSubtract)
    tempError = tempSubtract(indexError)/xOld(indexError);
    %kane abs
    if (abs(tempError) < errorBoundary)
        tempBin = horzcat(tempBin, 0);
    else
        tempBin = horzcat(tempBin, 1);
    end
end
Bin = tempBin;

end

```

calBinTableVector.m

```

function [Bin] = calBinTableVector(xOld, xExpected, errorBoundary)

%vazoume 1 an einai h diafora megalyterh apo N% enw 0 an den einai
tempSubtract = (xExpected - xOld);

tempBin = 0;

totalAbsError = [];
%size(tempSubtract)
for indexError = 1:size(tempSubtract)
    %an einai mhden h diafora tote to error einai 0
    if (tempSubtract(indexError) ~= 0)
        tempError = tempSubtract(indexError)/xOld(indexError);
    else
        tempError = 0;
    end

    %tempError
    %pros8ese to absError
    totalAbsError = horzcat(totalAbsError, abs(tempError));
    %totalAbsError
end
%totalAbsError

```

```
%diarese me to plh8os twn deigmatwn
%averageError = mean(totalAbsError,2);
%exclude Inf and NaN
averageError = mean(totalAbsError(isfinite(totalAbsError)),2);

%averageError - kane abs
if (averageError < errorBoundary)
    tempBin = 0;
else
    tempBin = 1;
end
Bin = tempBin;

end
```

calMseTable.m

```
function [Bin] = calMseTable(xExpected, xJustCame)

tempSubtract = (xJustCame - xExpected);

Bin = tempSubtract.^2;

end
```

getNumberOfPCForMinimumExplainedValue.m

```
function [num] = getNumberOfPCForMinimumExplainedValue(explained,
minimumExplainedValue)
num = 1;
explainedTotal = explained(1);

for index = 2:size(explained)
    tempValue = explained(index);

    if (explainedTotal >= minimumExplainedValue)
        return;
    end

    explainedTotal = explainedTotal + tempValue;
    num = num + 1;
end
```

predict.m

```
%Lamvanontas ws eisodo to pinaka A kai ton sintelesth w apo to montelo
%arfit, ka8ws epishs kai ta teleytaia p samples tou v, provlepei to
%epomeno sample tou v
function [pr_v]=predict(v, p, A, w,m,lim)
sum_ =zeros(m,1);
pastValuesToDisplay = [];
for(i=1:p)
    sum_ =sum_ +A(:,m*(i-1)+1:m*i)*v(lim-i,:);
%first accessed is newer
```

```
%disp(v(lim-i,:));  
pastValuesToDisplay = vertcat( v(lim-i,:),pastValuesToDisplay);  
end  
%disp(pastValuesToDisplay);  
pr_v=sum_+w;
```

ΑΝΑΦΟΡΕΣ

- [1] Weiser M. : The computer for the 21st century. Scientific American, September 1991, 66-75
- [2] Katz, R.H., Long, D., Satyanarayanan, M.m Tripathi, S.Workspaces in the Information Age. In Report of the NSF Workshop on Workspaces in the Information Age. Leesburg, VA, October, 1996. Available at <http://www.cs.berkeley.edu/~randy/NSFWS>.
- [3] Weiser, M, Brown, J.S. The Coming Age of Calm Technology. In Denning, P.J., Metcalfe, R.M. (editors), Beyond Calculation: The Next Fifty Years of Computing. Copernicus, 1998.
- [4] D. Abowd, Towards a Better Understanding of Context and Context-Awareness, in Proc. International Conference of Human Factors in Computing Systems, (Hague, Netherlands, 2000)
- [5] P.Prekop and M. Burnett, "Activities, Context and Ubiquitous Computing", Computer Communications, Vol. 26 pp.1168-1176, 2003
- [6] C. Intanagonwiwat, R. Govindan, D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks", Proceedings of the ACM Mobi-Com'00, Boston, MA, 2000, pp.56-67.
- [7] C. Perkins, "Ad Hoc Networks", Addison-Wesley, Reading, MA, 2000.
- [8] Anagnostopoulos, C., Hadjiefthymiades, S., Katsikis, A., and Maglogiannis, I. (2014) Autoregressive energy-efficient context forwarding in wireless sensor networks for pervasive healthcare systems. *Personal and Ubiquitous Computing*, 18(1). pp. 101-114.
- [9] C. Anagnostopoulos, S. Hadjiefthymiades, "Context Compression: using Principal Component Analysis for Efficient Wireless Communications" , IEEE International Conference on Mobile Data Management (MDM 2011), Lulea, Sweden, 2011.
- [10] A. Neumaier and T. Schneider, 2001: Estimation of parameters and eigenmodes of multivariate autoregressive models. ACM Trans. Math. Softw., 27, 27-57.
- [11] T. Schneider and A. Neumaier, 2001: Algorithm 808: ARfit - A Matlab package for the estimation of parameters and eigenmodes of multivariate autoregressive models. ACM Trans. Math. Softw., 27, 58-65.
- [12] Lutkepohl, H." Introduction to Multiple Time Series Analysis (2nd ed.)". Springer-Verlag, Berlin, 1993.
- [13] Akaike, H. "Autoregressive model fitting for control" Ann. Inst. Statist. Math. 23,163-180. 1971
- [14] Schwarz, G.. "Estimating the dimension of a model". Ann. Statist. 6, 461- 464, 1978.
- [15] Lutkepohl, H. "Comparison of criteria for estimating the order of a vector autoregressive process" J. Time Ser. Anal. 6, 35-52. Correction, 8 (1987), 373, 1985.