



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Συμπύεση Πληροφορίας Πλαισίου: χρήση του
Αυτοπαλίνδρομου Υποδείγματος για επίτευξη Αποδοτικών
Ασύρματων Επικοινωνιών**

Αντώνιος Ν. Κατσίκης

**Επιβλέποντες: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής
Χρήστος Αναγνωστόπουλος, Επίκουρος Καθηγητής**

ΑΘΗΝΑ

Νοέμβριος 2011

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Συμπύεση Πληροφορίας Πλαισίου: χρήση του Αυτοπαλίνδρομου Υποδείγματος για επίτευξη Αποδοτικών Ασύρματων Επικοινωνιών

Αντώνιος Ν. Κασίκης

A.M.: M0986

ΕΠΙΒΛΕΠΟΝΤΕΣ: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής
Χρήστος Αναγνωστόπουλος, Επίκουρος Καθηγητής

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής
Λάζαρος Μεράκος, Καθηγητής

Νοέμβριος 2011

ΠΕΡΙΛΗΨΗ

Τα Συστήματα Επίγνωσης Πλαισίου αποτελούν μια ταχέως αναπτυσσόμενη ερευνητική περιοχή με εφαρμογή σε πολλούς τομείς της βιομηχανίας. Στα σύγχρονα συστήματα, αποτελεί αναπόσπαστο τμήμα αυτών μια ασύρματη δικτυακή υποδομή, η οποία προωθεί την πληροφορία πλαισίου από το φαινόμενο-χώρο που παρατηρείται έως την εφαρμογή. Η τελευταία θα την επεξεργαστεί και θα προσαρμόσει ανάλογα την λειτουργία του συστήματος. Αυτές οι δικτυακές υποδομές, αποτελούνται από αισθητήρες, οι όποιοι όπως είναι γνωστό χαρακτηρίζονται από περιορισμένα ενεργειακά (και όχι μόνο) αποθέματα.

Στόχος αυτής της εργασίας είναι να εισάγει νέους μηχανισμούς προώθησης της πληροφορίας πλαισίου, που αποσκοπούν στην όσο το δυνατόν αποτελεσματικότερη διαχείριση των ενεργειακών αποθεμάτων των Ασύρματων Δικτύων Αισθητήρων, ούτως ώστε να αυξήσουμε την διάρκεια ζωής του Συστήματος Επίγνωσης Πλαισίου.

Σχεδιάζουμε, υλοποιούμε και προσομοιώνουμε δυο νέους μηχανισμούς. Οι μηχανισμοί αυτοί, προκειμένου να έχουν τα επιθυμητά αποτελέσματα, εφαρμόζουν μια διαδικασία συμπίεσης της πληροφορίας πλαισίου. Η συμπίεση αυτή επιτυγχάνεται κάνοντας χρήση του αυτοπαλίνδρομου διανυσματικού μοντέλου και της ανάλυσης κυρίων συνιστωσών. Μια συγκριτική εκτίμηση με άλλους μηχανισμούς εξοικονόμησης ενέργειας, παρατίθεται προκειμένου να καταδείξει την ανωτερότητα των προτεινόμενων μεθόδων.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Συστήματα Επίγνωσης Πλαισίου

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Συμπύεση Πληροφορίας Πλαισίου, Ενεργειακή Αποδοτικότητα, Ασύρματα Δίκτυα Αισθητήρων, Μηχανισμός Έξυπνης Προώθησης, Διανυσματικό Αυτοπαλίνδρομο Υπόδειγμα

ABSTRACT

Context-Aware Systems (CAS) is a fast-growing research area with application in various sectors of industry. Contemporary systems, typically, contain a wireless network infrastructure, which forwards the contextual information from the observed area to the application. The latter will process that and adjust its behavior, correspondingly. That network infrastructure is composed of sensors that hold limited energy resources.

Our focus is on the proposal of a new scheme, which efficiently manage energy consumption in Wireless Sensor Networks (WSN) for the purpose of longer CAS's lifetime.

We design, implement and simulate two novel schemes. These models, in order to gain the desired results, compress contextual information. This compression is achieved by autoregressive vector model and principal components analysis. A comparative assessment with other energy efficient models is reported demonstrating the applicability and superiority of the proposed methods.

SUBJECT AREA: Context-Aware Systems (CAS)

KEYWORDS: Contextual Information Compression, Energy Efficiency, Wireless Sensor Networks (WSN), Intelligent Forwarding Scheme, Vector Autoregressive Model

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ	7
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	9
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ	10
ΠΡΟΛΟΓΟΣ	11
1 ΕΙΣΑΓΩΓΗ.....	12
1.1 Συστήματα επίγνωσης πλαισίου	12
1.2 Ασύρματα δίκτυα αισθητήρων.....	13
1.3 Σκοπός της διπλωματικής εργασίας	14
2 ΣΥΜΠΙΕΣΗ ΠΛΑΙΣΙΟΥ ΒΑΣΙΣΜΕΝΗ ΣΤΟ ΑΥΤΟΠΑΛΙΝΔΡΟΜΟ ΥΠΟΔΕΙΓΜΑ (AUTOREGRESSION-BASED CONTEXT COMPRESSION - ARCC).....	16
2.1 Σχεδίαση-Υλοποίηση Μηχανισμού.....	16
2.2 Υπόδειγμα Αυτοπαλίνδρομου Διανύσματος.....	19
2.3 Προσομοίωση - Εκτίμηση Απόδοσης.....	20
3 ΣΥΜΠΙΕΣΗ ΠΛΑΙΣΙΟΥ ΒΑΣΙΣΜΕΝΗ ΣΤΗΝ ΑΝΑΛΥΣΗ ΚΥΡΙΩΝ ΣΥΝΙΣΤΩΣΩΝ (PRINCIPAL COMPONENTS-BASED CONTEXT COMPRESSION - PC3).....	34
3.1 Τρόπος λειτουργίας του PC3	34
3.2 Σύγκριση Απόδοσης ARCC – PC3.....	37
4 ΣΥΜΠΙΕΣΗ ΠΛΑΙΣΙΟΥ ΒΑΣΙΣΜΕΝΗ ΣΤΗΝ ΑΝΑΛΥΣΗ ΚΥΡΙΩΝ ΣΥΝΙΣΤΩΣΩΝ ΚΑΙ ΣΤΟ ΑΥΤΟΠΑΛΙΝΔΡΟΜΟ ΥΠΟΔΕΙΓΜΑ (PRINCIPAL COMPONENT-AUTOREGRESSION- BASED CONTEXT COMPRESSION - PCARCC).....	46
4.1 Σχεδίαση-Υλοποίηση του PCARCC.....	46
4.2 Εκτίμηση και σύγκριση απόδοσης του PCARCC.....	49
5 ΣΥΜΠΕΡΙΦΟΡΑ ΤΩΝ ΜΗΧΑΝΙΣΜΩΝ ARCC, PC3 ΚΑΙ PCARCC ΜΕΤΑΒΑΛΛΟΝΤΑΣ ΤΙΣ ΣΤΑΘΕΡΕΣ (p, m, l, MAX_{error})	59
5.1 Μεταβλητή μνήμη διαδικασίας VAR (p)	59
5.1.1 Μνήμη διαδικασίας VAR (p) – Ενεργειακό κόστος.....	59
5.1.2 Μνήμη διαδικασίας VAR (p) – Μέσο Σφάλμα ανακατασκευής.....	60
5.2 Μεταβλητή φάση μάθησης (m)	61
5.2.1 Φάση μάθησης (m) - Ενεργειακό κόστος	61
5.2.2 Φάση μάθησης (m) - Μέσο Σφάλμα ανακατασκευής.....	62
5.3 Μεταβλητή φάση συμπίεσης (l)	63
5.3.1 Φάση συμπίεσης (l) - Ενεργειακό κόστος.....	63
5.3.2 Φάση συμπίεσης (l) - Μέσο Σφάλμα ανακατασκευής.....	64
5.4 Μεταβλητό σφάλμα πρόβλεψης.....	65
5.4.1 Μέγιστο σφάλμα πρόβλεψης (MAX_{error}) - Ενεργειακό κόστος	65

5.4.2	Μέγιστο σφάλμα πρόβλεψης (MAX_{error}) - Μέσο Σφάλμα ανακατασκευής.....	66
5.5	Ενεργειακό Κόστος – Σφάλμα ανακατασκευής	67
6	ΣΥΜΠΕΡΑΣΜΑΤΑ	69
	ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ	70
	ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ	71
	ΠΑΡΑΡΤΗΜΑ Ι	72
	ΠΑΡΑΡΤΗΜΑ ΙΙ.....	75
	ΠΑΡΑΡΤΗΜΑ ΙΙΙ	76
	ΑΝΑΦΟΡΕΣ	95

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Τα δεδομένα πλαισίου που χρησιμοποιούνται στην πειραματική διαδικασία (α) υγρασία (β) θερμοκρασία (γ) ταχύτητα ανέμου.....	23
Σχήμα 2: Το ενεργειακό κόστος που καταναλίσκεται σε κάθε μηχανισμό	24
Σχήμα 3: (α) ποσοστιαίο ενεργειακό κέρδος της ARCC σε σχέση με την SDF, (β) σχετικό σφάλμα πρόβλεψης στο κόμβο i σε κάθε διάνυσμα, (γ) σχετικό σφάλμα ανακατασκευής σε κάθε διάνυσμα πλαισίου, (δ) Εξέλιξη του μέσου σχετικού σφάλματος ανακατασκευής.....	26
Σχήμα 4: Οι τιμές της (α) θερμοκρασίας, (β) υγρασίας και (γ) ταχύτητας ανέμου στα πραγματικά δεδομένα πλαισίου που έχει αποθηκεύσει ο κόμβος i (μπλε) σε σχέση με αυτά που αποθηκεύει ο κόμβος j (πράσινο) για $Max_{error} = 10\%$ (δ) οι τιμές της θερμοκρασίας στα πραγματικά δεδομένα πλαισίου που έχει αποθηκεύσει ο κόμβος i (μπλε) σε σχέση με αυτά που αποθηκεύει ο κόμβος j (πράσινο) για $Max_{error} = 1\%$	30
Σχήμα 5: Συντελεστής απόδοσης W , του μηχανισμού ARCC και SDF	32
Σχήμα 6: Συντελεστής απόδοσης f , των μηχανισμών ARCC και SDF	33
Σχήμα 7: Το ενεργειακό κόστος που καταναλίσκεται σε κάθε μηχανισμό	38
Σχήμα 8: Ποσοστιαίο ενεργειακό κέρδος των ARCC (μπλε) και PC3 (πράσινο) σε σχέση με την SDF	40
Σχήμα 9: Το σχετικό σφάλμα ανακατασκευής σε κάθε διάνυσμα πλαισίου στους μηχανισμούς ARCC (μπλε) και PC3 (πράσινο)	41
Σχήμα 10: Οι τιμές της θερμοκρασίας στα πραγματικά δεδομένα πλαισίου που έχει αποθηκεύσει ο κόμβος (μπλε) σε σχέση με αυτά που αποθηκεύει ο κόμβος (πράσινο) στον μηχανισμό PC3	42
Σχήμα 11: Εξέλιξη του μέσου σχετικού σφάλματος ανακατασκευής τους μηχανισμούς PC3 (πράσινο) και ARCC (μπλε).....	43
Σχήμα 12: Συντελεστής απόδοσης για τους μηχανισμούς ARCC (πράσινο), PC3 (μπλε) και SDF (κόκκινο)	44
Σχήμα 13: Συντελεστής απόδοσης για τους μηχανισμούς ARCC (κόκκινο), PC3 (πράσινο) και SDF (μπλε).....	45
Σχήμα 14: Το ενεργειακό κόστος που καταναλίσκεται σε κάθε μηχανισμό	50
Σχήμα 15: Ποσοστιαίο ενεργειακό κέρδος των ARCC (πράσινο), PC3 (μπλε) και PCARCC (κόκκινο) σε σχέση με την SDF.....	52
Σχήμα 16: Το σχετικό σφάλμα ανακατασκευής σε κάθε ένα διάνυσμα πλαισίου, στους μηχανισμούς ARCC (κόκκινο), PCARCC (μπλε) και PC3 (πράσινο) .	53
Σχήμα 17: Εξέλιξη του μέσου σχετικού σφάλματος ανακατασκευής στους μηχανισμούς PC3(πράσινο), PCARCC (κόκκινο) και ARCC (μπλε)	54
Σχήμα 18: Πλήθος αυτούσιων προωθούμενων διανυσμάτων, σημάτων ανακατασκευής και μέσο σφάλμα πρόβλεψης σε κάθε μια από τις περιόδους του PCARCC	55

Σχήμα 19: Συντελεστής απόδοσης w για τους μηχανισμούς ARCC (πράσινο), PC3 (μπλε), PCARCC (κόκκινο) και SDF (θαλασσί).....	56
Σχήμα 20: Συντελεστής απόδοσης f για τους μηχανισμούς ARCC (μπλε), PC3 (θαλασσί), PCARCC (κόκκινο) και SDF (πράσινο)	57
Σχήμα 21: Σχέση μεταξύ μνήμης διαδικασίας VAR και Ενεργειακού κόστους για τους μηχανισμούς PCARCC(θαλασσί), PC3(πράσινο), SDF (μπλε) και ARCC (κόκκινο)	59
Σχήμα 22: Σχέση μεταξύ μνήμης διαδικασίας VAR () και μέσου σφάλματος ανακατασκευής για τους μηχανισμούς PCARCC(κόκκινο), PC3(μπλε), και ARCC (πράσινο).....	60
Σχήμα 23: Σχέση μεταξύ φάσης μάθησης και ενεργειακού κόστους για τους μηχανισμούς PCARCC(θαλασσί), PC3(πράσινο), SDF (μπλε) και ARCC (κόκκινο)	61
Σχήμα 24: Σχέση μεταξύ φάσης μάθησης m και μέσου σφάλματος ανακατασκευής για τους μηχανισμούς PCARCC(κόκκινο), PC3(μπλε), και ARCC (πράσινο) ...	62
Σχήμα 25: Σχέση μεταξύ φάσης συμπίεσης και ενεργειακού κόστους για τους μηχανισμούς PCARCC(θαλασσί), PC3(πράσινο), SDF (μπλε) και ARCC (κόκκινο)	63
Σχήμα 26: Σχέση μεταξύ φάσης συμπίεσης I και μέσου σφάλματος ανακατασκευής για τους μηχανισμούς PCARCC(κόκκινο), PC3(μπλε), και ARCC (πράσινο) ...	64
Σχήμα 27: Σχέση μεταξύ μεγίστου σφάλματος πρόβλεψης και ενεργειακού κόστους για τους μηχανισμούς PCARCC(θαλασσί), PC3(πράσινο), SDF (μπλε) και ARCC (κόκκινο).....	65
Σχήμα 28: Σχέση μεταξύ μεγίστου σφάλματος πρόβλεψης και μέσου σφάλματος ανακατασκευής για τους μηχανισμούς PCARCC(κόκκινο), PC3(μπλε), και ARCC (πράσινο)	66
Σχήμα 29: Σχέση εξάρτησης ανάμεσα στο κόστος και στο σφάλμα ανακατασκευής στον ARCC	67
Σχήμα 30: Σχέση εξάρτησης ανάμεσα στο κόστος και στο σφάλμα ανακατασκευής στον PC3	68
Σχήμα 31: Σχέση εξάρτησης ανάμεσα στο κόστος και στο σφάλμα ανακατασκευής στον PCARCC	68

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Κάθε κόμβος αποστέλλει, είτε τα πραγματικά δεδομένα $dc(t)$ ή ένα σήμα $u(t)=1$, ανακατασκευής στον επόμενο κόμβο	17
Εικόνα 2: Ο κόμβος i , την χρονική στιγμή t , αποφασίζει το περιεχόμενο της επόμενης αποστολής του.....	18
Εικόνα 3: Μηχανισμός Συμπίεσης πλαισίου PC3	34
Εικόνα 4: Φάσεις Λειτουργίας του μηχανισμού PC3	35
Εικόνα 5: Διαδικασίες που εκτελούνται μέσα στο κόμβο πριν προώθηση το πλαίσιο που έλαβε/ανίχνευσε.....	37
Εικόνα 6: Μηχανισμός συμπίεσης πλαισίου PCARCC	46
Εικόνα 7: Φάσεις Λειτουργίας μηχανισμού PCARCC	48

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Κόστη των διάφορων διαδικασιών στο μοντέλο Mica2	20
---	----

ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο φοίτησης μου, στο μεταπτυχιακό πρόγραμμα σπουδών του τμήματος Πληροφορικής και Τηλεπικοινωνιών, ΕΚΠΑ, στην κατεύθυνση “Δικτύων και Συστημάτων Επικοινωνίας”.

Βασική μου επιθυμία ήταν να ασχοληθώ και να μελετήσω έναν τομέα, που βρίσκεται στο προσκήνιο της ερευνητικής δραστηριότητας, τα αποτελέσματα της οποίας θα αποτελέσουν την βάση των μελλοντικών τεχνολογικών επιτευγμάτων και εξελίξεων. Η ευρεία περιοχή των συστημάτων επίγνωσης πλαισίου, όπως αυτή συναντάται σε διάφορες πτυχές της βιομηχανίας της Πληροφορικής, κάλυπτε τον πρωταρχικό μου αυτό σκοπό.

Θα ήθελα να ευχαριστήσω σε αυτό το σημείο, τον Επίκ. Καθηγητή κ. Ευστάθιο Χατζηευθυμιάδη, που μου έδωσε την δυνατότητα να ασχοληθώ με το συγκεκριμένο θέμα και χάραξε τους βασικούς άξονες πάνω στους οποίους κινήθηκε η εργασία. Επίσης, τον Επίκ. Καθηγητή κ. Χρήστο Αναγνωστόπουλο για την συστηματική και συντεταγμένη καθοδήγηση και υποστήριξη, που μου παρείχε, καθ' όλη την διάρκεια μελέτης και εκπόνησης της εργασίας.

1 ΕΙΣΑΓΩΓΗ

1.1 Συστήματα επίγνωσης πλαισίου

Οι αναδυόμενες και ταχύτατα εξελισσόμενες τα τελευταία χρόνια τεχνολογίες διάχυτου υπολογισμού (pervasive computing), παρουσιάζουν ευρεία αποδοχή, γιατί χαρακτηρίζονται από το γεγονός πως μπορούν να προσφέρουν υπηρεσίες σε οποιονδήποτε, οποιαδήποτε στιγμή, οπουδήποτε και αν βρίσκεται. Οι χρήστες αυτών των τεχνολογιών, προσδοκούν να λάβουν ποιοτικές και κατάλληλες υπηρεσίες, βασιζόμενες στις οντότητες που τους περιβάλλουν, χωρίς εκείνοι να χρειάζεται να ασχοληθούν στο ελάχιστο με αυτές (τις υπηρεσίες).

Στα συμβατικά συστήματα, ο χρήστης ήταν αναγκασμένος να ζητάει με σαφήνεια κάποια υπηρεσία. Ένα σύστημα επίγνωσης πλαισίου (context aware-system), είναι αυτό που με κατάλληλο τρόπο, διαδραστικά και αυτόνομα προσαρμόζεται, ούτως ώστε να παρέχει κατάλληλες υπηρεσίες ή περιεχόμενο στους χρήστες του, χρησιμοποιώντας αυτό που αποκαλούμε πληροφορία πλαισίου (contextual information), ελαχιστοποιώντας την παρέμβαση που χρειάζεται να κάνει ο ίδιος ο χρήστης. Για παράδειγμα, ένα τέτοιο σύστημα θα μπορούσε να ρυθμίζει τον ηλεκτρικό φωτισμό ενός χώρου, ανάλογα με την ένταση της φωτεινότητας που υπάρχει στο χώρο από εξωγενείς παράγοντες, πετυχαίνοντας έτσι την μέγιστη εξοικονόμηση ενέργειας και χρημάτων, παρέχοντας ταυτόχρονα αποτελεσματικό φωτισμό στους χρήστες του χώρου.

Βασικό στοιχείο αυτών των συστημάτων είναι ο όρος πλαίσιο (context). Ένας κοινά αποδεκτός και αντιπροσωπευτικός ορισμός για το πλαίσιο είναι αυτός των Dey και Abowd [1] που αναφέρει πως πλαίσιο : «είναι οποιαδήποτε πληροφορία μπορεί να χρησιμοποιηθεί για να περιγράψει την κατάσταση μιας οντότητας (πρόσωπο, αντικείμενο, τοποθεσία), η οποία (οντότητα) σχετίζεται με την αλληλεπίδραση μεταξύ του χρήστη και της εφαρμογής, συμπεριλαμβανομένου του χρήστη και της εφαρμογής». Από αυτό τον ορισμό γίνεται αντιληπτό πως το πλαίσιο είναι ένα σύνολο καταστάσεων και ενεργειών. Αυτές οι καταστάσεις μεταβάλλονται συνεχώς κατά την διάρκεια του χρόνου σαν αποτέλεσμα ενεργειών που εφαρμόζονται, περιγράφοντας έτσι την συμπεριφορά των χρηστών, και τις συνθήκες του περιβάλλοντος και των εφαρμογών.

Στα συστήματα επίγνωσης πλαισίου διακρίνονται τρεις οντότητες [2] . Αυτές είναι οι τοποθεσίες (δωμάτια, κτήρια), τα πρόσωπα (άτομα, ομάδες) και τα αντικείμενα (φυσικά αντικείμενα, υπολογιστικά εξαρτήματα). Κάθε μία από αυτές τις οντότητες μπορεί να περιγραφεί χρησιμοποιώντας διάφορες μεταβλητές, οι οποίες με τη σειρά τους διαχωρίζονται σε τέσσερις κατηγορίες: ταυτότητας (κάθε οντότητα έχει ένα μοναδικό χαρακτηριστικό), τοποθεσίας (η θέση στο χώρο κάθε οντότητας), κατάστασης (ή δραστηριότητας, περιγράφει τα φυσικά χαρακτηριστικά της οντότητας π.χ. θερμοκρασία και ένταση φωτεινότητας σε ένα δωμάτιο, διεργασίες που εκτελούνται τώρα σε μια συσκευή) και χρονικές μεταβλητές (χρησιμοποιούνται για να αποτυπώσουν την χρονική στιγμή που λαμβάνει χώρα ένα γεγονός).

Μια διαδεδομένη κατηγοριοποίηση των μορφών που μπορεί να έχει το πλαίσιο, είναι αυτή που το διακρίνει σε εξωτερικό (ή φυσικό) και εσωτερικό (ή λογικό) [2]. Η πρώτη κατηγορία, αυτή του εξωτερικού πλαισίου είναι και η πιο συνηθισμένη. Αφορά το πλαίσιο που μπορεί να μετρηθεί με ειδικά συστήματα αισθητήρων, όπως θέσης, φωτός, ήχου, κίνησης, αφής, θερμοκρασίας, υγρασίας, ταχύτητας ανέμου. Δηλαδή οι πληροφορίες πλαισίου που είναι άμεσα μετρήσιμες. Από την άλλη, το εσωτερικό πλαίσιο καθορίζεται ως επί το πλείστον από τον χρήστη ή ανιχνεύεται παρακολουθώντας τις αλληλεπιδράσεις του χρήστη, δηλαδή τους στόχους, τα καθήκοντα, το πλαίσιο εργασίας, τις επιχειρηματικές διαδικασίες, και την ψυχολογική κατάσταση του χρήστη.

Το πρώτο στάδιο που πρέπει να εκτελεσθεί ένα σύστημα επίγνωσης πλαισίου, δεν είναι άλλο από το να εντοπιστεί με κάποιο τρόπο το πλαίσιο, ούτως ώστε στη συνέχεια να ερμηνευθεί κατάλληλα και να προσαρμοστεί αντίστοιχα στο περιβάλλον του. Ο εντοπισμός αυτός λοιπόν, από τον οποίο εκκινούν όλα τα υπόλοιπα, γίνεται ως επί το πλείστον με μεμονωμένους αισθητήρες ή ασύρματα δίκτυα αισθητήρων.

1.2 Ασύρματα δίκτυα αισθητήρων

Παράλληλα με την ανάπτυξη στα συστήματα διάχυτου υπολογισμού, παρουσιάστηκε πρόοδος στην τεχνολογία των μικρό-ηλεκτρομηχανικών συστημάτων (ΜΗΜΣ), στην ασύρματη επικοινωνία και στα ψηφιακά ηλεκτρονικά. Αυτό έδωσε την δυνατότητα για την ανάπτυξη κόμβων αισθητήρων χαμηλού-κόστους, χαμηλής κατανάλωσης ενέργειας και πολλών λειτουργιών, οι οποίοι είναι μικροί σε μέγεθος και επικοινωνούν, χωρίς ανθρώπινη παρέμβαση ή επιτήρηση, μεταξύ τους σε μικρές αποστάσεις. Αυτοί οι μικροσκοπικοί κόμβοι αισθητήρων, που αποτελούνται από εξαρτήματα ανίχνευσης, επεξεργασίας δεδομένων και επικοινωνίας, οδηγούν στην ιδέα δικτύων αισθητήρων που βασίζονται στην συνεργατική λειτουργία ενός μεγάλου συνόλου κόμβων. Τα δίκτυα αισθητήρων αντιπροσωπεύουν μια σημαντική εξέλιξη έναντι των παραδοσιακών αισθητήρων, οι οποίοι εγκαθίστανται με τους ακόλουθους δύο τρόπους [3] :

- Εγκατάσταση αισθητήρων μακριά από το πραγματικό φαινόμενο. Με αυτήν την προσέγγιση απαιτούνται μεγάλοι αισθητήρες που χρησιμοποιούν πολύπλοκες τεχνικές για να διακρίνουν τους στόχους από τον θόρυβο του περιβάλλοντος.
- Εγκατάσταση αισθητήρων που εκτελούν μόνο αισθητήρια εργασία. Η θέση και η επικοινωνιακή τοπολογία σχεδιάζονται προσεκτικά. Αυτοί εκπέμπουν μια σειρά του υπό παρακολούθηση φαινομένου στους κεντρικού κόμβους όπου εκτελούνται οι υπολογισμοί και συγχωνεύονται τα δεδομένα.

Ένα δίκτυο αισθητήρων αποτελείται από ένα μεγάλο αριθμό κόμβων αισθητήρων, οι οποίοι αναπτύσσονται πυκνά, είτε μέσα στο φαινόμενο-χώρο ,που θέλουν να παρατηρήσουν, είτε πολύ κοντά σε αυτό. Η θέση των κόμβων αισθητήρων δεν είναι ανάγκη να προσχεδιαστεί ή να προαποφασιστεί. Αυτό επιτρέπει την τυχαία εξάπλωσή σε μη προσβάσιμα εδάφη ή σε επιχειρήσεις αντιμετώπισης καταστροφών. Από την άλλη πλευρά, αυτό σημαίνει ότι τα πρωτοκολλά και οι αλγόριθμοι των δικτύων αισθητήρων πρέπει να διαθέτουν αυτό-οργανωτικές δυνατότητες. Ένα άλλο μοναδικό χαρακτηριστικό των δικτύων αισθητήρων είναι η συνεργατική λειτουργία των κόμβων αισθητήρων. Οι αισθητήριοι κόμβοι εξοπλίζονται με έναν επεξεργαστή. Αντί να στέλνουν ακατέργαστα δεδομένα στους κόμβους που είναι υπεύθυνοι για την συγχώνευση των δεδομένων, οι αισθητήριοι κόμβοι χρησιμοποιούν τις δυνατότητες επεξεργασίας που διαθέτουν προκειμένου να εκτελέσουν τοπικά απλούς υπολογισμούς και να εκπέμπουν μόνο τα επεξεργασμένα δεδομένα. Το παραπάνω χαρακτηριστικό εξασφαλίζει ένα μεγάλο πλήθος εφαρμογών για δίκτυα αισθητήρων. Μερικές από τις περιοχές εφαρμογής είναι η υγεία, ο στρατός και η ασφάλεια. Για παράδειγμα, οι σωματικές μετρήσεις (πυρετός, αρτηριακή πίεση κτλ.) ενός ασθενή μπορούν να παρατηρούνται απομακρυσμένα από το γιατρό του. Αυτό είναι και πιο βολικό για τον ασθενή, αλλά επίσης επιτρέπει και στο γιατρό να έχει πλήρη εικόνα της παρούσας κατάστασης του ασθενή και της εξέλιξης της υγείας του. Οι αισθητήριοι κόμβοι μπορούν επίσης να χρησιμοποιηθούν για να εντοπίσουν ξένα χημικά στοιχεία στον αέρα και το νερό. Μπορούν να βοηθήσουν για να ανακαλύψουν τον τύπο, την συγκέντρωση και την θέση μολυσματικών ουσιών.

Προκειμένου να υλοποιηθούν οι παραπάνω αλλά και άλλες εφαρμογές των δικτύων αισθητήρων απαιτούνται τεχνικές ad-hoc. Παρόλο που αρκετοί αλγόριθμοι και πρωτοκολλά έχουν προταθεί για τα παραδοσιακά ad-hoc ασύρματα δίκτυα, δυστυχώς δεν είναι δυνατόν να χρησιμοποιηθούν προκειμένου να καλύψουν τα μοναδικά χαρακτηριστικά και τις απαιτήσεις των εφαρμογών των δικτύων αισθητήρων. Παρακάτω αναφέρονται περιληπτικά οι διαφορές μεταξύ των δύο αυτών δικτύων που δικαιολογεί το παραπάνω πρόβλημα [4]:

- Ο αριθμός των κόμβων σε ένα δίκτυο αισθητήρων μπορεί να είναι πολλές φορές πιο μεγάλος από ότι σε ένα ad hoc δίκτυο.
- Η χωρική πυκνότητα των δικτύων αισθητήρων είναι συχνά μεγάλη.
- Οι αισθητήριοι κόμβοι είναι εύκολο να καταστραφούν.
- Η τοπολογία ενός δικτύου αισθητήρων αλλάζει πολύ συχνά.
- Οι αισθητήριοι κόμβοι χρησιμοποιούν κυρίως επικοινωνία broadcast ενώ τα περισσότερα δίκτυα ad hoc βασίζονται στην επικοινωνία σημείου προς σημείο.
- Οι αισθητήριοι κόμβοι διακρίνονται για τους σημαντικούς περιορισμούς που έχουν, από κατασκευής, στους τομείς της ενέργειας, της υπολογιστικής ισχύος και της μνήμης.
- Οι αισθητήριοι κόμβοι συνήθως δεν έχουν κάποιο παγκόσμιο αναγνωριστικό (ID), εξαιτίας του μεγάλου μεγέθους της επικεφαλίδας που απαιτεί μια τέτοια ιδιότητα, καθώς και του μεγάλου αριθμού των κόμβων.

Ένας από τα πιο σημαντικούς περιορισμούς στα δίκτυα ασύρματων αισθητήρων είναι η απαίτηση για χαμηλή κατανάλωση ενέργειας. Οι αισθητήριοι κόμβοι έχουν περιορισμένες και συνήθως αναντικατάστατες πηγές ενέργειας. Έτσι ενώ τα παραδοσιακά δίκτυα στοχεύουν να παρέχουν υπηρεσίες υψηλής ποιότητας, τα δίκτυα ασύρματων αισθητήρων έχουν ως πρωταρχικό στόχο την διατήρηση της ενέργειας. Επίσης θα πρέπει να έχουν ένα μηχανισμό που θα δίνει στον χρήστη του δικτύου την επιλογή να παρατείνει την ζωή του δικτύου με αντάλλαγμα την μικρότερη ρυθμαπόδοση ή την μεγαλύτερη καθυστέρηση στην μετάδοση.

1.3 Σκοπός της διπλωματικής εργασίας

Ο εντοπισμός λοιπόν, της πληροφορίας πλαισίου από τα συστήματα επίγνωσης ή διαφορετικά η συλλογή και προώθηση πληροφοριών πλαισίου από έναν ασύρματο δίκτυο αισθητήρων προς τον κόμβο συλλέκτη (sink node), ο οποίος επικοινωνεί άμεσα με την εφαρμογή, αποτελεί ένα ανοιχτό και ιδιαίτερος ενδιαφέρον θέμα. Από την στιγμή που υπάρχουν περιορισμένοι, προς χρήση, πόροι, όπως για παράδειγμα η ενέργεια των κόμβων, η υπολογιστική τους ισχύς, ο αποθηκευτικός τους χώρος και το εύρος ζώνης του δικτύου, θα ήταν αναποτελεσματικό να προωθείται προς τον κόμβο συλλέκτη (εφαρμογή) οποιαδήποτε πληροφορία πλαισίου συλλέγεται από του αισθητήρες. Αυτό το πρόβλημα, αποτελεί για πολλά χρόνια ένα τεράστιο εμπόδιο στο βιομηχανικό και ερευνητικό τομέα των συστημάτων αυτών. Είναι μείζονος σημασίας λοιπόν να λαμβάνουμε υπόψη την φύση της ανιχνευόμενης πληροφορίας πλαισίου, με σκοπό να αποφύγουμε την μεγάλη ενεργειακή κατανάλωση και να έχουμε όσο το δυνατό καλύτερη χρήση του διαθέσιμου εύρους. Σκοπός αυτής της εργασίας είναι να προτείνει τρόπους με τους οποίους μπορεί να επιτευχθεί η όσο το δυνατό αποτελεσματικότερη διαχείριση των ενεργειακών αποθεμάτων των ασύρματου δικτύου αισθητήρων, ούτως ώστε να αυξήσουμε την διάρκεια ζωής του συστήματος επίγνωσης πλαισίου. Σχεδιάζουμε, υλοποιούμε και προσομοιώνουμε δυο νέους μηχανισμούς. Οι μηχανισμοί

αυτοί, προκειμένου να έχουμε τα επιθυμητά αποτελέσματα, εφαρμόζουν μια διαδικασία συμπίεσης της πληροφορίας πλαισίου. Η συμπίεση αυτή επιτυγχάνεται χρησιμοποιώντας δυο μαθηματικά εργαλεία: (i) το αυτοπαλίνδρομο διανυσματικό μοντέλο και (ii) την ανάλυση κυρίων συνιστωσών.

2 ΣΥΜΠΙΕΣΗ ΠΛΑΙΣΙΟΥ ΒΑΣΙΣΜΕΝΗ ΣΤΟ ΑΥΤΟΠΑΛΙΝΔΡΟΜΟ ΥΠΟΔΕΙΓΜΑ (AUTOREGRESSION-BASED CONTEXT COMPRESSION - ARCC)

2.1 Σχεδίαση-Υλοποίηση Μηχανισμού

Σκοπός της προτεινόμενης μεθόδου είναι η αύξηση της διάρκειας ζωής του ασύρματου δικτύου αισθητήρων, μέσω της καλύτερης διαχείρισης των ενεργειακών αποθεμάτων των κόμβων, που όπως είναι γνωστό είναι περιορισμένα, αφού τροφοδοτούνται ως επί το πλείστον από μπαταρίες. Για να γίνει το δίκτυο ενεργειακά αποδοτικό και συνεπώς να αυξηθεί η διαθεσιμότητα του, εφαρμόζεται σε κάθε κόμβο, είτε ανίχνευσης (sensor node) ή μεταφοράς (relay node), μια τεχνική αξιολόγησης των δεδομένων πλαισίου που πρόκειται να αποσταλούν. Αυτή η τεχνική βασίζεται στο αυτοπαλίνδρομο πολυμεταβλητό μοντέλο (Vector Autoregressive Model - VAR). Μέσω αυτής της διαδικασίας ο κάθε κόμβος αποφασίζει για τον αν χρειάζεται να αποστείλει στον επόμενο (upstream) όλη την πληροφορία, που είτε ανίχνευσε ο ίδιος (κόμβος ανιχνευτής) ή λήφθηκε από εκείνον (κόμβος μεταφοράς), προκειμένου να την μεταδώσει με την σειρά του είτε σε κάποιον επόμενο κόμβο μεταφοράς ή στον κόμβο συλλέκτη, ολοκληρώνοντας κατά αυτόν τον τρόπο μια ροή πληροφορίας (data flow).

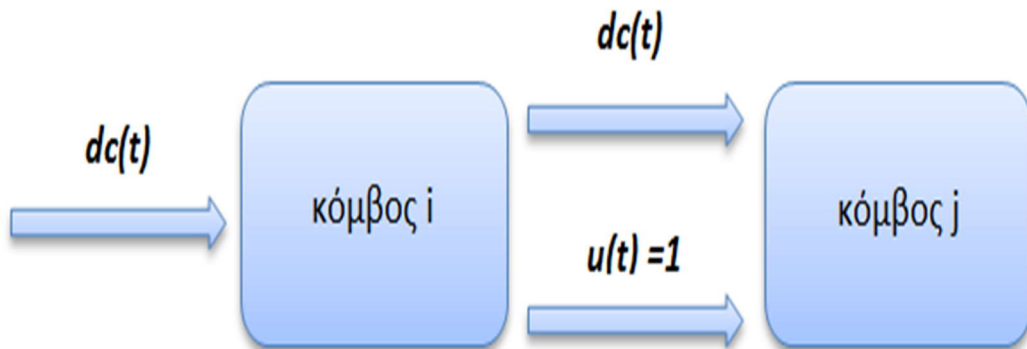
Το συγκεκριμένο μοντέλο βασίζεται στην πολυμεταβλητή (διανυσματική) φύση της πληροφορίας πλαισίου, που καλείται να διαχειριστεί ένα ασύρματο δίκτυο αισθητήρων. Επίσης, ένας ακόμη παράγοντας που ευνοεί αυτή την διαδικασία είναι πως οι μεταβλητές αυτές σχετίζονται μεταξύ τους. Κάθε κόμβος εφαρμόζοντας τον μηχανισμό ARCC, πετυχαίνει την όσο το δυνατό μεγαλύτερη εξοικονόμηση ενέργειας και συνεπώς αυξάνει την διάρκεια ζωής του και κατ'επέκταση την διαθεσιμότητα του δικτύου. Ένας κόμβος μεταφοράς χρειάζεται να δεσμεύει μνήμη και επικοινωνιακούς πόρους (αποστολή και λήψη) για κάθε ροή δεδομένων πλαισίου. Ένας κόμβος ανίχνευσης χρειάζεται να δεσμεύει πόρους για ανίχνευση και προώθηση. Επίσης πολύ συχνά ένας κόμβος μεταφοράς εφαρμόζει μηχανισμούς συνάθροισης όταν από αυτόν περνάνε πολλές ροές πληροφορίας.

Έστω λοιπόν πως $dc(t)$, είναι η πληροφορία πλαισίου, που έχει στην κατοχή του ο κόμβος i , την χρονική στιγμή t , και θέλει να την αποστείλει στον επόμενο κόμβο j . Ο κόμβος i , μπορεί είτε να αποστείλει ολόκληρο το $dc(t)$ ή ένα σήμα $u(t) \in \{0,1\}$, στον κόμβο j με το οποίο θα τον ενημερώνει πως πρέπει να αναπαράγει μόνος του την πραγματική πληροφορία πλαισίου, $dc(t)$. Στην τελευταία περίπτωση το $u(t) = 1$, ενώ στη πρώτη $u(t) = 0$. Προκειμένου να πάρει αυτή την απόφαση ο κόμβος i , υποθέτουμε πως όλοι οι κόμβοι του δικτύου έχουν συμφωνήσει εκ των προτέρων στην χρησιμοποίηση ενός συγκεκριμένου αυτοπαλίνδρομου μοντέλου VAR. Ο κόμβος i , που καλείται να επιλέξει ποιο θα είναι το περιεχόμενο της αποστολής προς τον j , με την βοήθεια του αυτοπαλίνδρομου μοντέλου υπολογίζει – προβλέπει μια τιμή $dc^*(t)$, για τα πραγματικά δεδομένα $dc(t)$, βασιζόμενο στις τελευταίες p μετρήσεις που ανίχνευσε (ή έλαβε) $dc(t-p), \dots, dc(t-1), p > 0$. Το αυτοπαλίνδρομο μοντέλο $VAR(p)$, εξαρτάται από αυτές τις p μετρήσεις, οι οποίες αποκαλούνται και «μνήμη» του μοντέλου. Η τιμή του $dc^*(t)$ συγκρίνεται με τα πραγματικά δεδομένα $dc(t)$, και υπολογίζεται το σφάλμα πρόβλεψης $e(t) = dc(t) - dc^*(t)$. Αυτό το εκτιμώμενο σφάλμα είναι που καθορίζει το τι θα επιλέξει ο κόμβος i . Αν ο κόμβος i , επιλέξει να μην προωθήσει στον j , τα πραγματικά δεδομένα, τότε ο τελευταίος θα πρέπει με την σειρά του να εκτελέσει μια αντίστοιχη διαδικασία πρόβλεψης, κάνοντας χρήση του $VAR(p)$ επίσης, υπολογίζοντας κατά αυτό

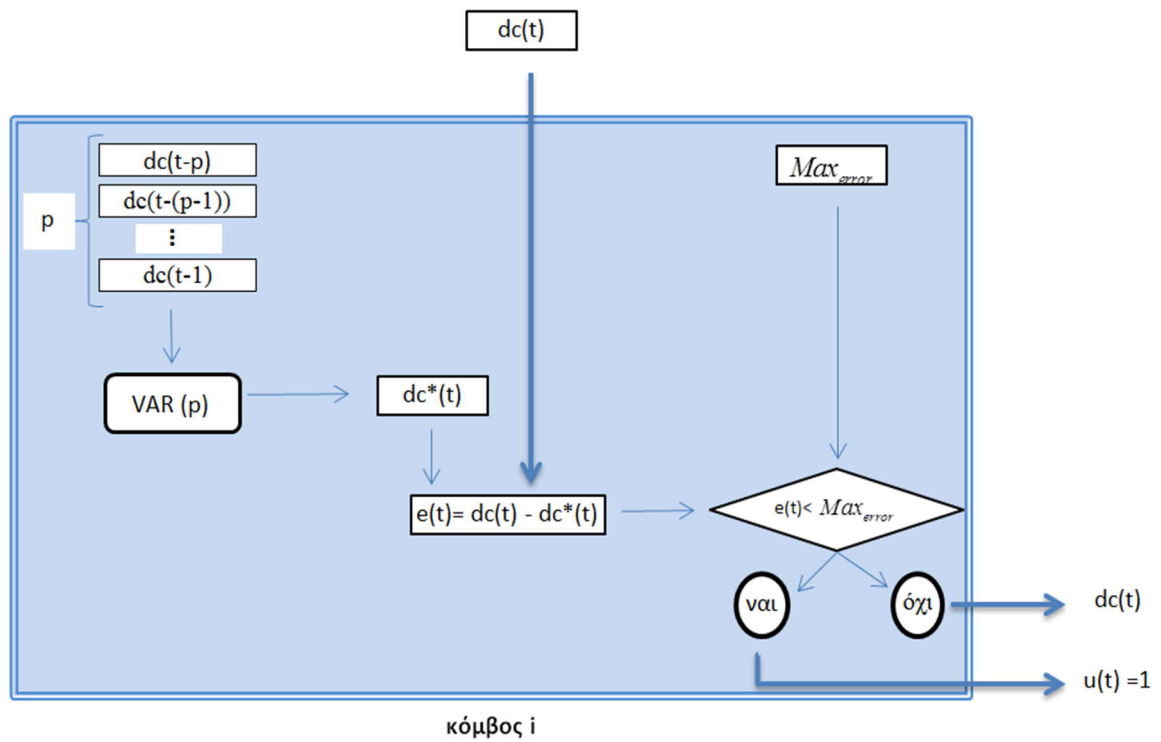
τον τρόπο το $dc'(t)$, δηλαδή την ληφθείσα μέτρηση. Αυτή η τακτική ακολουθείται για όλες τις ροές δεδομένων, που διέρχονται από τον κόμβο i .

Αξίζει να σημειωθεί ότι ο προτεινόμενος μηχανισμός, προσπαθεί να εκμεταλλευτεί την διανυσματική φύση των ανταλλασσόμενων πληροφοριών και την πιθανή στατιστική εξάρτηση μεταξύ διαφορετικών δεδομένων πλαισίου. Το παρόν σχήμα βασίζεται πάνω σε αυτή την υπόθεση, η οποία συχνά συναντάται σε πραγματικές εφαρμογές και ολοκληρωμένα συστήματα.

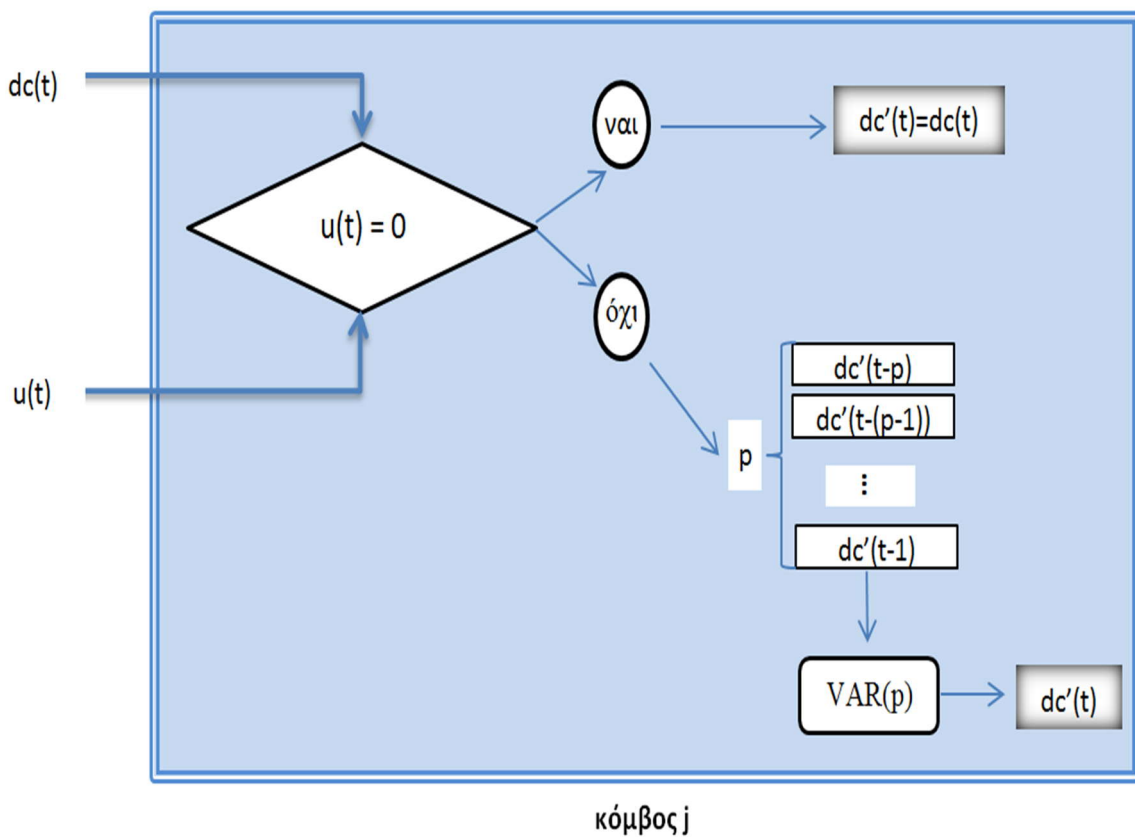
Οι διαδικασίες αποστολής και λήψης ανάμεσα σε δυο κόμβους i και j , καθώς και οι μηχανισμοί που εφαρμόζονται εσωτερικά σε κάθε κόμβο φαίνονται στις παρακάτω εικόνες (1, 2 & 3), αναλυτικότερα :



Εικόνα 1: Κάθε κόμβος αποστέλλει, είτε τα πραγματικά δεδομένα $dc(t)$ ή ένα σήμα, $u(t)=1$, ανακατασκευής στον επόμενο κόμβο



Εικόνα 2: Ο κόμβος i, την χρονική στιγμή t, αποφασίζει το περιεχόμενο της επόμενης αποστολής του



Εικόνα 3: Ο κόμβος j, την χρονική στιγμή t, ανάλογα τα δεδομένα που έλαβε, αποφασίζει ποιο είναι το περιεχόμενο της πληροφορίας πλαισίου την χρονική στιγμή t

2.2 Υπόδειγμα Αυτοπαλίνδρομου Διανύσματος

Βασικό στοιχείο της όλης διαδικασίας, και παράγων ζωτικής σημασίας για την επιτυχία της, είναι ο μηχανισμός πρόβλεψης που εφαρμόζεται, το αυτοπαλίνδρομο διάνυσμα. Το Υπόδειγμα Αυτοπαλίνδρομου Διανύσματος (Vector Autoregression - VAR) είναι ένα από τα πιο επιτυχημένα, ευέλικτα, και εύκολα στη χρήση πρότυπα για την ανάλυση πολυμεταβλητών χρονοσειρών. Στην πράξη αποτελεί μια φυσική επέκταση του υποδείγματος αυτοπαλινδρόμησης μιας μεταβλητής για χρήση σε δυναμική πολυμεταβλητή χρονοσειρά. Το υπόδειγμα VAR έχει αποδειχθεί ότι είναι ιδιαίτερα χρήσιμο στην περιγραφή δυναμικών συμπεριφορών των οικονομικών και χρηματοοικονομικών χρονοσειρών και για την πρόβλεψη αυτών. Παρέχει συχνά ανώτερη πρόβλεψη σε σχέση με τα αντίστοιχα μονομεταβλητά μοντέλα. Οι προβλέψεις από το μοντέλο VAR είναι αρκετά ευέλικτες, διότι μπορεί να εξαρτώνται από την πιθανή μελλοντική πορεία των μεταβλητών που καθορίζονται στο μοντέλο [5].

Ας ορίσουμε το $Y_t = (y_{1t}, y_{2t}, \dots, y_{nt})$, σαν έναν διάνυσμα διαστάσεων $(nx1)$, που αποτελεί ένα στιγμιότυπο μια πολυμεταβλητής χρονοσειράς, για την χρονική στιγμή t .

Η βασική σχέση, που δείχνει την εξάρτηση της χρονοσειράς την στιγμή t , από τις παρελθοντικές χρονοσειρές σε βάθος p , έχει την εξής δομή:

$$Y_t = c + \Pi_1 Y_{t-1} + \Pi_2 Y_{t-2} + \dots + \Pi_p Y_{t-p} + \varepsilon_t, t = 1, \dots, T$$

Όπου, Π_i είναι πίνακες συντελεστών, διαστάσεων (nxn) και ε_t εκφράζει τα σφάλμα της πρόβλεψης. Για παράδειγμα σε ένα δι-μεταβλητό μοντέλο, με παράγων υστέρησης (p) ίσο με 2, η παραπάνω εξίσωση θα έπαιρνε την εξής μορφή:

$$\begin{pmatrix} y_{1t} \\ y_{2t} \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} \pi_{11}^1 & \pi_{12}^1 \\ \pi_{21}^1 & \pi_{22}^1 \end{pmatrix} \begin{pmatrix} y_{1t-1} \\ y_{2t-1} \end{pmatrix} + \begin{pmatrix} \pi_{11}^2 & \pi_{12}^2 \\ \pi_{21}^2 & \pi_{22}^2 \end{pmatrix} \begin{pmatrix} y_{1t-2} \\ y_{2t-2} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix}$$

ή

$$y_{1t} = c_1 + \pi_{11}^1 y_{1t-1} + \pi_{12}^1 y_{2t-1} + \pi_{11}^2 y_{1t-2} + \pi_{12}^2 y_{2t-2} + \varepsilon_{1t}$$

$$y_{2t} = c_2 + \pi_{21}^1 y_{1t-1} + \pi_{22}^1 y_{2t-1} + \pi_{21}^2 y_{1t-2} + \pi_{22}^2 y_{2t-2} + \varepsilon_{2t}$$

Αξίζει να σημειωθεί, ότι η παραπάνω διεργασία επιφέρει έναν επιπλέον υπολογιστικό κόστος στην διαδικασία αποστολής και λήψης ενός πλαισίου δεδομένων από έναν κόμβο σε έναν επόμενο, το οποίο προφανώς και θα λάβουμε υπόψη μας αργότερα, όταν θα αξιολογήσουμε τον προτεινόμενο μηχανισμό.

2.3 Προσομοίωση - Εκτίμηση Απόδοσης

Σε αυτό το τμήμα, θα εξετάσουμε το κατά πόσο ο μηχανισμός ARCC επιτυγχάνει αποδοτική προώθηση των δεδομένων πλαισίου, εις βάρος βέβαια της ακρίβειας αυτών. Το ARCC συγκρίνεται με τον μηχανισμό Απλής Προώθηση Δεδομένων (Simple Data Forwarding - SDF), στο οποίο ο κάθε κόμβος προωθεί στον επόμενο, χωρίς καμία αλλαγή τα δεδομένα που λαμβάνει.

Οι πειραματικές διαδικασίες έχουν διεξαχθεί, κάνοντας χρήση πραγματικών δεδομένων ανίχνευσης (Παράρτημα I), σε δίκτυα αισθητήρων. Το κάθε πλαίσιο αποτελείται από ένα διάνυσμα 7 μεταβλητών με περιεχόμενο (θερμοκρασία1, υγρασία1, θερμοκρασία2, υγρασία2, θερμοκρασία3, υγρασία3, ταχύτητα ανέμου), υπάρχουν συνολικά 387 δείγματα, τα οποία έχουν δειγματοληπτηθεί με συχνότητα 10^{-2} Hz.

Για την προσομοίωση της προώθησης της πληροφορίας πλαισίου από τον κόμβο i στον j , χρησιμοποιήθηκε το Matlab. Στο περιβάλλον αυτό υλοποιήθηκε ο μηχανισμός ARCC (Παράρτημα III) ενώ έγινε χρήση ενός ετοιμού πακέτου συναρτήσεων, του arfit (Παράρτημα II), οι οποίες χρησιμοποιήθηκαν για την υλοποίηση του μοντέλου VAR

Υιοθετούμε το μοντέλο κατανάλωσης ενέργειας Mica2 [6]. Το Mica2 λειτουργεί με ένα ζεύγος μπαταριών AA, το οποίο προσεγγιστικά παρέχει 2200 mAh με μέση ενεργή τάση 3V. Καταναλώνει 20mA αν εκτελεί μια εφαρμογή ανίχνευσης συνεχώς, το οποίο οδηγεί σε διάρκεια ζωής 100 ωρών.

Στο παρακάτω πίνακα, φαίνεται το ενεργειακό κόστος, που καταναλίσκεται για μία εντολή CPU (ενέργεια ανά εντολή) και για την αποστολή/λήψη των περιεχομένων πλαισίου (ενέργεια ανά bit). Η πληροφορία τοποθετείται σε πακέτα, κάθε ένα από τα οποία έχει μήκος επικεφαλίδας 7 bytes. Το μέγεθος που δεσμεύουν από το πακέτο τα δεδομένα που στέλνουμε έχει ως εξής: 4bytes για την αποστολή κάθε τιμής πραγματικών δεδομένων (θερμοκρασία, υγρασία, ταχύτητα) και 1bit για την αποστολή του σήματος ανακατασκευής.

Τρόπος Λειτουργία Κόμβου	Ενεργειακό Κόστος
Εκτέλεση εντολής	4 nJ/εντολή
Αδρανής - Αναμονή	9.6 mJ/s – 0.33mJ/s
Αποστολή - Λήψη	720 nJ/bit – 110 nJ/bit

Πίνακας 1: Κόστη των διάφορων διαδικασιών στο μοντέλο Mica2

Επίσης, όπως αναφέραμε και προηγουμένως θα πρέπει να ενσωματώσουμε στην όλη διαδικασία το κόστος που προκαλεί σε αυτή, η υλοποίηση της VAR. Συνεπώς, το συνολικό κόστος $c(t)$ σε Joule την χρονική στιγμή t για έναν κόμβο i , αναπαρίσταται ως εξής:

$$c(t) = c(t-1) + c_R(t) + c_T(t) + c_I(t) + c_0(t) \quad (1)$$

Όπου $c_R(t)$, $c_T(t)$ είναι το κόστος των διαδικασιών λήψης και αποστολής αντίστοιχα, είτε για τα πραγματικά δεδομένα $dc(t)$ ή για το σήμα ανακατασκευής $u(t)$. $c_I(t)$ είναι το ενεργειακό κόστος, για την εκτέλεση των εντολών της CPU, κατά την υλοποίηση του μηχανισμού VAR. $c_0(t)$ είναι το κόστος της αρχικής κατάστασης του κόμβου i . Στόχος

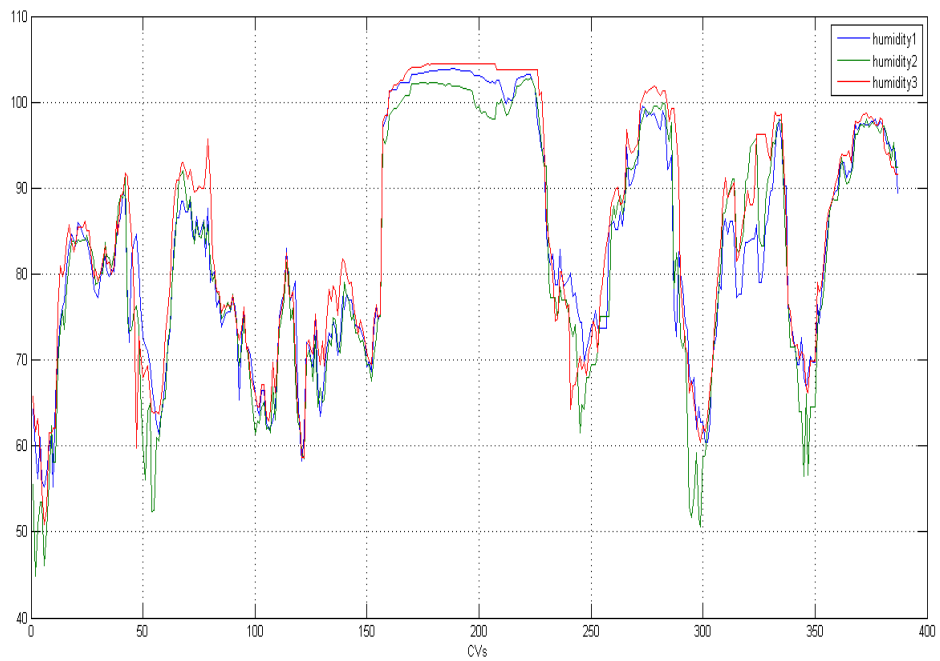
είναι να εξοικονομήσουμε ενέργεια , κρατώντας σε όσο το δυνατό χαμηλότερο επίπεδο το άθροισμα $c_R(t) + c_T(t)$, με το να μεταδίδουμε περισσότερα σήματα ανακατασκευής $u(t)$ (1 bit + 1 bit) , αντί για τα πραγματικά δεδομένα πλαισίου $dc(t)$ (7x4 bytes + 7x4 bytes), εις βάρος του επιπρόσθετου κόστους πράξεων $c_I(t)$, που υφιστάμεθα, και της ακρίβειας των δεδομένων. Η ακρίβεια των δεδομένων πλαισίου, εκτιμάται μέσω του σφάλματος αναπαραγωγής $e(t) = dc_i(t) - dc_j(t)$, για κάθε χρονική στιγμή t . Όπου $dc_i(t)$ είναι το πλαίσιο που προώθησε ο κόμβος i την χρονική στιγμή t και $dc_j(t)$ το πλαίσιο που αποθήκευσε (έλαβε ή προέβλεψε) ο κόμβος j για την ίδια χρονική στιγμή.

Ορίζουμε ως ποσοστιαίο ενεργειακό κέρδος (percentage cost gain), $pcg(t)_{arcc-sdf} \in [0,1]$ την ποσοστιαία εξοικονόμηση που έχουμε στο ενεργειακό κόστος, όταν σε έναν ασύρματο δίκτυο αισθητήρων , εφαρμόζουμε τον μηχανισμό ARCC, αντί για την απλή προώθηση πλαισίων, SDF.

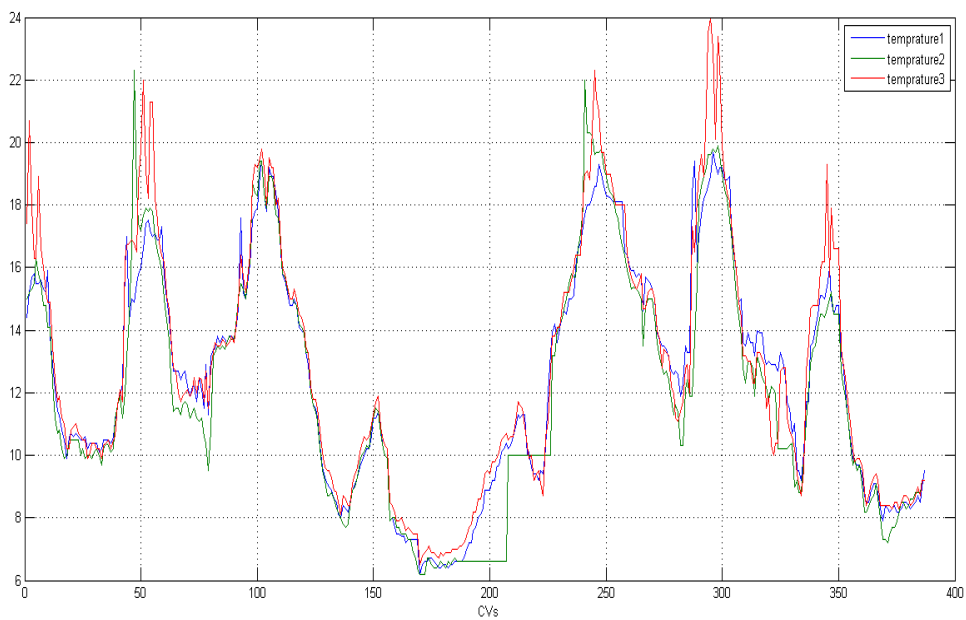
$$pcg(t)_{arcc-sdf} = \frac{c_{sdf}(t) - c_{arcc}(t)}{c_{sdf}(t)}$$

Επιθυμούμε το $pcg(t)_{arcc-sdf}$ να έχει όσο το δυνατόν μεγαλύτερη απόλυτη τιμή, ιδανική περίπτωση να ισούται με 1.

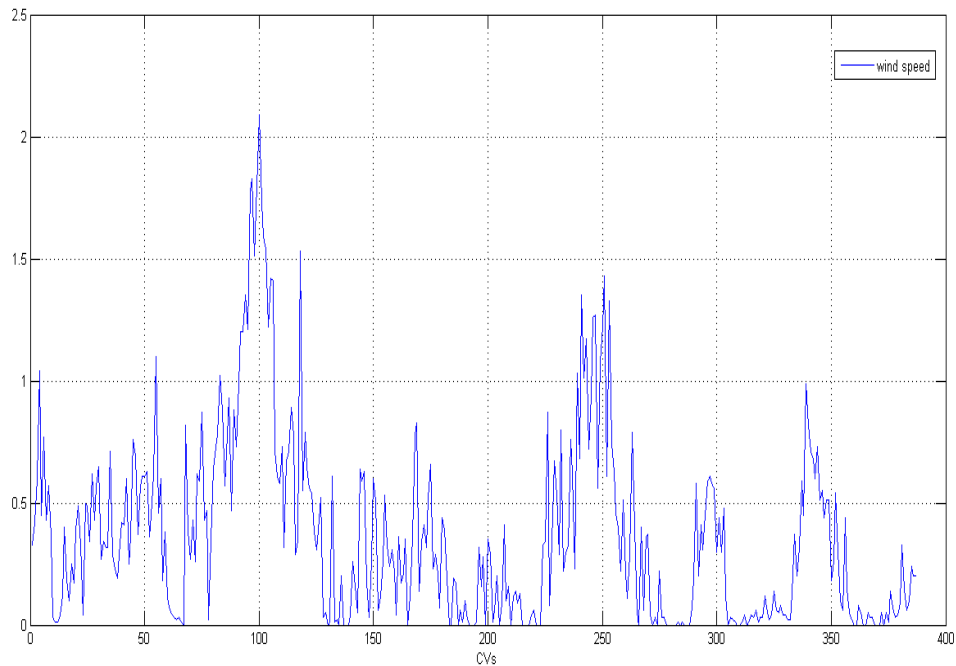
Παρακάτω απεικονίζονται οι γραφικές παραστάσεις των δεδομένων που χρησιμοποιούνται στην πειραματική διαδικασία, χωρισμένα ανάλογα με το είδος τους (υγρασία, θερμοκρασία, ταχύτητα ανέμου) 4(α,β,γ), είναι φανερό, όπως αναφέρθηκε και προηγουμένως, πως υπάρχει συσχέτιση μεταξύ των τιμών ενός δείγματος με τις τιμές επόμενων δειγμάτων, όσον αφορά τις ίδιες μεταβλητές.



(α)



(β)



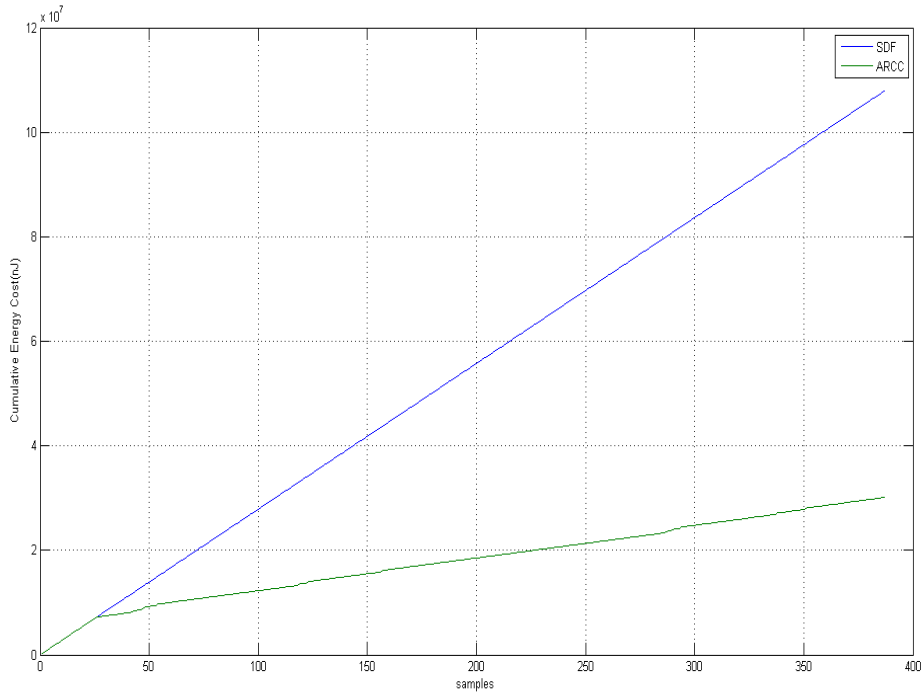
(γ)

Σχήμα 1: Τα δεδομένα πλαισίου που χρησιμοποιούνται στην πειραματική διαδικασία (α) υγρασία (β) θερμοκρασία (γ) ταχύτητα ανέμου

Το ενεργειακό κόστος που καταναλίσκεται στις διαδικασίες SDF και ARCC, φαίνεται στο σχήμα 1, παρατηρούμε πως τα δύο κόστη ταυτίζονται στο ξεκίνημα μέχρι ενός σημείου. Αυτό είναι αναμενόμενο αφού σε εκείνο το διάστημα δεν εφαρμόζεται καθόλου το μοντέλο VAR στην διαδικασία ARCC, αντ' αυτού τα δεδομένα προωθούνται από τον έναν κόμβο στον επόμενο αναλλοίωτα.

Αυτό συμβαίνει επειδή δεν έχει δημιουργηθεί «μνήμη» στον κόμβο αποστολής προκειμένου να μπορεί να εκτελέσει πρόβλεψη μέσω του μοντέλου VAR, και να προσαρμόσει το περιεχόμενο της αποστολής του. Αντίστοιχα ο κόμβος λήψης δεν έχει την «μνήμη» για να ανακατασκευάσει το μη ληφθέν πλαίσιο.

Έπειτα από αυτό το διάστημα «μάθησης» (“learning”), όπως συχνά ονομάζεται διακρίνουμε μια αισθητή μείωση του ενεργειακού κόστους στον μηχανισμό ARCC σε σχέση με την διαδικασία απλής προώθησης δεδομένων. Αυτή η μείωση, οφείλεται στην ελαχιστοποίηση του αθροίσματος $c_R(t) + c_T(t)$ από την εξίσωση υπολογισμού του κόστους (1).



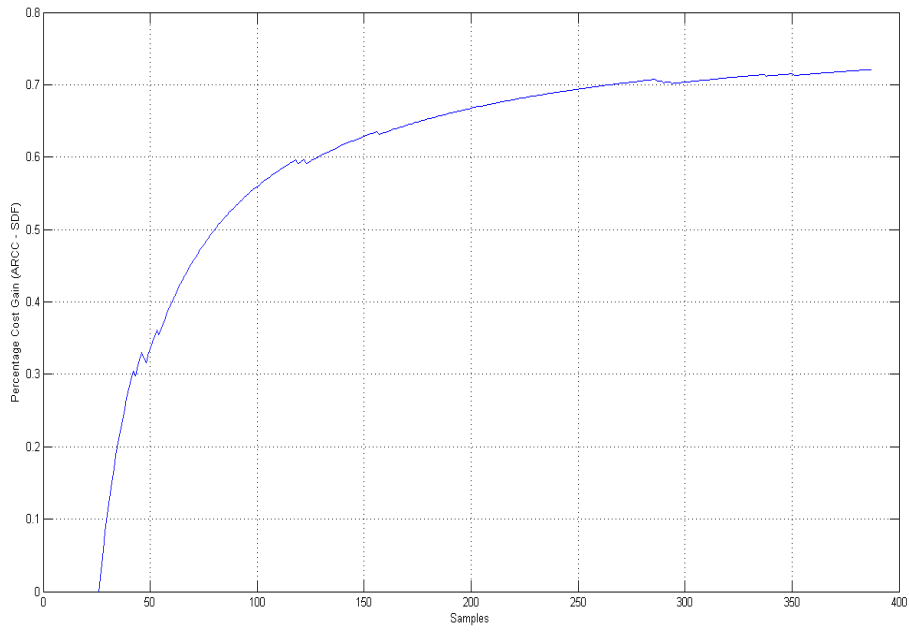
Σχήμα 2: Το ενεργειακό κόστος που καταναλίσκεται σε κάθε μηχανισμό

Αυτό συμβαίνει γιατί όπως αναφέραμε και παραπάνω, αντί ολόκληρων των διανυσμάτων (7 τιμές x 4 byte) ο κόμβος προωθεί στον επόμενο, όταν πληρούνται κάποιες προδιαγραφές, ένα σήμα ανακατασκευής μεγέθους ενός bit. Στο συγκεκριμένο πείραμα, οι προδιαγραφές αυτές (δηλ. $e(t) < Max_{error}$), πληρούνται 349 φορές, από τα 387 δείγματα που αποστέλλει συνολικά (για $Max_{error} = 10\%$ και $\rho=3$). Το 90% δηλαδή των πραγματικών δεδομένων πλαισίου δεν αποστέλλεται αλλά προβλέπεται και ανακατασκευάζεται από τον επόμενο κόμβο.

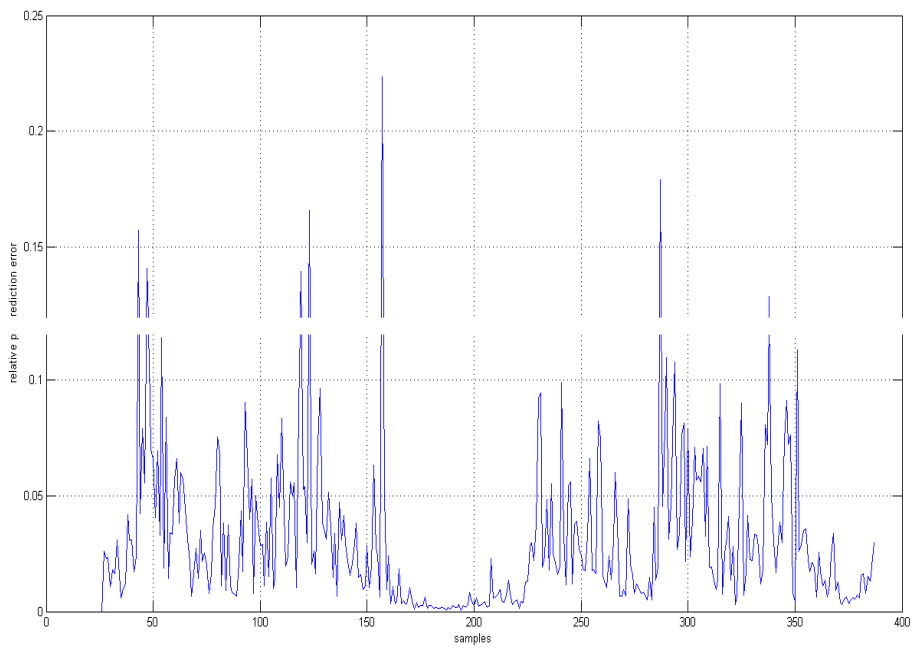
Στο σχήμα 3(α), φαίνεται η εξέλιξη του ποσοστιαίου ενεργειακού κέρδους, σε σχέση με τα διανύσματα πλαισίου που διέρχονται από τον κόμβο αποστολής, παρατηρούμε ότι το όφελος συγκλίνει σε μια συγκεκριμένη τιμή.

Έχουμε λοιπόν 72% εξοικονόμηση ενεργείας χρησιμοποιώντας το ARCC σε σχέση με την απλή προώθηση των διανυσμάτων πλαισίου.

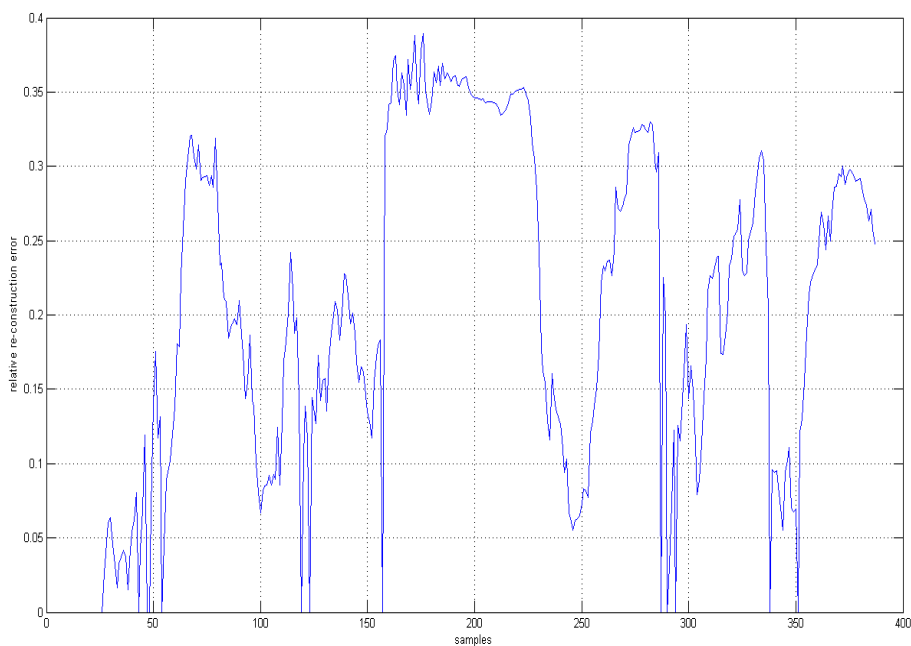
Αυτού του μεγέθους το όφελος καταδεικνύει την καταλληλότητα του προτεινόμενου μοντέλου για την επίλυση του προβλήματος της ενεργειακά αποδοτικής προώθησης δεδομένων.



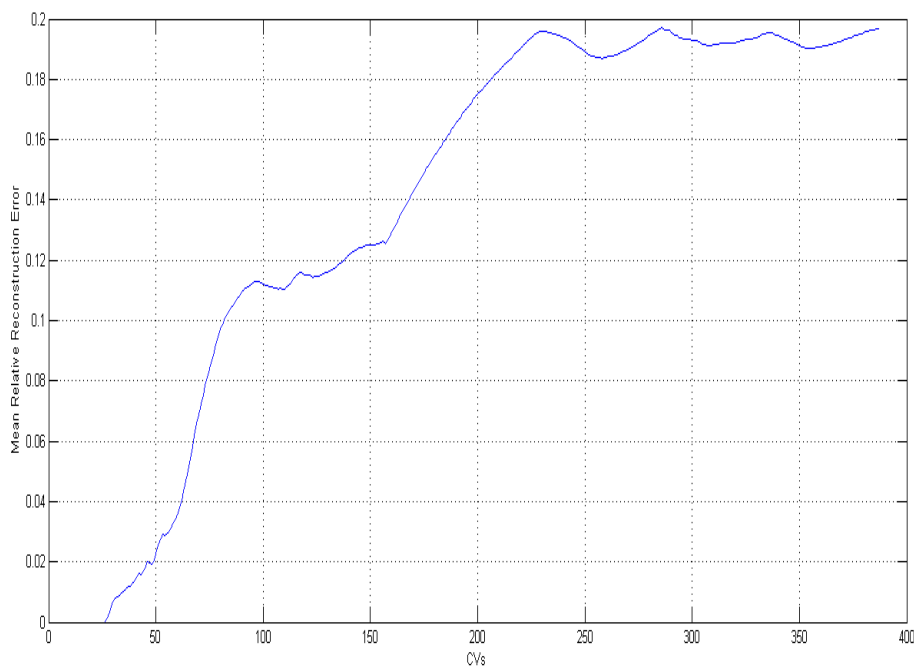
(α)



(β)



(γ)



(δ)

Σχήμα 3: (α) ποσοστιαίο ενεργειακό κέρδος της ARCC σε σχέση με την SDF, (β) σχετικό σφάλμα πρόβλεψης στο κόμβο i σε κάθε διάνυσμα, (γ) σχετικό σφάλμα ανακατασκευής σε κάθε διάνυσμα πλαισίου, (δ) Εξέλιξη του μέσου σχετικού σφάλματος ανακατασκευής

Στα σχήματα 3(β) και 3(γ), απεικονίζονται το σχετικό σφάλμα πρόβλεψης και ανακατασκευής αντίστοιχα. Το σχετικό σφάλμα πρόβλεψης έχει να κάνει με το σφάλμα που εκτιμά ο κόμβος i , σχετικά με μια πρόβλεψη $dc^*(t)$, που έχει κάνει για ένα διάνυσμα πλαισίου $dc(t)$ την χρονική στιγμή t , δηλ. $e_{i-i}(t) = \frac{\|dc(t) - dc^*(t)\|}{\|dc(t)\|}$. Το μέγεθος

αυτής της τιμής καθορίζει και το ποια απόφαση θα πάρει ο κόμβος i , για την χρονική στιγμή t .

Το σχετικό σφάλμα ανακατασκευής, αναφέρεται στο σφάλμα μεταξύ του πραγματικού πλαισίου $dc(t)$ για την χρονική στιγμή t και αυτού που έχει προβλέψει είτε λάβει ο επόμενος κόμβος j δηλ. $dc'(t)$, για την ίδια χρονική στιγμή, και υπολογίζεται ως εξής

$$e_{i-j}(t) = \frac{\|dc(t) - dc'(t)\|}{\|dc(t)\|}.$$

Όπως είναι λογικό, το μέσο σχετικό σφάλμα πρόβλεψης ή ανακατασκευής, είναι η μέση τιμή, των $e_{i-i}(t)$ και $e_{i-j}(t)$ αντίστοιχα, για όλο το χρονικό διάστημα προώθησης των δεδομένων πλαισίου.

Στην παρούσα πειραματική διαδικασία, το μέσο σχετικό σφάλμα πρόβλεψης είναι 2,8%. Αυτή η τιμή, η ακρίβεια δηλαδή της πρόβλεψης που γίνεται στον κόμβο i , επηρεάζεται από το μέγεθος της «μνήμης» p ($p=3$ στο πείραμα), που επιλέγουμε να έχουμε. Η σχέση των δυο είναι αντιστρόφως ανάλογη. Όσο μεγαλύτερη μνήμη έχουμε τόσο καλύτερη πρόβλεψη (συνεπώς μικρότερο σφάλμα) μας δίνει ο μηχανισμός $VAR(p)$. Από την άλλη μεριά, το τίμημα που πληρώνουμε για μία μεγαλύτερη τιμή p (δηλ. μια καλύτερη πρόβλεψη), είναι η αποστολή περισσότερων αναλλοίωτων πλαισίων, στον επόμενο κόμβο, συνεπώς αύξηση του ενεργειακού κόστους της όλης διαδικασίας, άρα υπονόμηση του αρχικού στόχου για ενεργειακά αποδοτική προώθηση δεδομένων πλαισίου.

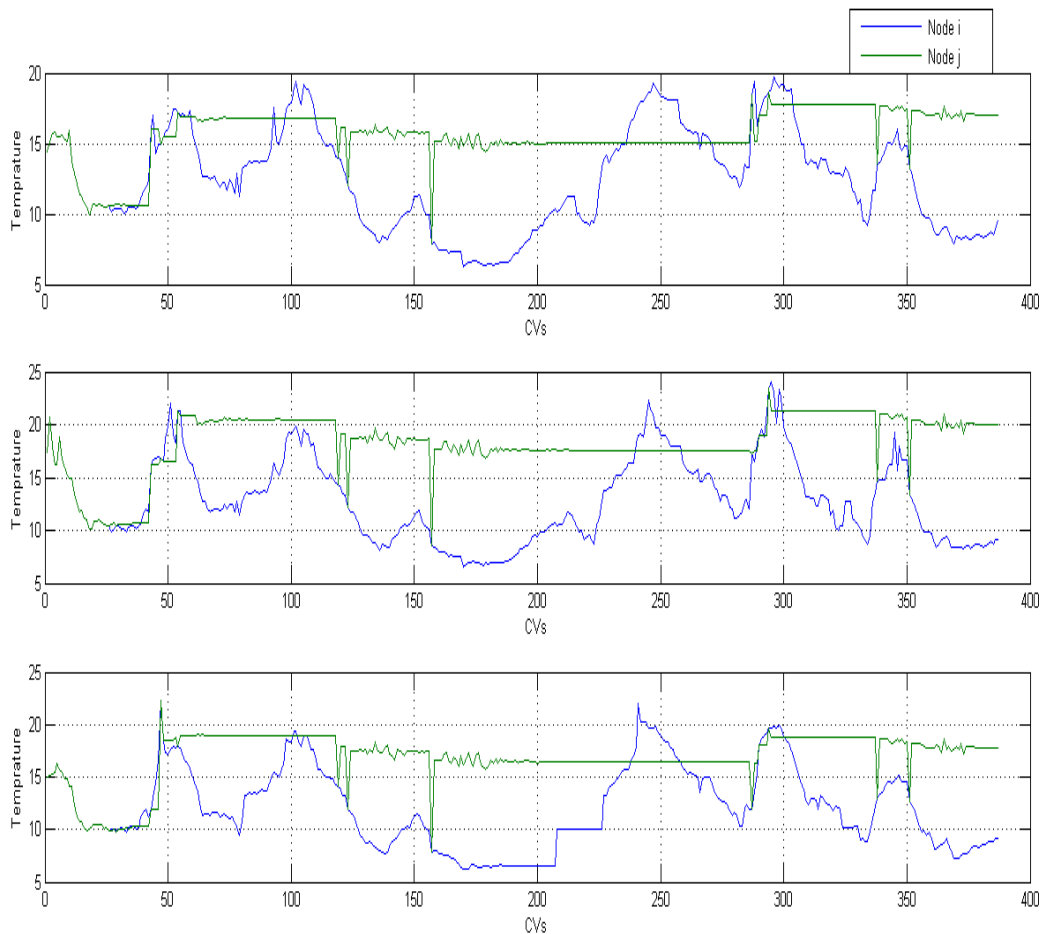
Το μέσο σχετικό σφάλμα ανακατασκευής υπολογίζεται στο 19,68%, αυτή η τιμή μας υποδεικνύει σε τι βαθμό έχουν κοινή εικόνα για τα διανύσματα πλαισίου οι κόμβοι που συμμετέχουν στον μηχανισμό. Ο κόμβος j δηλαδή, κάθε χρονική στιγμή, που λαμβάνει ένα σήμα ανακατασκευής από τον κόμβο i , και όχι το πραγματικό πλαίσιο, εκτίμα μέσω της αντίστοιχης διαδικασίας $VAR(p)$, την τιμή που έχει το πραγματικό πλαίσιο για την συγκεκριμένη χρονική στιγμή. Η εκτίμηση αυτή ως επί το πλείστον είναι διαφορετική από την αντίστοιχη που γίνεται στον κόμβο i , διότι βασίζεται στην μνήμη του j και όχι του i , η οποία μνήμη έχει δημιουργηθεί με ανάλογο τρόπο, δηλαδή μέσω εκτιμήσεων-προσεγγίσεων.

Στο σχήμα 4 (α, β & γ) έχουμε μια οπτική εκτίμηση για το κατά πόσο κοινά είναι τα δεδομένα πλαισίου που έχει αποθηκευμένα ο κόμβος j , τα οποία όπως αναφέραμε μπορεί να έχουν προέλθει από εκτίμηση του j είτε από την λήψη αυτούσιου του πραγματικού πλαισίου από τον i , σε σχέση με τα πραγματικά δεδομένα πλαισίου, τα οποία και έχει αποθηκευμένα στην μνήμη του ο κόμβος i .

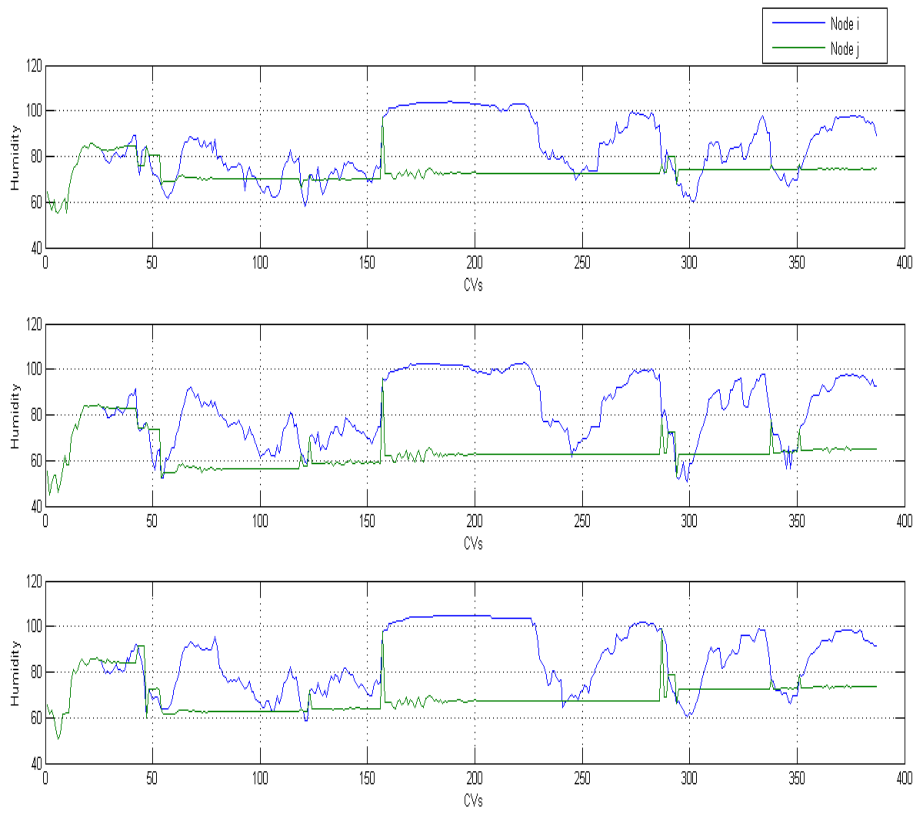
Στο συγκεκριμένο πείραμα, δεδομένου των τιμών που έχουμε θέσει σε κάποιες μεταβλητές (p, Max_{error}) διακρίνουμε και οπτικά πως υπάρχει κάποια σημαντική απόκλιση στις τιμές των δεδομένων που έχει στην κατοχή του ο j σε σχέση με τα πραγματικά δεδομένα, που κατέχει ο i , κάτι το οποίο βεβαίως καταδεικνύεται και υπολογιστικά από το σχετικά υψηλό σφάλμα ανακατασκευής που υπολογίσαμε και παραπάνω.

Το σφάλμα ανακατασκευής στο κόμβο j , συνδέεται με το μέγιστο όριο σφάλματος (error threshold) Max_{error} , της πρόβλεψης που έχουμε επιλέξει για τον κόμβο i . Όσο πιο αυστηρό (μικρότερη τιμή) είναι αυτό το όριο, τόσο μικρότερο είναι το μέσο σχετικό σφάλμα ανακατασκευής και συνεπώς υπάρχει μεγαλύτερη ταύτιση στην εικόνα, που έχουν για τα δεδομένα πλαισίου οι κόμβοι του δικτύου.

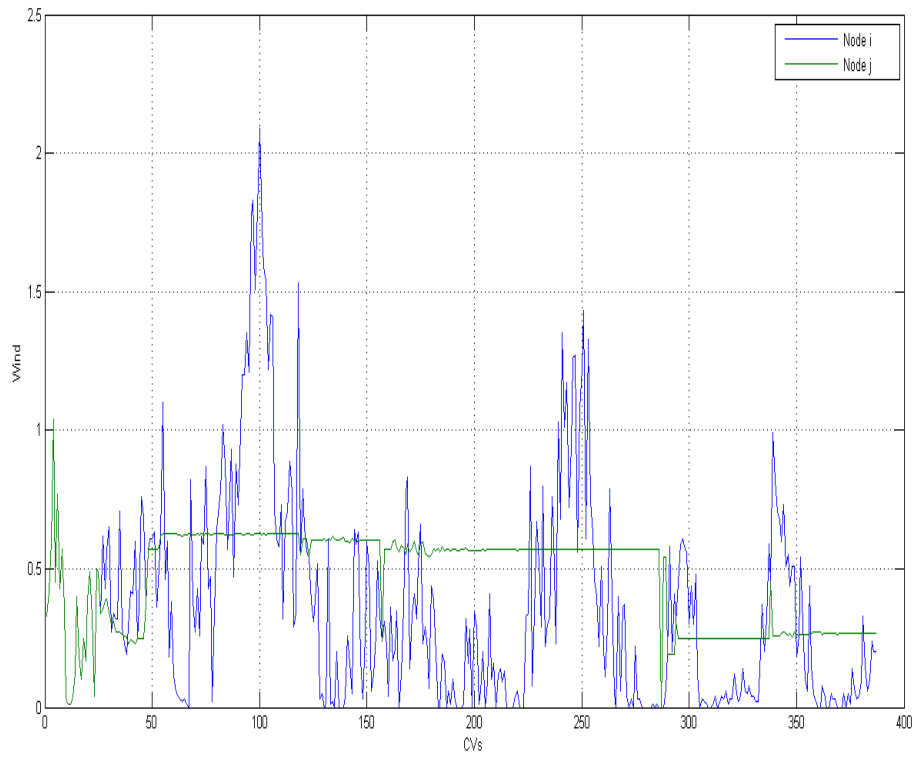
Και σε αυτήν την περίπτωση όμως υπάρχει κάποιο τίμημα με το οποίο επιβαρυνόμαστε. Ένα αυστηρό όριο έχει σαν αποτέλεσμα, να αυξηθούν οι φορές κατά τις οποίες η συνθήκη απόφασης στον κόμβο i , δηλ. $e_{i-i}(t) < Max_{error}$, δεν ικανοποιείται, αυτό έχει σαν επακόλουθο την αποστολή ολοκλήρου του διανύσματος πλαισίου, συνεπώς αύξηση στο ενεργειακό κόστος του μηχανισμού, άρα μικρότερη διάρκεια ζωής του κόμβου.



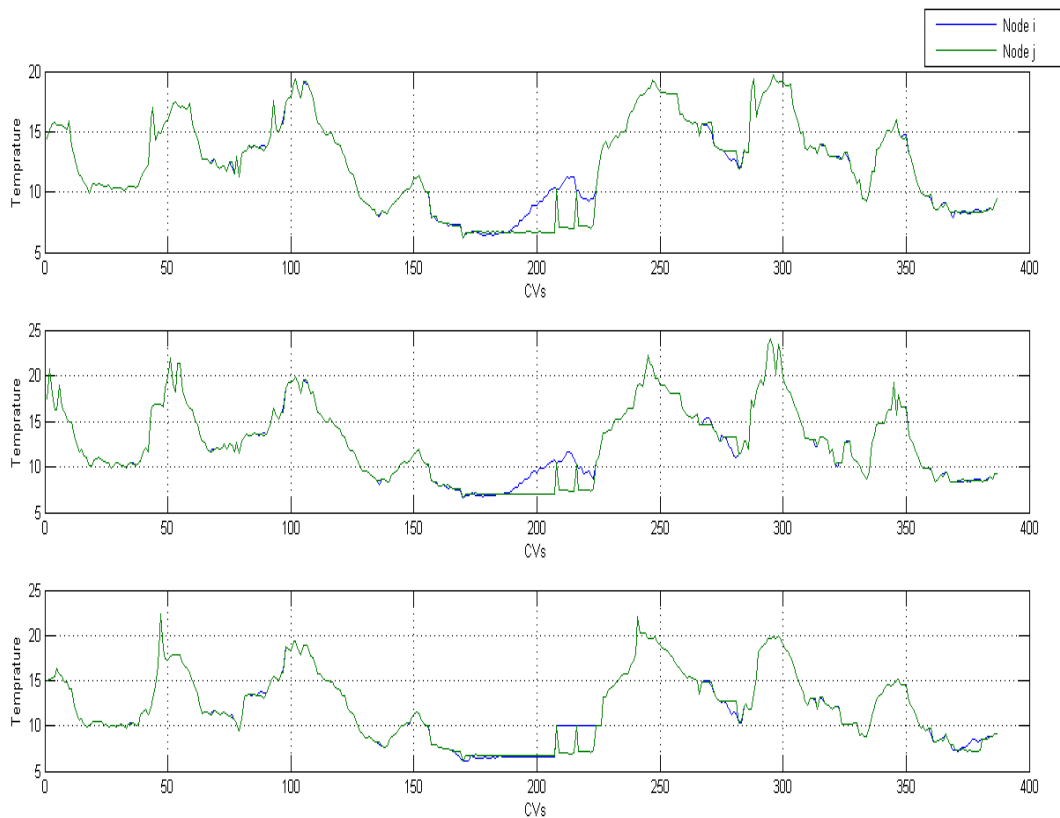
(α)



(β)



(γ)



(δ)

Σχήμα 4: οι τιμές της (α) θερμοκρασίας, (β) υγρασίας και (γ) ταχύτητας ανέμου στα πραγματικά δεδομένα πλαισίου που έχει αποθηκεύσει ο κόμβος i (μπλε) σε σχέση με αυτά που αποθηκεύει ο κόμβος j (πράσινο) για $Max_{error} = 10\%$ (δ) οι τιμές της θερμοκρασίας στα πραγματικά δεδομένα πλαισίου που έχει αποθηκεύσει ο κόμβος i (μπλε) σε σχέση με αυτά που αποθηκεύει ο κόμβος j (πράσινο) για $Max_{error} = 1\%$

Στο σχήμα 4(δ), παρατηρούμε ξεκάθαρα την εξάρτηση αυτή που αναφέραμε, δηλ. μεταξύ Max_{error} , που έχει τεθεί στον κόμβο i , και της εικόνας που έχουν οι δυο γειτονικοί κόμβοι i και j για τα δεδομένα πλαισίου, τα οποία προωθούνται μέσα στο ασύρματο δίκτυο αισθητήρων. Βλέπουμε την εικόνα που έχουν οι δυο κόμβοι για τα δεδομένα πλαισίου στις τιμές της θερμοκρασίας (το ίδιο παρατηρείται και για την υγρασία και την ταχύτητα του ανέμου) όταν θέσουμε μια πολύ αυστηρή τιμή για το Max_{error} , ίση με 1%. Τα δυο σήματα σχεδόν ταυτίζονται, κάτι που σημαίνει ότι υπάρχει πολύ μεγάλη ακρίβεια δεδομένων, κάτι που προκύπτει και από το μέσο σφάλμα ανακατασκευής που σε αυτή την περίπτωση είναι μόλις 0.3%.

Εκτός του συνολικού μέσου σχετικού σφάλματος ανακατασκευής (όπως δηλ. έχει διαμορφωθεί στο τέλος της διαδικασίας), αξίζει να μελετήσουμε και την εξέλιξη αυτού

σε κάθε χρονική στιγμή t , όπως προκύπτει από τον τύπο $\hat{e}(t) = \frac{\sum_1^t e_{i-j}(t)}{t}$. Στην εικόνα 6(δ), απεικονίζεται η εξέλιξη του μέσου σχετικού σφάλματος ανακατασκευής,

καθόλη την διάρκεια προώθησης δεδομένων πλαισίου. Παρατηρούμε, πως καθώς περισσότερα δεδομένα πλαισίου προωθούνται, υπάρχει σε γενικές γραμμές μια αύξουσα πορεία της συγκεκριμένης τιμής.

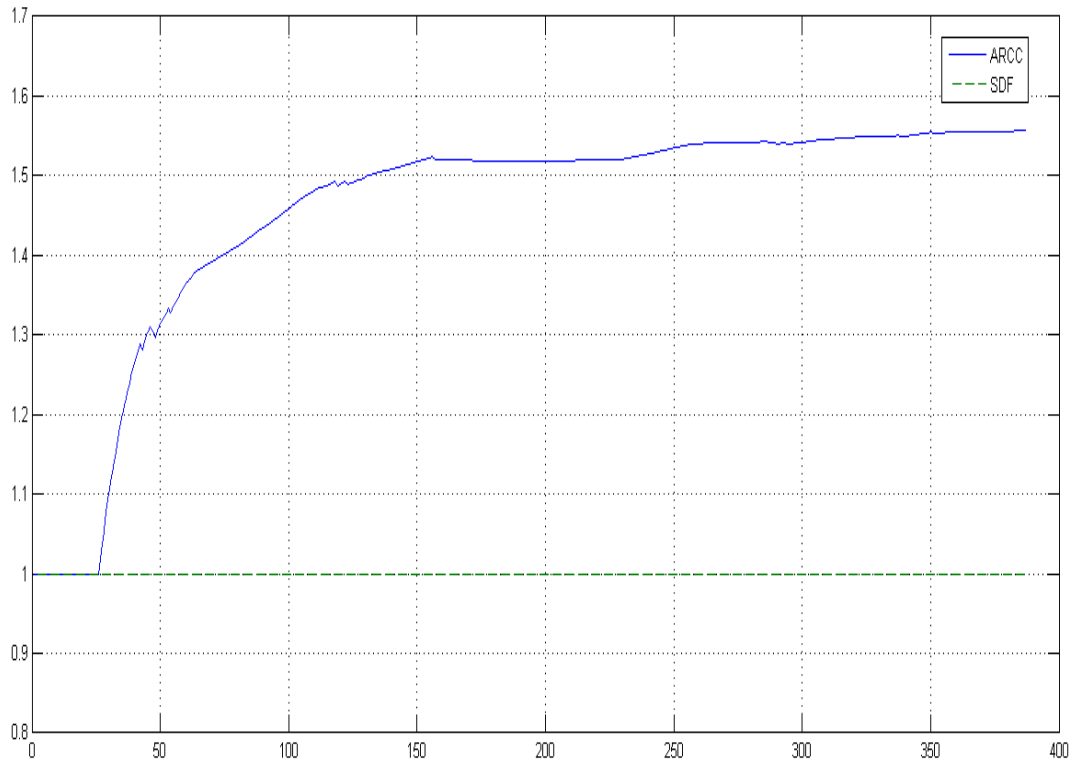
Στόχος, λοιπόν, του προτεινόμενου μοντέλου, όπως έχει αναφερθεί ήδη είναι η ενεργειακή αποδοτικότητα, κατά την προώθηση των δεδομένων πλαισίου, σε ένα ασύρματο δίκτυο αισθητήρων. Αυτό επιτυγχάνεται μέσω της ελαχιστοποίησης του ενεργειακού κόστους που απαιτείται για την προώθηση των δεδομένων, μέσω της συμπίεσης αυτών των δεδομένων πλαισίου. Σε αντάλλαγμα αυτής της συμπίεσης, που ελαττώνει κατά πολύ την ενέργεια που καταναλώνεται από τους κόμβους, υπάρχει μια επιπρόσθετη «σπατάλη» ενέργειας λόγω των εντολών που θα πρέπει να εκτελεστούν για την υλοποίηση του μηχανισμού VAR. Όλα αυτά τα κόστη συμπεριλαμβάνονται μέσα στην (1), συνεπώς έχουμε μια σαφή εικόνα για το τι κερδίζουμε και τι χάνουμε στο θέμα ενέργειας. Για να έχουμε όμως μια ξεκάθαρη εκτίμηση για τον προτεινόμενο μηχανισμό, θα πρέπει να λάβουμε υπόψη μας εκτός του παράγοντα ενέργεια και τον παράγοντα ακρίβεια των δεδομένων, δηλ. το σφάλμα ανακατασκευής των πλαισίων, στο οποίο μόλις αναφερθήκαμε. Γι αυτό το λόγο θα ορίσουμε μια μετρική η οποία περιλαμβάνει και τις δυο προαναφερθείσες παραμέτρους, με αποτέλεσμα να έχουμε μια ολοκληρωμένη και σφαιρική προσέγγιση στο θέμα της εκτίμησης του προτεινόμενου μοντέλου,

$$w(t) = pcg_{arcc-sdf}(t) + \frac{1}{1 + e_{i-j}(t)}$$

Αυτό που επιδιώκουμε είναι η τιμή του $w(t)$ να είναι όσο το δυνατό εγγύτερα στο 2 και μεγαλύτερη από την μονάδα, $w(t) \in [1, 2]$, αυτό προκύπτει από την στιγμή που επιθυμούμε να έχουμε όσο το δυνατό μεγαλύτερο ποσοστιαίο όφελος, δηλ $pcg_{arcc-sdf}(t) \rightarrow 1$ και μηδενικό σφάλμα ανακατασκευής $e_{i-j}(t) \rightarrow 0$.

Όπως προκύπτει από τον παραπάνω τύπο, με την απλή προώθηση όλων των δεδομένων, παίρνουμε σαν συντελεστή $w(t) = 1$.

Στο σχήμα 5, απεικονίζεται η εξέλιξη του συντελεστή απόδοσης $w(t)$, καθώς τα διανυσματικά πλαίσια διέρχονται από τους κόμβους, για το προτεινόμενο μοντέλο και για την απλή προώθηση δεδομένων.



Σχήμα 5: Συντελεστής απόδοσης W , του μηχανισμού ARCC και SDF

Στο προτεινόμενο μηχανισμό ARCC, η μέση τιμή του συντελεστή απόδοσης ισούται με $\tilde{w}=1.457$, παρουσιάζοντας τελική τιμή, αφού έχει ολοκληρωθεί η προώθηση όλων των δεδομένων, ίση με $w=1.557$.

Μια άλλη μετρική (συντελεστή απόδοσης), με την οποία συνεκτιμούμε ταυτόχρονα τους παράγοντες κόστος και ενέργεια, για τους οποίους ενδιαφερόμαστε πρωτίστως είναι αυτή που προκύπτει από τον παρακάτω τύπο και ισοζυγίζει τους δυο αυτούς παράγοντες,

$$f(t) = 0.5 * pcg(t) + 0.5 * (1 - e_{i-j}(t))$$

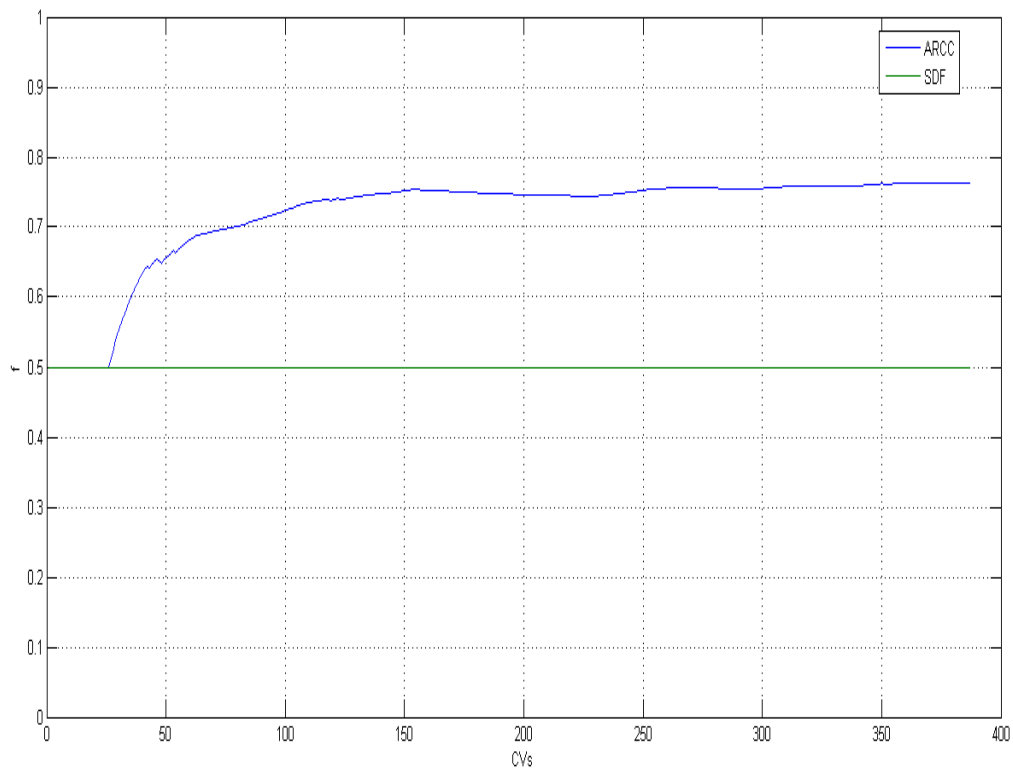
Όπως είναι φυσικό, ανάλογα με το ύψος της κάθε εφαρμογής επίγνωσης πλαισίου, που χρησιμοποιεί κάποιο ασύρματο δίκτυο αισθητήρων προκειμένου να απόκτη γνώση του περιβάλλοντος της, και να προσαρμόζεται αντίστοιχα, η βαρύτητα που θα δοθεί σε κάθε έναν από τους παράγοντες μπορεί να είναι διαφορετική.

Οι τιμές που μπορεί να πάρει η μετρική f , κυμαίνονται στο διάστημα $[0,1]$.

Η ιδανική περίπτωση για τον μηχανισμό μας, αλλά και για κάθε μηχανισμό που αποσκοπεί στο μέγιστο ενεργειακό κέρδος ($pcg \rightarrow 1$) και στο όσο το δυνατό μικρότερο σφάλμα ($e_{i-j}(t) \rightarrow 0$) είναι να προσεγγίζει το άνω όριο αυτού του διαστήματος.

Σε αυτήν την περίπτωση η απλή προώθηση δεδομένων μας δίνει $f(t) = 0.5$, σταθερό σε όλη την χρονική διάρκεια προώθησης των δεδομένων.

Στην σχήμα 6, απεικονίζεται η εξέλιξη του συντελεστή απόδοσης f , καθώς τα διανυσματικά πλαίσια διέρχονται από τους κόμβους, για το προτεινόμενο μοντέλο ARCC και για την απλή προώθηση δεδομένων.



Σχήμα 6: Συντελεστής απόδοσης f , των μηχανισμών ARCC και SDF

Στο προτεινόμενο μηχανισμό ARCC, η μέση τιμή του συντελεστή απόδοσης f ισούται με $\tilde{f} = 0.719$, παρουσιάζοντας τελική τιμή, αφού έχει ολοκληρωθεί η προώθηση όλων των δεδομένων, ίση με $f = 0.762$.

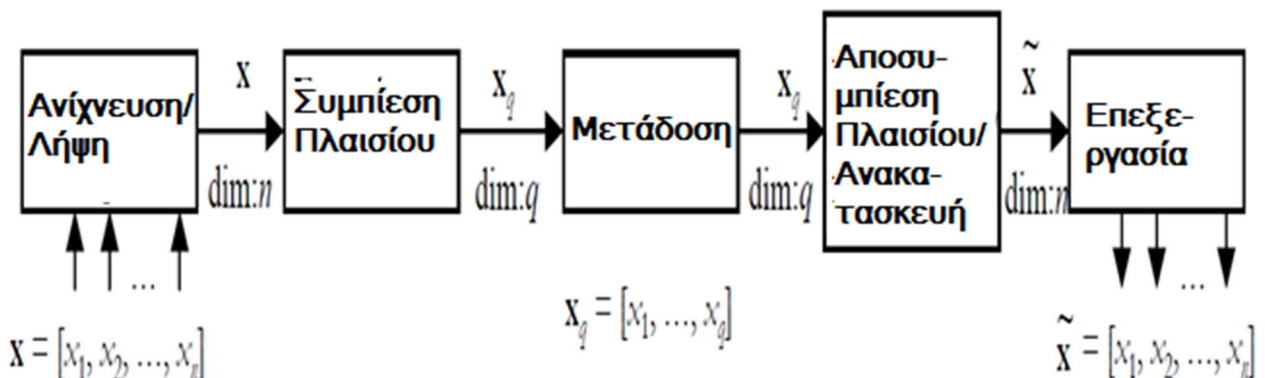
3 ΣΥΜΠΙΕΣΗ ΠΛΑΙΣΙΟΥ ΒΑΣΙΣΜΕΝΗ ΣΤΗΝ ΑΝΑΛΥΣΗ ΚΥΡΙΩΝ ΣΥΝΙΣΤΩΣΩΝ (PRINCIPAL COMPONENTS-BASED CONTEXT COMPRESSION – PC3)

Στο εδάφιο αυτό, θα κάνουμε μια συνοπτική παρουσίαση των βασικών λειτουργιών ενός ήδη υπάρχοντος μηχανισμού, του PC3 [7]. Στόχος του μηχανισμού αυτού είναι, όπως και του ARCC που αναπτύχθηκε παραπάνω, η αποδοτική προώθηση της πληροφορίας πλαισίου, ανάμεσα στους κόμβους, ενός ασύρματου δικτύου αισθητήρων. Με σκοπό την ελαχιστοποίηση του ενεργειακού κόστους και άρα την μεγιστοποίηση της διάρκειας ζωής του κόμβου.

Στην συνέχεια θα έχουμε την δυνατότητα να συγκρίνουμε τον συγκεκριμένο μηχανισμό, με αυτόν που παρουσιάσαμε πρωτύτερα, και σε επόμενο εδάφιο θα συνδυάσουμε αυτούς τους 2 μηχανισμούς σε ένα νέο υβριδικό μοντέλο που θα παρουσιάσουμε.

3.1 Τρόπος λειτουργίας του PC3

Ο μηχανισμός PC3, εκμεταλλεύεται, και αυτός, την διανυσματική φύση των ανταλλασσόμενων πληροφοριών πλαισίου αλλά και την σχέση εξάρτησης ανάμεσα στα διαδοχικά πλαίσια. Η λειτουργία της συμπίεσης στο συγκεκριμένο πρότυπο επιτυγχάνεται μέσω της προβολής (πριν την μετάδοση) των παραγόμενων διανυσμάτων πλαισίου σε έναν χώρο μικρότερης διάστασης. Πυρήνας του συγκεκριμένου μηχανισμού είναι η Ανάλυση Κυρίων Συνιστωσών.



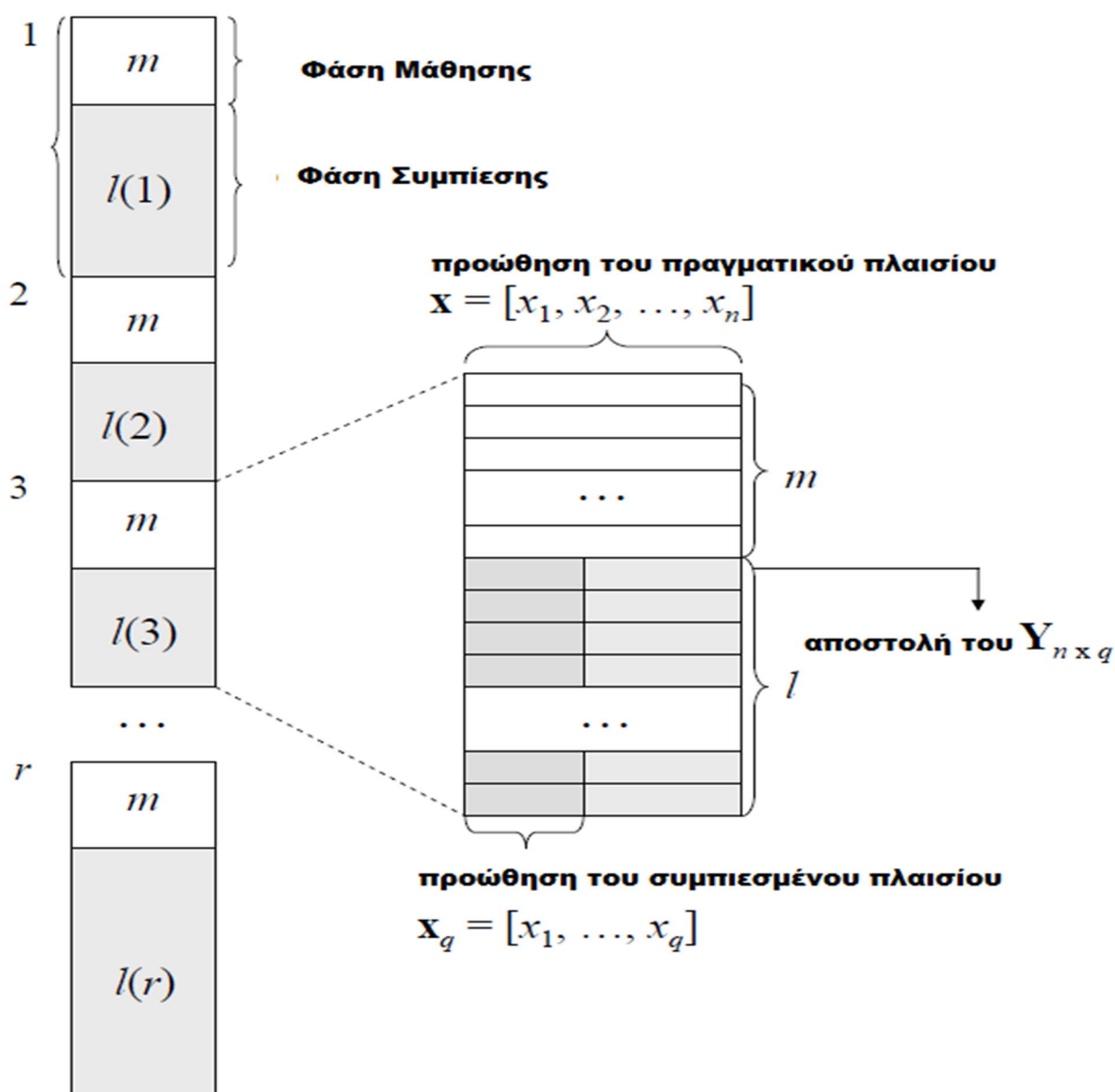
Εικόνα 3: Μηχανισμός Συμπίεσης πλαισίου PC3

Στην εικόνα 3, απεικονίζεται η βασική ιδέα για τον μηχανισμό συμπίεσης πλαισίου. Ένας κόμβος ανιχνεύει/λαμβάνει ένα n -διάστατο διάνυσμα πλαισίου (x), ο κόμβος συμπιέζει αυτό το διάνυσμα σε ένα αντίστοιχο διάστασης q (x_q), όπου $q < n$ και στη συνέχεια το προωθεί στον επόμενο κόμβο. Ο κόμβος-παραλήπτης αναπαράγει με την αντίστροφη διαδικασία ένα n -διάστατο διάνυσμα πλαισίου (\tilde{x}), για περαιτέρω επεξεργασία (ή προώθηση σε επόμενους κόμβους).

Έστω $x_i(t) = [x_{ij}]_{j=1, \dots, n}$, το διάνυσμα πλαισίου (CV), αποτελούμενο από n μετρήσεις x_{ij} , οι οποίες έχουν συλλεχθεί από έναν κόμβο ανίχνευσης i , την χρονική στιγμή t . Η x_{ij} αντιστοιχεί στην τιμή της j -οστής συνιστώσας του πλαισίου (π.χ.

θερμοκρασία, ταχύτητα ανέμου) , η οποία ανιχνεύεται (ή λαμβάνεται) από τον κόμβο i . Σε κάθε χρονική στιγμή t ο κάθε κόμβος λαμβάνει (ή ανιχνεύει) ένα πλαίσιο $x(t) \in \mathbb{R}^n$. Ο κάθε κόμβος συγκεντρώνει τα τελευταία $m > 0$ ληφθέντα πλαίσια, $x(t-m), x(t-m+1), \dots, x(t)$, και κατασκευάζει έναν πίνακα X διαστάσεων $m \times n$. Αυτός ο πίνακας X αποτελεί βάση για την μετέπειτα διαδικασία ανάλυσης κυρίων συνιστωσών. Από αυτόν τον πίνακα και βασιζόμενη στην διαδικασία ανάλυσης κυρίων συνιστωσών, για την οποία δεν θα αναφερθούμε σε λεπτομέρειες, προκύπτει ένας νέος πίνακας Y διαστάσεων $n \times q$, από τον οποίο, μέσω της σχέσης $x_q(t) = Y^T x(t)$, παίρνουμε το συμπιεσμένο πλαίσιο. Με την αντίστροφη διαδικασία ($\tilde{x}(t) = Yx_q(t)$) ανακτούμε το πραγματικό από το συμπιεσμένο

Το μοντέλο PC3 αποτελείται από δυο φάσεις λειτουργίας: μάθησης και συμπίεσης, όπως φαίνεται και στην εικόνα 4.



Εικόνα 4: Φάσεις Λειτουργίας του μηχανισμού PC3

Κατά την διάρκεια της φάσης μάθησης $1 \leq t \leq m$, ο κόμβος i αποστέλλει στον επόμενο κόμβο το πραγματικό διάνυσμα πλαισίου $x(t)$ και παράλληλα διατηρεί την κάθε μέτρηση στην μονάδα αποθήκευσης του, ούτως ώστε στο τέλος της φάσης αυτή να έχει συγκεντρώσει m μετρήσεις. Όταν συμπληρωθούν m μετρήσεις, μπορεί πλέον να κατασκευαστεί ο πίνακας Y .

Στην φάση συμπίεσης, $m \leq t \leq m+l$, ο «εκπαιδευμένος» πλέον κόμβος i αποστέλλει στον γειτονικό κόμβο j τα συμπιεσμένα διανύσματα πλαισίου και ο τελευταίος με την σειρά του ανακατασκευάζει τα πραγματικά διανύσματα πλαισίου, βασιζόμενο επίσης στον πίνακα Y .

Από την στιγμή που ο κόμβος j υπολογίζει κατά προσέγγιση το διάνυσμα πλαισίου, κατά την φάση της συμπίεσης, γίνεται αντιληπτό, πως προκαλείται και το αντίστοιχο

σχετικό σφάλμα ανακατασκευής, το οποίο ισούται με $e_{i-j}(t) = \frac{\|\tilde{x}(t) - x(t)\|}{\|x(t)\|}$,

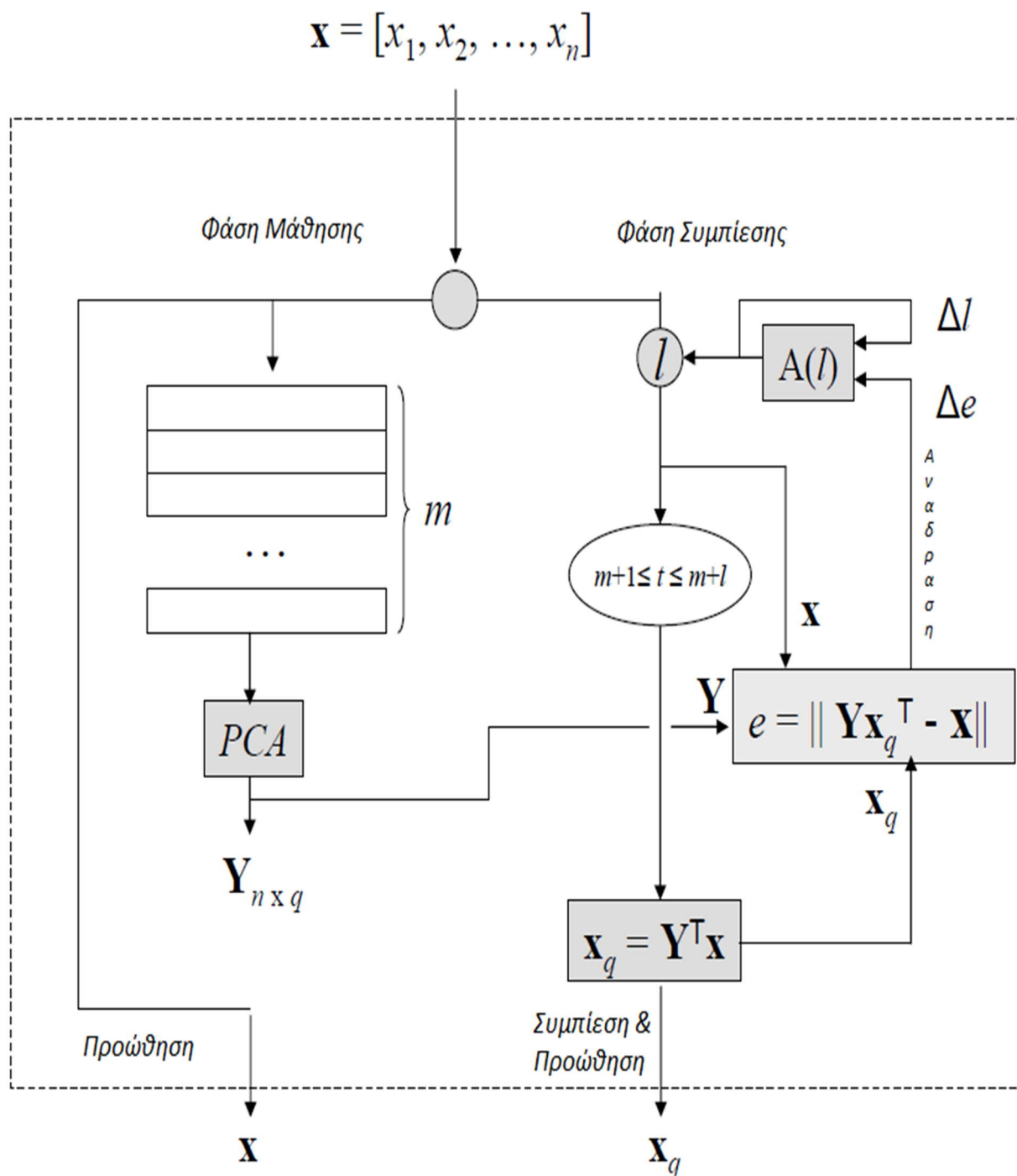
για την χρονική στιγμή t .

Στο μηχανισμό PC3, λοιπόν, επιτυγχάνεται μείωση του ενεργειακού κόστους που απαιτείται για την διαδικασία της προώθησης, αφού στις φάσεις συμπίεσης δεν προωθείται το πραγματικό διάνυσμα πλαισίου, αλλά ένα άλλο πολύ μικρότερης διάστασης, άρα λιγότερες τιμές, συνεπώς υπάρχει μικρότερο κόστος για αποστολή και παραλαβή των ανταλλασσόμενων πληροφοριών. Αυτό γίνεται, εις βάρος της ακρίβειας των δεδομένων, αφού εισάγεται το σφάλμα ανακατασκευής, όπως ακριβώς συμβαίνει και στην ARCC.

Για να έχουμε όσο το δυνατό μεγαλύτερη συμπίεση στα δεδομένα πλαισίου, επιδιώκουμε να έχουμε (i) μια τιμή q πολύ μικρότερη από την n και (ii) το διάστημα συμπίεσης l πολύ μεγαλύτερο από το διάστημα μάθησης m . Είναι ξεκάθαρο, πως σε μία τέτοια κατάσταση των παραπάνω μεταβλητών έχουμε μικρότερο κόστος για αποστολή και λήψη των δεδομένων (αφού προωθούνται λιγότερα στοιχεία), εις βάρος της ακρίβειας των δεδομένων.

Το πλήρες μοντέλο PC3, περιλαμβάνει και έναν μηχανισμό προσαρμογής, $A(l)$ του διαστήματος συμπίεσης l με σκοπό να επιτυγχάνεται η καλύτερη δυνατή ισορροπία ανάμεσα στο κόστος και το σφάλμα. Ο μηχανισμός δεν θα μελετηθεί εκτενώς στην παρούσα εργασία,

Στη παρακάτω εικόνα 5, απεικονίζονται, όλες οι λειτουργίες που εκτελούνται, στο κόμβο i , στα πλαίσια του μηχανισμού PC3.



Εικόνα 5: Διαδικασίες που εκτελούνται μέσα στο κόμβο πριν προώθηση το πλαίσιο που έλαβε/ανίχνευσε

3.2 Σύγκριση Απόδοσης ARCC – PC3

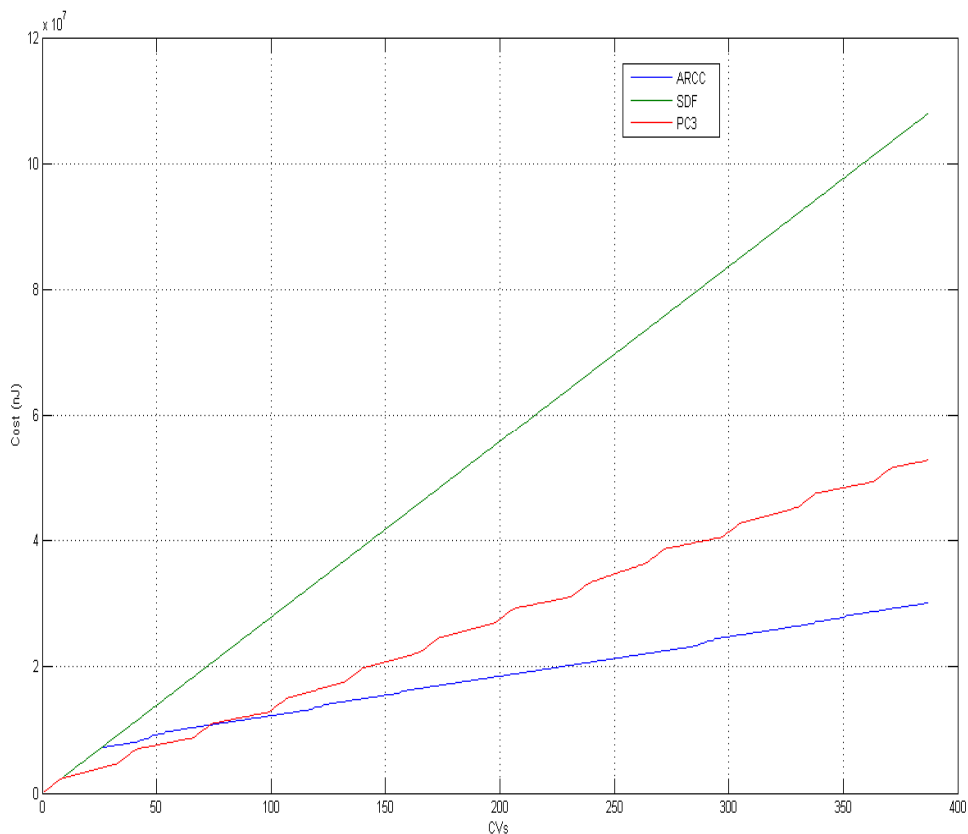
Πλέον, αφού παρουσιάσαμε τα βασικά στοιχεία του μηχανισμού PC3, θα προχωρήσουμε σε μια σύγκριση όσον αφορά την απόδοση που επιτυγχάνουν οι μηχανισμοί ARCC και PC3.

Τα δεδομένα που θα χρησιμοποιηθούν στην πειραματική διαδικασία είναι τα ίδια που χρησιμοποιήθηκαν και προηγουμένως, δηλ 387 διανύσματα πλαισίου, που έχουν ανιχνευτεί σε πραγματικές συνθήκες. Κάθε πλαίσιο αποτελείται από ένα διάνυσμα 7

μεταβλητών με περιεχόμενο (θερμοκρασία1, υγρασία1, θερμοκρασια2, υγρασία2, θερμοκρασια3, υγρασία3, ταχύτητα ανέμου), τα οποία έχουν δειγματοληπτηθεί με συχνότητα 10^{-2} Hz. Χρησιμοποιείται και πάλι το μοντέλο κατανάλωσης ενέργειας Mica2.

Όπως αναφέραμε και παραπάνω, το PC3 κάνει χρήση ενός μηχανισμού προσαρμογής, για να ρυθμίζει την τιμή του διαστήματος συμπίεσης σε κάθε περίοδο, στο πλαίσιο της πειραματικής διαδικασίας δεν έγινε χρήση αυτού του μηχανισμού και επιλέχθηκαν σταθερά διαστήματα μάθησης ($m = 8$) και συμπίεσης ($l = 25$), κατά τέτοιο τρόπο που να προσεγγίζονται οι μέσες τιμές του μήκους αυτών των διαστημάτων, στην περίπτωση που είχε χρησιμοποιηθεί ο ανάλογος μηχανισμός. Η υλοποίηση έχει γίνει στο Matlab (Παράρτημα III)

Το πρώτο στοιχείο που θα μελετήσουμε, αποτελεί και τον πρωτεύοντα στόχο των δύο μηχανισμών, που δεν είναι άλλος από το ενεργειακό κόστος που δαπανάται κατά την εφαρμογή τους. Στο σχήμα 7, βλέπουμε την εξέλιξη του κόστους καθόλη την διάρκεια, προώθησης των πλαισίων.



Σχήμα 7: Το ενεργειακό κόστος που καταναλίσκεται σε κάθε μηχανισμό

Είναι φανερό πως ο μηχανισμός ARCC είναι λιγότερο κοστοβόρος από τον αντίστοιχο του PC3. Για την ακρίβεια, στον ARCC καταναλίσκεται 42% λιγότερη ενέργεια, για την προώθηση των πληροφοριών πλαισίου, ανάμεσα σε δύο κόμβους, σε ένα ασύρματο δίκτυο αισθητήρων. Αυτή η σημαντική διαφορά, θα μπορούσαμε να πούμε πως είναι αναμενόμενη, καθώς η ARCC υπό φυσιολογικές συνθήκες, εξαιρουμένου ενός μικρού τμήματος στην αρχή, όπου είναι απαραίτητο για την κατασκευή της μνήμης της διαδικασίας VAR, εφαρμόζει την διαδικασία της συμπίεσης σχεδόν σε όλα τα επόμενα

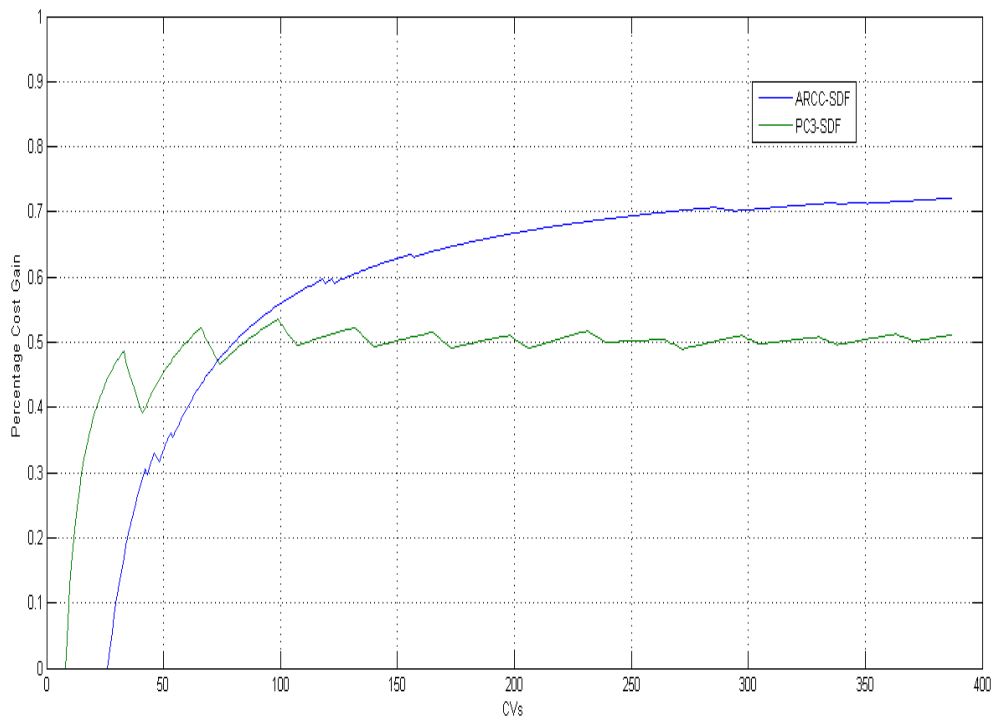
(εξαρτάται από το κατώφλι σφάλματος που έχουμε θέσει) δείγματα που διέρχονται από τον κόμβο i .

Υπενθυμίζουμε πως σύμφωνα με την πειραματική διαδικασία στο ARCC το 90% των πραγματικών δεδομένων δεν αποστέλλονται, αλλά αντικαθιστούνται από έναν σήμα ανακατασκευής. Αντιθέτως, στον PC3, αποστέλλονται λιγότερα δεδομένα από τα πραγματικά, μόνο κατά την αντίστοιχη φάση συμπίεσης, η οποία αποτελεί το 68% της συνολικής λειτουργίας, ενώ στην διαδικασία μάθησης η προώθηση γίνεται, όπως και στην απλούστατη διαδικασία SDF, δηλαδή προώθηση αυτούσιων των πληροφοριών πλαισίου. Είναι ξεκάθαρο μάλιστα στο σχήμα 7, πως υπάρχουν διαστήματα στα οποία ο ρυθμός αύξησης της καμπύλης κόστους των μηχανισμών PC3 και SDF είναι ίδιος. Αυτά είναι τα διαστήματα μάθησης που αναφέραμε.

Επιπροσθέτως, στον μηχανισμό ARCC η συμπίεση που εφαρμόζεται είναι μεγαλύτερη σε ένταση. Υπενθυμίζουμε πως σύμφωνα με τον τρόπο λειτουργίας του μηχανισμού, το διάνυσμα πλαισίου αντικαθίσταται κατά την συμπίεση με ένα σήμα μεγέθους 1 bit. Για τα δεδομένα του πειράματός μας δηλ. $1bit$ αντικαθιστά $27 + 15 = 32bytes \times 8bits = 224bits$ δηλ. συμπίεση 99,55% κάθε φορά που αυτή πραγματοποιείται. Ενώ στην ανάλογη διαδικασία συμπίεσης του PC3, στην καλύτερη των περιπτώσεων, όταν δηλαδή η ανάλυση μας δίνει μόνο μια κύρια συνιστώσα, τότε $4bytes \times 8bits = 32bits$ αντικαθιστούν $7 \times 4bytes \times 8bits = 224bits$, προκύπτει λοιπόν συμπίεση 85%, Στην πραγματικότητα η ανάλυση δίνει συνήθως περισσότερες της μία συνιστώσας, στο πείραμα μας η μέση τιμή των κυρίων συνιστωσών, που έδινε η ανάλυση σε κάθε περίοδο, ήταν 1,58 την φορά, με διακύμανση στο διάστημα [1,3].

Συνεπώς, η ARCC, πετυχαίνει μεγαλύτερη συμπίεση και σε εύρος και σε βάθος.

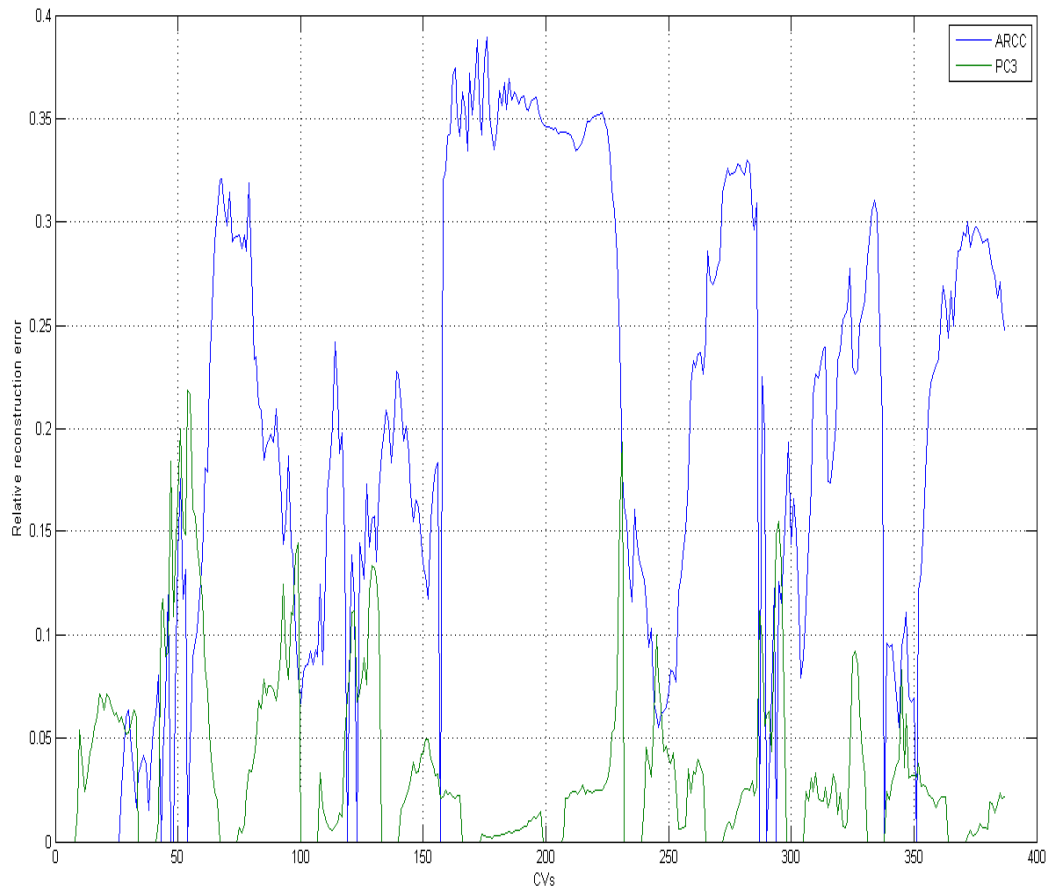
Τα παραπάνω αποτελέσματα, διακρίνονται και στο σχήμα 8, όπου συγκρίνονται οι δυο διαδικασίες, όσον αφορά το ποσοστιαίο όφελος που πετυχαίνουν, σε σχέση με τον μηχανισμό απλής προώθησης όλων των πλαισίων.



Σχήμα 8: Ποσοστιαίο ενεργειακό κέρδος των ARCC (μπλε) και PC3 (πράσινο) σε σχέση με την SDF

Όπως έχουμε αναφέρει, επανειλημμένως, το τίμημα που πληρώνουν αυτοί οι δυο μηχανισμοί, προκειμένου να επιτύχουν την προσδοκώμενη συμπύεση των ανταλλασσόμενων πληροφοριών, και συνεπώς την μείωση του ενεργειακού κόστους που καταναλίσκεται, είναι το προκύπτον σφάλμα ανακατασκευής, δηλαδή η μείωση στην ακρίβεια των δεδομένων. Είναι σημαντικό λοιπόν να εξετάσουμε κατά πόσο επηρεάζεται αυτή η ακρίβεια, κατά την εφαρμογή του κάθε μηχανισμού.

Στο παρακάτω σχήμα 9, απεικονίζεται το σχετικό σφάλμα ανακατασκευής των διανυσμάτων πλαισίου στο κόμβο j , σε σχέση με την πραγματική τιμή που έχουν αυτά, δηλαδή την τιμή που έχει ο κόμβος i .



Σχήμα 9: Το σχετικό σφάλμα ανακατασκευής σε κάθε διάνυσμα πλαισίου στους μηχανισμούς ARCC (μπλε) και PC3 (πράσινο)

Φαίνεται λοιπόν, με μια πρώτη μάτια από το σχεδιάγραμμα, πως ο μηχανισμός ARCC, επιφέρει ένα κατά πολύ μεγαλύτερο σφάλμα ανακατασκευής σε σχέση με το PC3.

Πιο συγκεκριμένα το μέσο σχετικό σφάλμα του μηχανισμού ARCC, είναι 19,68%, ενώ το αντίστοιχο σφάλμα στον PC3, μόλις 3,5%. Είναι φανερό πως αυτή μεγάλη διαφορά οφείλεται, στην επίσης μεγάλη διαφορά, από την αντίθετη μεριά όμως, που επιφέρει ο κάθε μηχανισμός στην συμπύεση των δεδομένων πλαισίου.

Ο ARCC, συμπιέζει κατά πολύ περισσότερο τα δεδομένα, όπως είδαμε λίγο πριν, συνεπώς προκαλείται και μεγαλύτερο σφάλμα ανακατασκευής αυτών, άρα χαμηλότερο επίπεδο στην ακρίβεια των δεδομένων, σε σχέση με ότι συμβαίνει στη PC3.

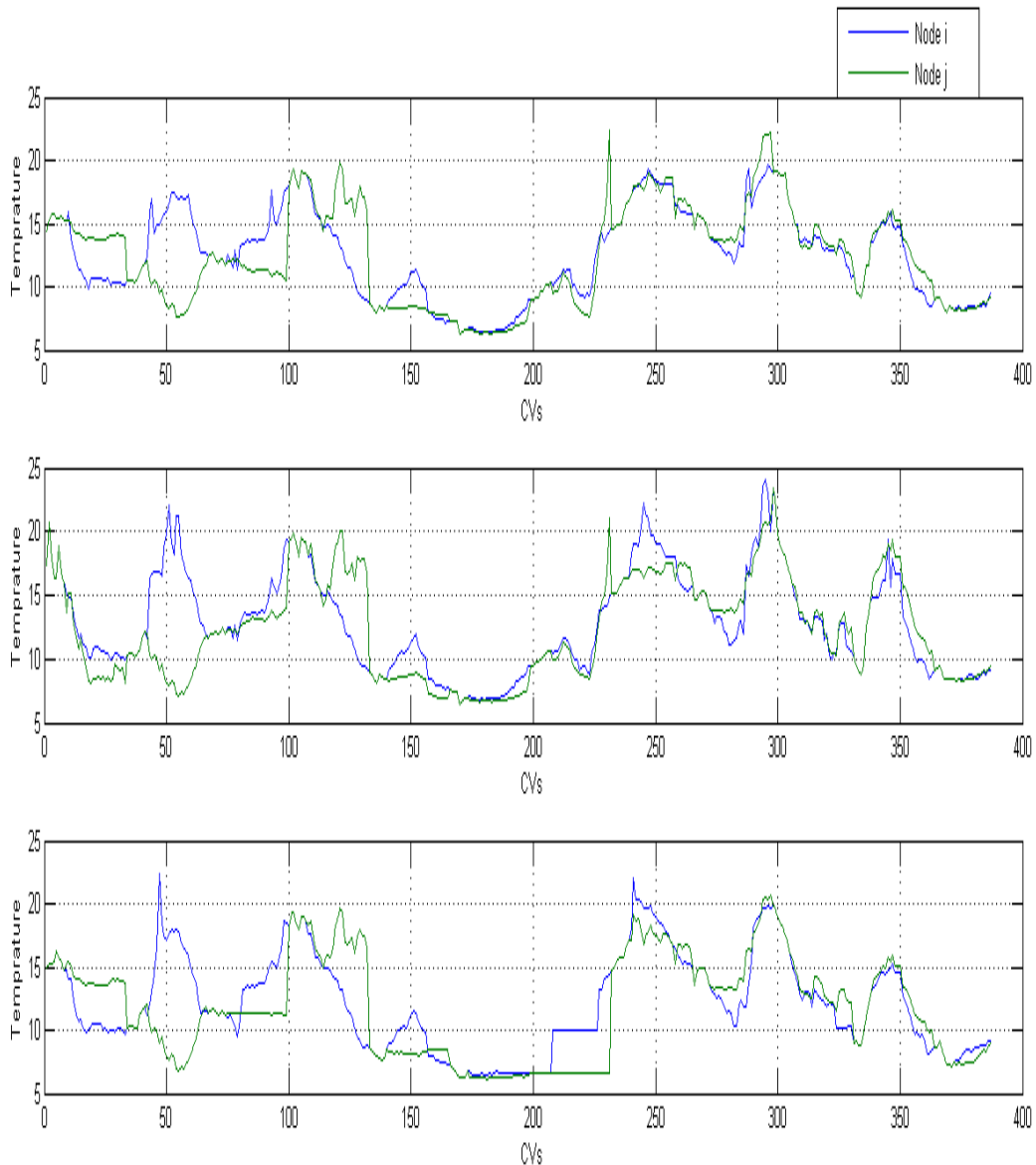
Από την διακύμανση του σφάλματος, κατά την διάρκεια της προώθησης των πλαισίων, διακρίνουμε πως στο PC3, περιοδικά εμφανίζονται διαστήματα με μηδενικό σφάλμα, όπως γίνεται αντιληπτό, αυτά τα διαστήματα είναι οι φάσεις μάθησης (οι οποίες θυμίζουμε πως αποτελούν το 32% ολόκληρης της διαδικασίας), κατά τις οποίες ο ένας κόμβος προωθεί αυτούσιο το διάνυσμα πλαισίου στον επόμενο, συνεπώς δεν τίθεται θέμα ανακατασκευής αυτού και άρα σφάλματος.

Από την άλλη, στον ARCC, εάν εξαιρέσουμε, ένα μικρό διάστημα στην αρχή της διαδικασίας προώθησης, το οποίο χρησιμοποιείται για την δημιουργία μνήμης της διαδικασίας VAR, είναι ελάχιστες οι φορές (συγκεκριμένα μόλις 12 από τα 361

πλαίσια($\approx 3\%$) που προωθείται ολόκληρο το διάνυσμα πλαισίου από τον έναν κόμβο στον άλλο, και συνεπώς δεν υπάρχει σφάλμα ανακατασκευής.

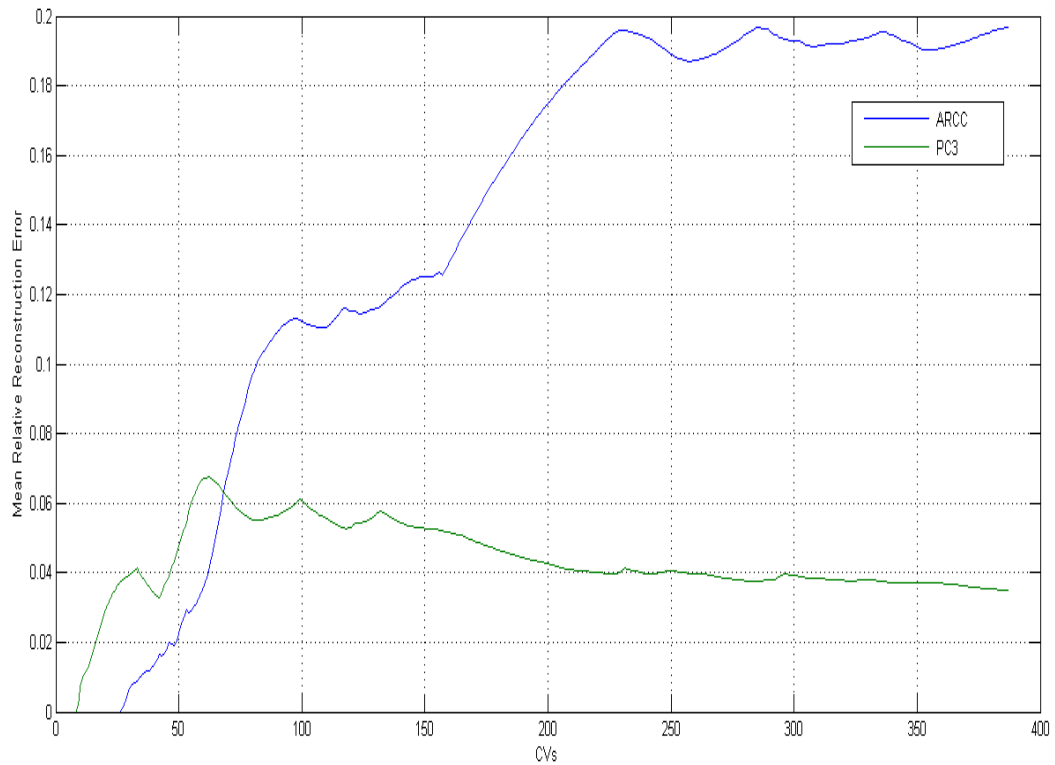
Η μεγαλύτερη ακρίβεια που υπάρχει στα δεδομένα που έχει στην κατοχή του ο κάθε κόμβος μετά την προώθηση όλων, αντικατοπτρίζεται και από το σχήμα 10, στο οποίο έχουμε μια οπτική αντίληψη των σημάτων (απεικονίζονται οι τιμές της θερμοκρασίας μονό, παρόμοιες παρατηρήσεις ισχύουν και για τις υπόλοιπες μεταβλητές) που κατέχει ο κάθε κόμβος.

Μια σύγκριση με το αντίστοιχο σχεδιάγραμμα 4(α), που απεικονίζει την αντίστοιχη ακρίβεια των σημάτων σε κάθε κόμβο στον μηχανισμό ARCC, αρκεί για να κατανοήσουμε την μεγάλη διαφορά που υπάρχει στο σφάλμα ανακατασκευής της κάθε διαδικασίας.



Σχήμα 10: οι τιμές της θερμοκρασίας στα πραγματικά δεδομένα πλαισίου που έχει αποθηκεύσει ο κόμβος (μπλε) σε σχέση με αυτά που αποθηκεύει ο κόμβος (πράσινο) στον μηχανισμό PC3

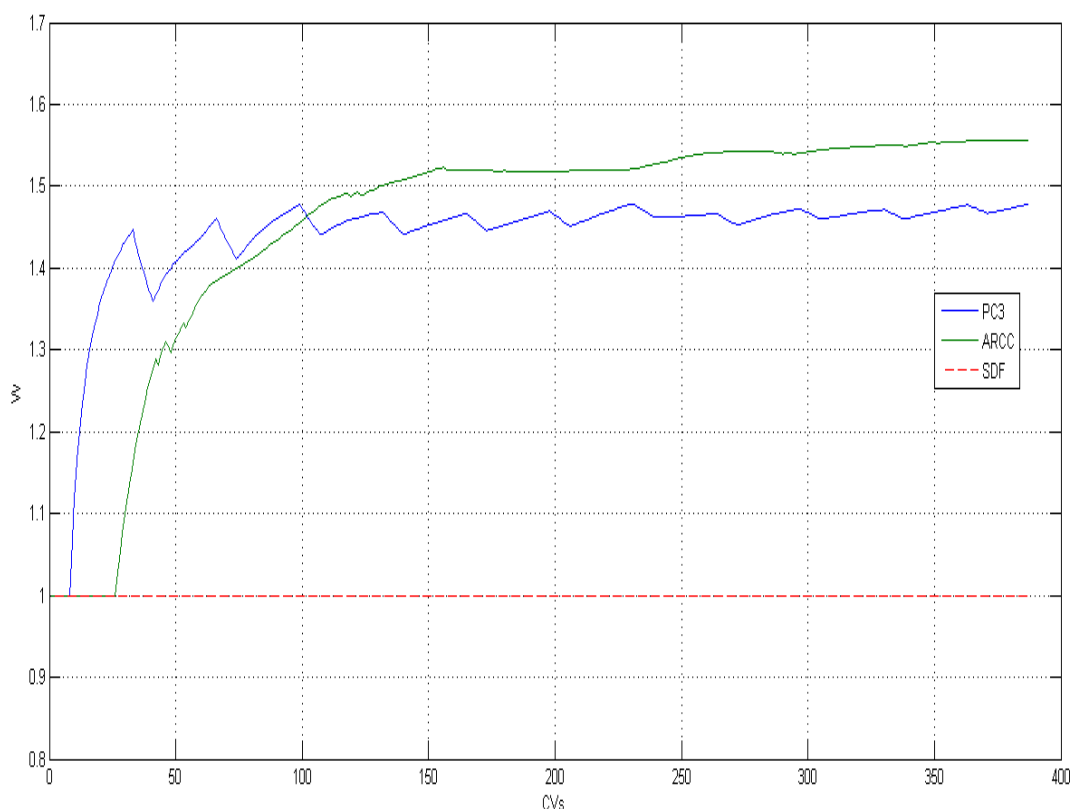
Μια πληρέστερη εικόνα, έχουμε στο παρακάτω σχεδιάγραμμα, στο οποίο φαίνεται η εξέλιξη του μέσου σχετικού σφάλματος ανακατασκευής, όπως αυτό ορίστηκε παραπάνω, καθόλη την διάρκεια, της προώθησης των διανυσμάτων πλαισίου, από τον έναν κόμβο στον επόμενο. Στον μηχανισμό ARCC, παρατηρούμε ότι, όσο προωθούνται πλαίσια, τόσο αυξάνεται το σφάλμα. Αντιθέτως, στον μηχανισμό PC3 δεν παρουσιάζονται αυξητικές τάσεις του σφάλματος, εκτός ενός διαστήματος στην αρχή της διαδικασίας, καθώς διέρχονται τα δεδομένα από τον έναν κόμβο στο επόμενο. Με αποτέλεσμα στο τέλος της εκτέλεσης του μηχανισμού PC3, το μέγεθος του σφάλματος να έχει διατηρηθεί σε χαμηλά επίπεδα, κατά πολύ μικρότερα από αυτά που κυμαίνεται το ARCC.



Σχήμα 11: Εξέλιξη του μέσου σχετικού σφάλματος ανακατασκευής τους μηχανισμούς PC3 (πράσινο) και ARCC (μπλε)

Σε αυτό το σημείο, και αφού συγκρίναμε ήδη τους δυο μηχανισμούς, από την οπτική γωνία του ενεργειακού κόστους και του σφάλματος ανακατασκευής, μεμονωμένα, τώρα θα χρησιμοποιήσουμε τις συνδυαστικές μετρικές, w και f , τις οποίες ορίσαμε προηγουμένως. Οι μετρικές αυτές λαμβάνουν υπόψη και τις δυο αυτές παραμέτρους ταυτόχρονα, στην προσπάθεια να έχουμε μια περισσότερο σφαιρική εκτίμηση της απόδοσης των δυο μηχανισμών.

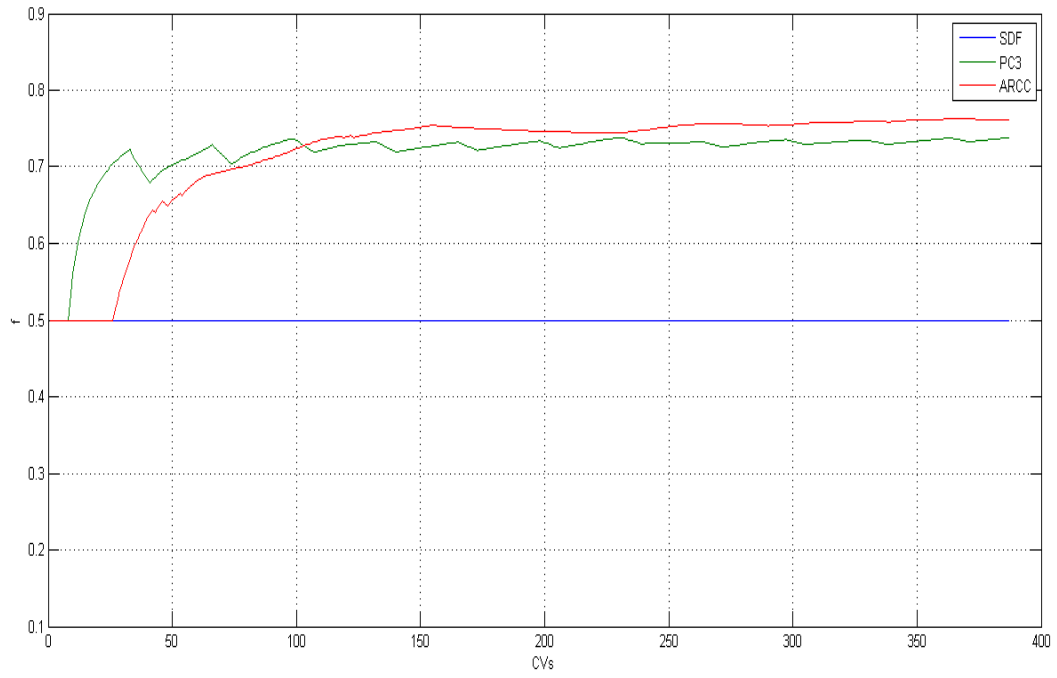
Από τον τύπο, της μετρικής w , είναι φανερό πως το μέτρο σύγκρισης είναι ο μηχανισμός απλής προώθησης δεδομένων πλαισίου, στον οποίο $w = 1$ για κάθε t . Έτσι έχοντας κοινό σημείο αναφοράς, μπορούμε να συγκρίνουμε και τους δυο μηχανισμούς αναμεταξύ τους.



Σχήμα 12: Συντελεστής απόδοσης για τους μηχανισμούς ARCC (πράσινο), PC3 (μπλε) και SDF (κόκκινο)

Η μέση τιμή της w , κατά την διάρκεια της προώθησης όλων των πλαισίων είναι $\tilde{w}=1.457$ για τον ARCC, και $\tilde{w}=1.440$ για τον PC3. Ενώ οι τελικές τιμές που παίρνουν αφού έχουν προωθηθεί όλα τα διανύσματα πλαισίου, είναι για τον ARCC $w=1.557$, ενώ για τον PC3 $w=1.478$. Παρατηρούμε λοιπόν, μια καλύτερη ολική απόδοση του μηχανισμού ARCC, της τάξεως του 5%.

Όσον αφορά τον άλλο συντελεστή f , που έχουμε ορίσει, παίρνει σαν μέση τιμή $\tilde{f}=0.7187$ για τον PC3 και $\tilde{f}=0.7188$ για τον ARCC. Ενώ οι τελικές τιμές που παίρνουν αφού έχουν προωθηθεί όλα τα διανύσματα πλαισίου, είναι για τον ARCC $f=0.762$, ενώ για τον PC3, $f=0.738$.



Σχήμα 13: Συντελεστής απόδοσης για τους μηχανισμούς ARCC (κόκκινο), PC3 (πράσινο) και SDF (μπλε)

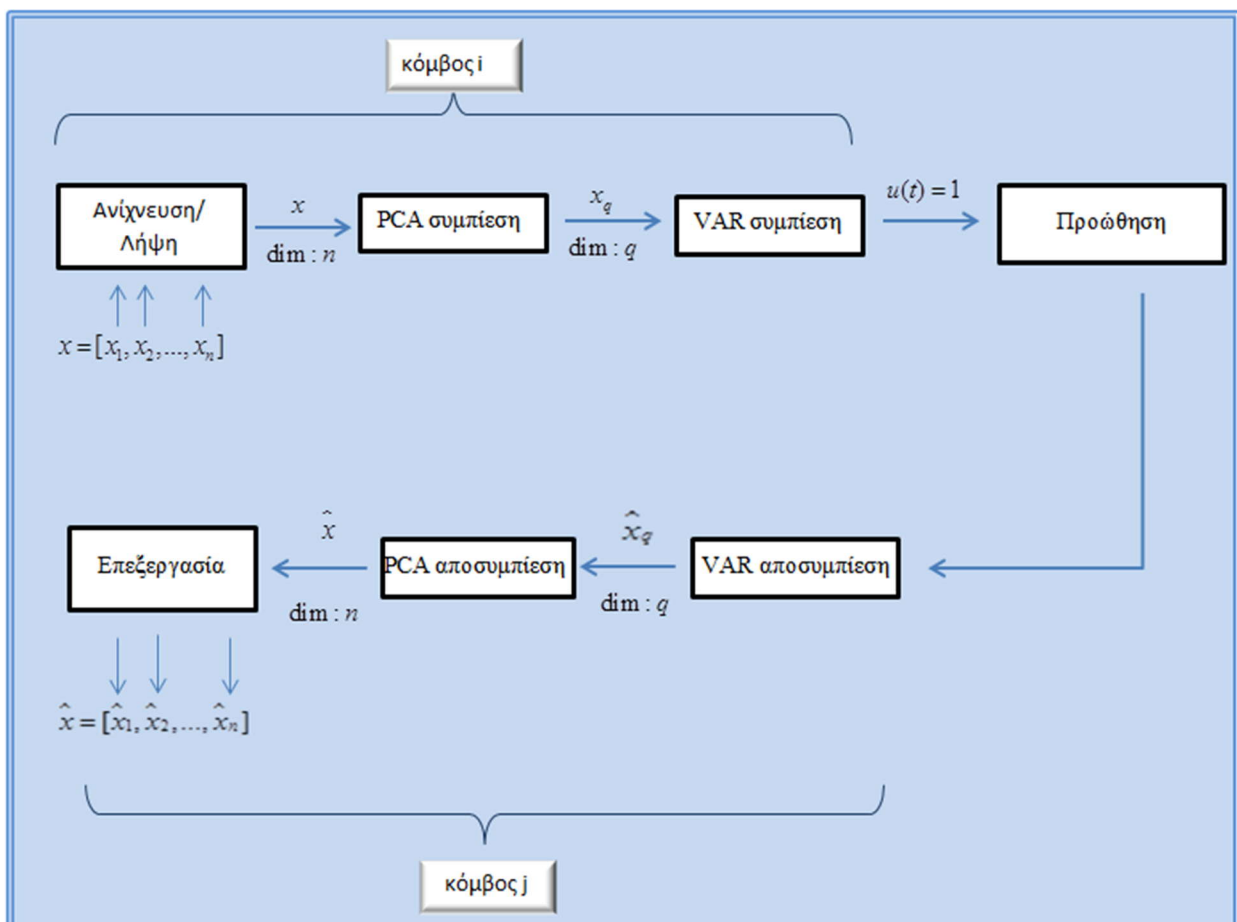
4 ΣΥΜΠΙΕΣΗ ΠΛΑΙΣΙΟΥ ΒΑΣΙΣΜΕΝΗ ΣΤΗΝ ΑΝΑΛΥΣΗ ΚΥΡΙΩΝ ΣΥΝΙΣΤΩΣΩΝ ΚΑΙ ΣΤΟ ΑΥΤΟΠΑΛΙΝΔΡΟΜΟ ΥΠΟΔΕΙΓΜΑ (PRINCIPAL COMPONENT-AUTOREGRESSION-BASED CONTEXT COMPRESSION – PCARCC)

Στην προσπάθειά μας να πετύχουμε την, κατά το δυνατό μεγαλύτερη, εξοικονόμηση ενέργειας, κατά την διαδικασία προώθησης των διανυσμάτων πλαισίου, μένοντας όμως πάντα σε έναν αποδεκτό επίπεδο όσον αφορά την ακρίβεια των δεδομένων, σε ένα ασύρματο δίκτυο αισθητήρων, προτείναμε έναν νέο μηχανισμό, τον ARCC και τον συγκρίναμε με έναν ήδη υπάρχον, τον PC3. Ένα συνοπτικό, αλλά αντιπροσωπευτικό συμπέρασμα που προκύπτει από αυτήν την διαδικασία σύγκρισης, είναι πως ο ARCC επιτυγχάνει μεγαλύτερη συμπίεση στις ανταλλασσόμενες πληροφορίες πλαισίου, αλλά και μεγαλύτερο σφάλμα, άρα μικρότερη ακρίβεια στα δεδομένα, σε σχέση με τον PC3.

Αυτό που επιδιώκουμε λοιπόν, είναι ένας μηχανισμός που συνδυάζει το πλεονέκτημα του ARCC συνοδευόμενο από το θετικό στοιχείο του PC3. Έναν τέτοιο μηχανισμό, ο οποίος βασίζεται στους δύο προηγούμενους, θα παρουσιάσουμε σε αυτήν την παράγραφο. Στην συνέχεια ο μηχανισμός αυτός θα συγκριθεί ως προς την συνολική αποδοτικότητα που επιτυγχάνει, σε σχέση πάντα με τους άλλους δυο και την απλή προώθηση δεδομένων.

4.1 Σχεδίαση-Υλοποίηση του PCARCC

Η βασική ιδέα αυτού του μηχανισμού απεικονίζεται στην παρακάτω εικόνα:



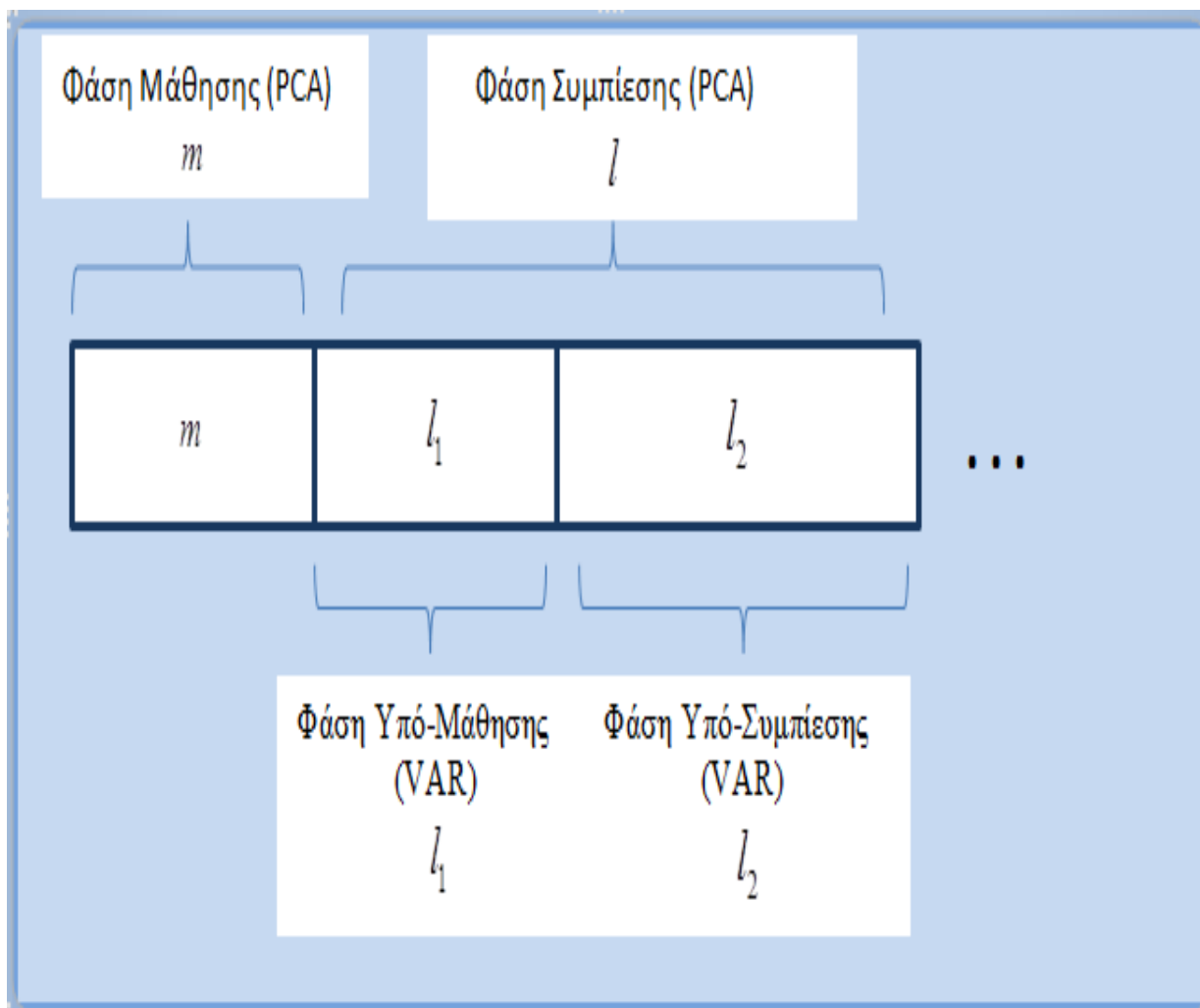
Εικόνα 6: Μηχανισμός συμπίεσης πλαισίου PCARCC

Ο κόμβος i ανιχνεύει/λαμβάνει ένα n -διάστατο διάνυσμα πλαισίου (x). Ο κόμβος συμπιέζει αυτό το διάνυσμα, κάνοντας χρήση της ανάλυσης κυρίων συνιστωσών, όπως δηλαδή ο PC3, σε ένα διάνυσμα διάστασης q (x_q), όπου $q < n$. Στη συνέχεια, στο ήδη μια φορά συμπιεσμένο διάνυσμα πλαισίου x_q εφαρμόζουμε μια επιπρόσθετη συμπίεση, βασιζόμενη αυτή την φορά στο αυτοπαλίνδρομο υπόδειγμα. Έτσι υπό την προϋπόθεση, ότι το σφάλμα πρόβλεψης της διαδικασίας VAR, είναι μικρότερο από το ανώτατο όριο που έχουμε θέσει, προωθούμε στον επόμενο κόμβο ένα σήμα αναπαραγωγής, $u(t)$. Ο κόμβος-παραλήπτης αναπαράγει με την αντίστροφη διαδικασία ένα n -διάστατο διάνυσμα πλαισίου (\tilde{x}), για περαιτέρω επεξεργασία (ή προώθηση σε επόμενους κόμβους).

Ο συγκεκριμένο μηχανισμός, αποτελείται, όπως και το PC3, από τις φάσεις μάθησης και συμπίεσης. Η φάση μάθησης είναι ακριβώς ίδια με την αντίστοιχη φάση στο PC3. Ο κόμβος i , δηλαδή, μαθαίνει τις κύριες συνιστώσες των τελευταίων m μετρήσεων που έλαβε/ανίχνευσε, ενώ παράλληλα προωθεί χωρίς επεξεργασία τα διανύσματα αυτά στο επόμενο κόμβο j . Συγκεκριμένα, ο κόμβος i , κατά την διάρκεια της φάσης μάθησης $1 \leq t \leq m$, αποστέλλει στον επόμενο κόμβο το πραγματικό διάνυσμα πλαισίου $x(t)$ και παράλληλα συγκεντρώνει τις m μετρήσεις, όταν συμπληρωθούν m μετρήσεις, μπορεί πλέον να κατασκευαστεί ο πίνακας Y , ο οποίος χρησιμοποιείται όπως είδαμε και παραπάνω για την συμπίεση των διανυσμάτων πλαισίου, μέσω της διαδικασίας $x_q(t) = Y^T x(t)$.

Έτσι λοιπόν, ακολουθώντας ακριβώς την ίδια λογική με το PC3, επιτυγχάνουμε την πρώτη συμπίεση στο PCARCC. Πλέον έχοντας τα συμπιεσμένα, μέσω ανάλυσης κυρίων συνιστωσών, διανύσματα πλαισίου x_q , προχωρούμε στην δεύτερη συμπίεση αυτών εφαρμόζοντας την διαδικασία του αυτοπαλίνδρομου υποδείγματος VAR. Η φάση συμπίεσης λοιπόν του PCARCC, διαφοροποιείται από την αντίστοιχη του PC3, αφού οι συμπιεσμένες πλέον, πληροφορίες πλαισίου δεν αποστέλλονται στον επόμενο κόμβο, όπως συμβαίνει στην περίπτωση του PC3, αλλά αποτελούν είσοδο, στον μηχανισμό ARCC, που θα ακολουθήσει. Άρα, λοιπόν η φάση συμπίεσης αποτελείται από μία επιπλέον φάση μάθησης (υπό-μάθησης) και εκ νέου συμπίεσης (υπό-συμπίεσης), αυτήν την φορά του μηχανισμού ARCC.

Όπως φαίνεται, αναλυτικότερα, στην παρακάτω εικόνα, ο κόμβος i , κατά την διάρκεια της φάσης υπό-μάθησης $m \leq t \leq l_1$, αποστέλλει στον επόμενο κόμβο, το συμπιεσμένο διάνυσμα πλαισίου x_q , και παράλληλα, δημιουργεί την μνήμη (μάθηση) για την διαδικασία VAR, που πρόκειται να ακολουθήσει, κατά αντίστοιχο τρόπο με τον οποίο στην αρχική φάση μάθησης (m), ο κόμβος i προωθούσε το πραγματικό διάνυσμα πλαισίου ενώ ταυτόχρονα κατασκεύαζε την μνήμη για την διαδικασία PCA, που επρόκειτο να ακολουθήσει.



Εικόνα 7: Φάσεις Λειτουργίας μηχανισμού PCARCC

Μέχρι εδώ λοιπόν, ο μηχανισμός PCARCC ακολουθεί, σχεδόν την ίδια λογική με τον PC3, εξαιρουμένης της δεύτερης δημιουργίας μνήμης, την οποία θα εκμεταλλευτεί σε αυτή την φάση το προτεινόμενο μοντέλο. Κατά την διάρκεια, της φάσης υπό-συμπύεσης $l_1 \leq t \leq l_2$, ο κόμβος i , με την βοήθεια του αυτοπαλίνδρομου μοντέλου υπολογίζει – προβλέπει μια τιμή $x'_q(t)$, για τα συμπιεσμένα δεδομένα $x_q(t), l_1 \leq t \leq l_2$, βασιζόμενο στις τελευταίες l_1 μετρήσεις που έλαβε, κατά την φάση υπό-μάθησης. Η τιμή του $x'_q(t)$ συγκρίνεται με τα συμπιεσμένα δεδομένα $x_q(t)$, και υπολογίζεται το *σφάλμα πρόβλεψης*. Αυτό είναι το εκτιμώμενο σφάλμα είναι που καθορίζει το τι θα επιλέξει ο κόμβος i .

Ο κόμβος i , ανάλογα, με το εκτιμώμενο σφάλμα, είτε προωθεί στον επόμενο κόμβο το $x_q(t)$, χωρίς κάποια επιπλέον επεξεργασία-συμπύεση, (σε αυτή την περίπτωση η διαδικασία είναι ακριβώς ίδια με αυτή στον PC3), ή επιλέγει να μην προωθήσει στον j , τα PCA συμπιεσμένα δεδομένα, αλλά το σήμα ανακατασκευής (εφαρμογή και δεύτερης συμπύεσης). Στην τελευταία περίπτωση, θα πρέπει με την σειρά του, ο επόμενος κόμβος j , να εκτελέσει μια αντίστοιχη διαδικασία πρόβλεψης, κάνοντας χρήση του $\text{VAR}(l_1)$, υπολογίζοντας κατά αυτό τον τρόπο το $\hat{x}_q(t)$ και μετά εφαρμόζοντας της

αντίστροφη διαδικασία PCA, μέσω του τύπου $\hat{x}(t) = Yx_q^T(t)$, παίρνει την προσέγγιση της αρχικής μέτρησης $x(t)$. Αυτή η διαδικασία που μόλις περιγράψαμε, είναι περιοδική, με περίοδο $m+l_1+l_2$, και εφαρμόζεται για όλα τα διερχόμενα δεδομένα από το κόμβο i .

4.2 Εκτίμηση και σύγκριση απόδοσης του PCARCC

Όπως είδαμε λοιπόν μέχρι στιγμής, ο προτεινόμενος μηχανισμός PCARCC, κάνει χρήση επιμέρους στοιχείων των προαναφερθέντων μηχανισμών PC3 και ARCC. Στο κομμάτι της εκτέλεσης, ο μηχανισμός μπορεί να ταυτιστεί πλήρως με τον PC3, κατά την περίπτωση εκείνη στην οποία, στη φάση υπό-συμπύεσης l_2 , το εκτιμώμενο σφάλμα είναι πάντα μεγαλύτερο από το μέγιστο δυνατό, οπότε και θα γίνεται μονίμως προώθηση των PCA συμπιεσμένων πλαισίων $x_q(t)$. Σε αντίθετη περίπτωση, στην φάση υπό-συμπύεσης, αν οι συνθήκες το επιτρέψουν, δύναται να αποστέλλεται συνεχώς το σήμα ανακατασκευής $u(t)$.

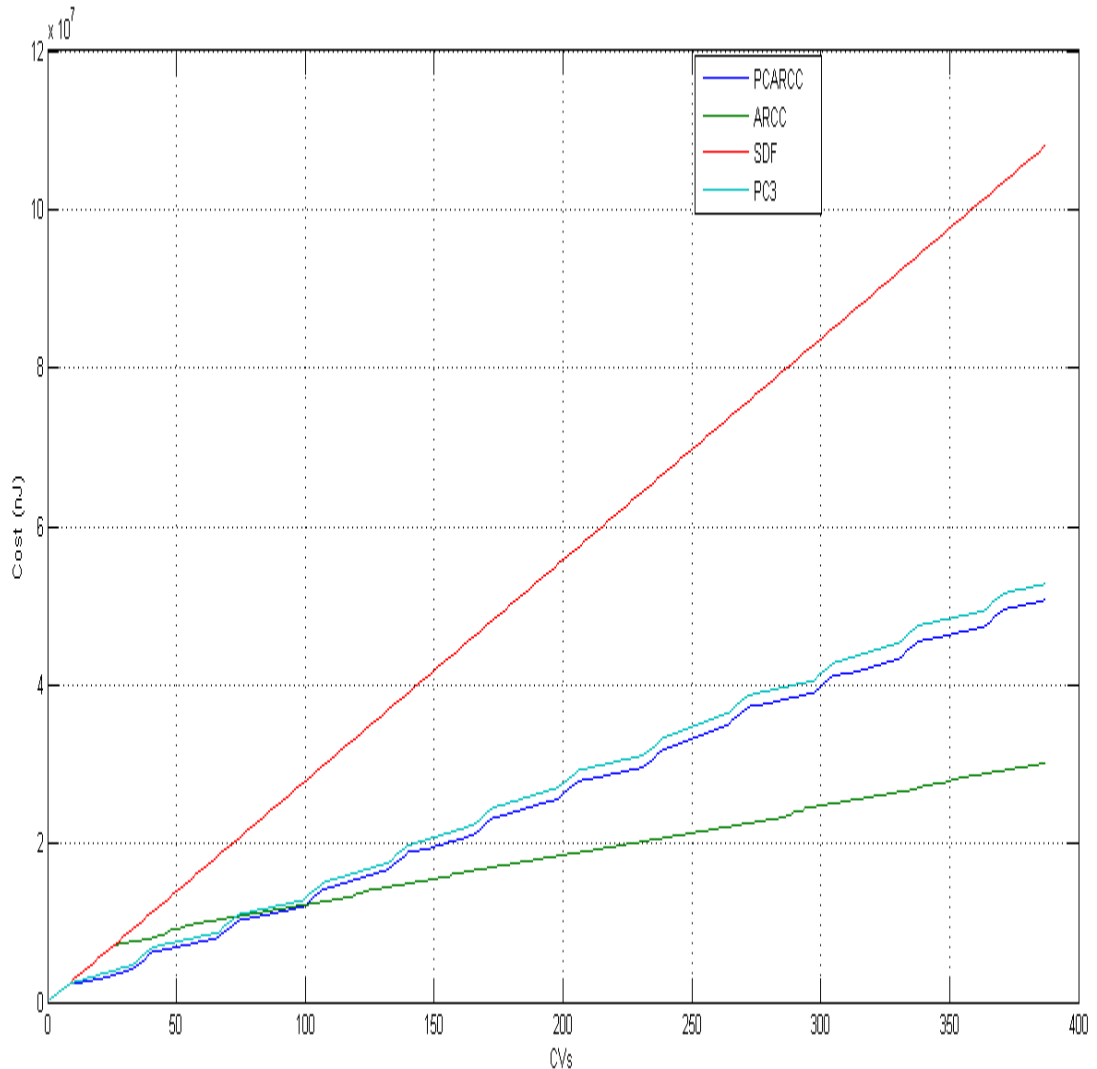
Στην μεριά του, επόμενου κόμβου j , οι πρώτες m μετρήσεις λαμβάνονται αναλλοίωτες, συνεπώς υπάρχει απόλυτη ακρίβεια δεδομένων και καθόλου εξοικονόμηση ενέργειας. Οι επόμενες l_1 λαμβάνονται συμπιεσμένες, αλλά μόνο με την μέθοδο κυρίων συνιστωσών, συνεπώς μέτριου επιπέδου εξοικονόμηση ενέργειας και επίσης μέτριου επιπέδου σφάλμα ανακατασκευής. Τέλος, στη τελευταία φάση l_2 της κάθε περιόδου, όπως αναφέρθηκε και παραπάνω, δεν έχουμε προκαθορισμένα αποτελέσματα όσον αφορά τον τρόπο εκτέλεσης, αλλά εξαρτώνται από το είδος των μετρήσεων κάθε φορά. Στην ιδανική περίπτωση όπου όλα τα πλαίσια αποστέλλονται υπό την μορφή σήματος ανακατασκευής, έχουμε υψηλό βαθμό στην ενεργειακή απόδοση και χαμηλό επίπεδο στην ακρίβεια των δεδομένων.

Πλέον, αφού παρουσιάσαμε αρκετά αναλυτικά την δομή και τον τρόπο λειτουργίας του προτεινόμενου μοντέλου PCARCC, μπορούμε να προβούμε σε μια σειρά συγκρίσεων, καλύπτοντας όλο το φάσμα της απόδοσης, του μηχανισμού PCARCC με τους PC3 και ARCC, καθώς και με αυτόν της απλής προώθησης δεδομένων SDF.

Τα δεδομένα που θα χρησιμοποιηθούν στην πειραματική διαδικασία είναι τα ίδια που χρησιμοποιήθηκαν και στις δύο προηγούμενες διαδικασίες σύγκρισης που πραγματοποιήθηκαν, δηλ 387 διανύσματα πλαισίου (Παράρτημα I), που έχουν ανιχνευτεί σε πραγματικές συνθήκες. Κάθε πλαίσιο αποτελείται από ένα διάνυσμα 7 μεταβλητών με περιεχόμενο (θερμοκρασία1, υγρασία1, θερμοκρασία2, υγρασία2, θερμοκρασία3, υγρασία3, ταχύτητα ανέμου), τα οποία έχουν δειγματοληπτηθεί με συχνότητα 10^{-2} Hz. Χρησιμοποιείται το μοντέλο κατανάλωσης ενέργειας Mica2. Η προσομοίωση έχει υλοποιηθεί στο περιβάλλον Matlab (Παράρτημα II & III).

Όπως αναφέραμε και παραπάνω, το PC3 κάνει χρήση ενός μηχανισμού προσαρμογής, για να ρυθμίζει την τιμή του διαστήματος συμπύεσης σε κάθε περίοδο, στο πλαίσιο της πειραματικής διαδικασίας δεν έγινε χρήση αυτού του μηχανισμού και επιλέχθηκαν σταθερά διαστήματα μάθησης ($m=8$) και συμπύεσης ($l=25$), κατά τέτοιο τρόπο που να προσεγγίζει τις μέσες τιμές του μήκους αυτών των διαστημάτων, στην περίπτωση που είχε χρησιμοποιηθεί ο ανάλογος μηχανισμός. Με ανάλογο τρόπο, για τον μηχανισμό PCARCC, έχουν επιλεγεί οι ίδιες τιμές για τα διαστήματα μάθησης και συμπύεσης, ενώ για την φάση της υπό-μάθησης έχει επιλεγεί επίσης σταθερή τιμή ίση με 3 ($p=3$), η οποία και καθορίζει το μήκος των διαστημάτων υπό-μάθησης και υπό-συμπύεσης.

Το πρώτο στοιχείο που θα μελετήσουμε, αποτελεί και τον πρωτεύοντα στόχο των τριών μηχανισμών, που δεν είναι άλλος από το ενεργειακό κόστος που καταναλίσκεται κατά την εφαρμογή τους. Στο σχήμα 14, βλέπουμε την εξέλιξη του κόστους καθόλη την διάρκεια, προώθησης των πλαισίων.



Σχήμα 14: Το ενεργειακό κόστος που καταναλίσκεται σε κάθε μηχανισμό

Διακρίνουμε λοιπόν, πως την μικρότερη κατανάλωση ενέργειας συνεχίζει να την έχει ο μηχανισμός ARCC, κάτι που εν πολλοίς ήταν αναμενόμενο, αφού ο βαθμός συμπίεσης, που παίζει και το μεγαλύτερο ρόλο, είναι πολύ μεγαλύτερος από αυτόν των άλλων δυο μηχανισμών. Όσον αφορά, το PCARCC είναι εμφανές ότι ακολουθεί στο μεγαλύτερο μέρος του την εξέλιξη του PC3, όπως αναφέραμε και πριν οι φάσεις μάθησης και υπό-μάθησης είναι παρόμοιες και στους δύο μηχανισμούς, η διαφοροποίηση έγκειται στην φάση υπό-συμπύεσης, και αυτή υπό όρους όπως προαναφέραμε. Λεπτομερέστερα, στην φάση μάθησης m , και οι δύο μηχανισμοί επιβαρύνονται από το κόστος προώθησης των πραγματικών πληροφοριών πλαισίου $x(t)$, στο τέλος αυτής της φάσης, υπάρχει το κόστος υπολογισμού των κυρίων

συνιστωσών και του αντίστοιχου πίνακα που θα χρησιμοποιηθεί για την δημιουργία των συμπιεσμένων πλαισίων.

Ακολουθεί στην φάση υπό-μάθησης l_1 , το κόστος προώθησης των συμπιεσμένων πλαισίων.

Μέχρις εδώ λοιπόν $m+l_1$, τα κόστη των διαδικασιών ταυτίζονται, η διαφοροποίηση όπως έχουμε τονίσει υφίσταται στην φάση l_2 της κάθε περιόδου. Σε αυτή την φάση, υπό-συμπύεσης, το κόστος του PC3 συνεχίζει να αποτελείται από την προώθηση των συμπιεσμένων διανυσμάτων πλαισίου, όπως δηλαδή συνέβαινε και στην φάση υπό-μάθησης. Στον μηχανισμό PCARCC, όμως σε αυτήν την φάση τα επιμέρους κόστη που συνεισφέρουν στο συνολικό κόστος της διαδικασίας είναι διαφορετικά. Υπάρχει το κόστος υπολογισμού-πρόβλεψης που επιφέρει η διαδικασία VAR, σε κάθε ένα από τα διανύσματα πλαισίου στο διάστημα l_2 , και το ανάλογο κόστος προώθησης, είτε του συμπιεσμένου πλαισίου, ή του σήματος αναπαραγωγής.

Φαίνεται λοιπόν και από την αναπαράσταση του κόστους των διαδικασιών, ότι το όφελος που μπορεί να επιτύχει ο PCARCC σε σχέση με το PC3, εντοπίζεται μονό στην φάση υπό-συμπύεσης της κάθε περιόδου. Σε περίπτωση απόλυτης ταύτισης των δυο διαδικασιών, στο κομμάτι της εκτέλεσης, ο μηχανισμός PCARCC θα έχει καταναλώσει μεγαλύτερο ποσό ενέργειας, αφού εμπεριέχει όπως αναφέραμε το επιμέρους κόστος του υπολογισμού που οφείλεται στην διαδικασία VAR.

$$\text{PeriodicCost}_{PC3} = \text{TrasmCost}_m + \text{PCACompres} + \text{TransmCost}_{l_1}$$

$$\begin{aligned} \text{PeriodicCost}_{PCARCC} = & \text{TrasmCost}_m + \text{PCACompres} + \text{TransmCost}_{l_1} \\ & + \text{VARCost}_{l_2} + \text{TransmCost}_{l_2} \end{aligned}$$

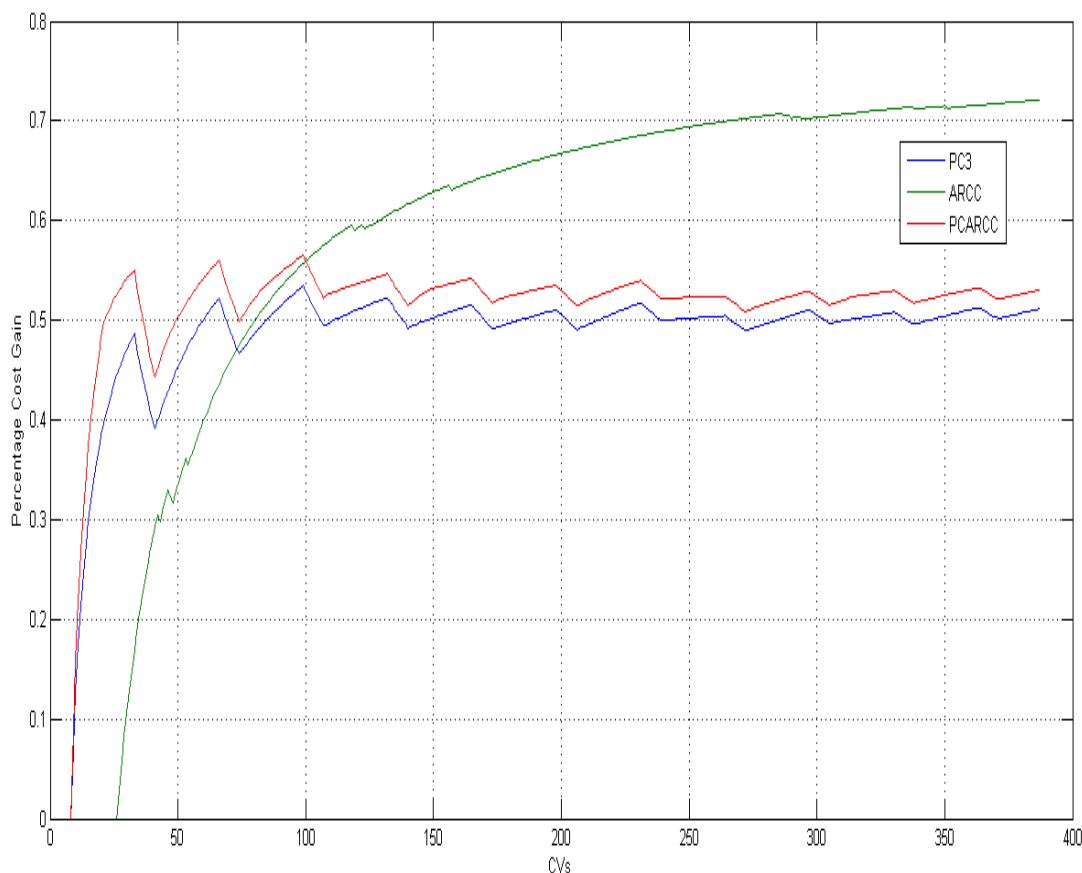
Από τις δυο εξισώσεις που μας δίνουν το περιοδικό κόστος της κάθε διαδικασίας, είναι εμφανές πως το σημείο υπεροχής του PCARCC σε σχέση με το PC3 έγκειται στο γεγονός πως, συνήθως (εξαρτάται από την φύση και την σχέση μεταξύ των διανυσμάτων πλαισίου) ισχύει πως:

$$\text{TransmCost}_1 > \text{TrasmCost}_{l_1} + \text{VARCost}_{l_2} + \text{TransmCost}_{l_2}$$

Στην παρούσα πειραματική διαδικασία, στον μηχανισμό PCARCC υπάρχει μια βελτίωση της τάξης του 3,88%, όσον αφορά την ενέργεια που καταναλώνεται, σε σχέση με τον μηχανισμό PC3. Όπως έχουμε αναφέρει, την καλύτερη ενεργειακή απόδοση την παρουσιάζει φυσικά ο ARCC, ο οποίος εν συγκρίσει με τον PCARCC, καταναλώνει 40% λιγότερη ενέργεια.

Για μια συνολικότερη προσέγγιση, μπορούμε να δούμε την παρακάτω αναπαράσταση, στην οποία φαίνεται, καθόλη την διάρκεια προώθησης των δεδομένων πλαισίου, το

ποσοστιαίο όφελος που πετυχαίνει η κάθε διαδικασία σε σχέση με το ενεργειακό κόστος της απλής προώθησης όλων των πλαισίων.

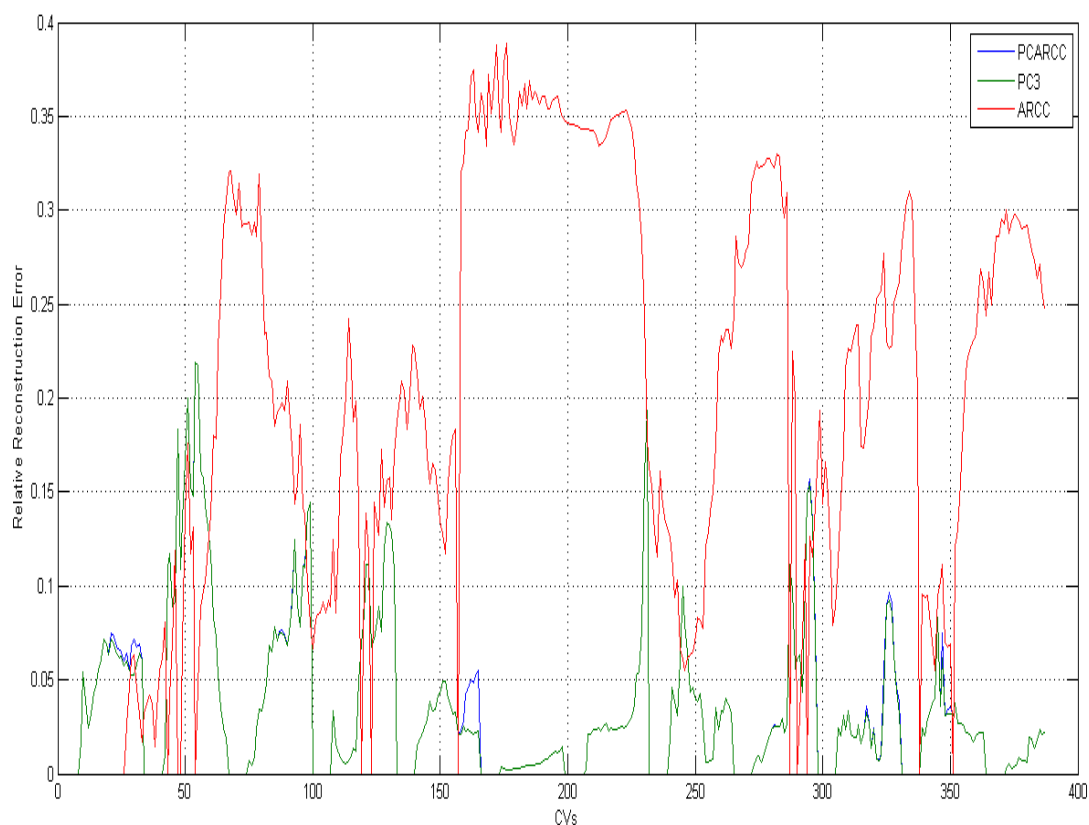


Σχήμα 15: Ποσοστιαίο ενεργειακό κέρδος των ARCC (πράσινο), PC3 (μπλε) και PCARCC (κόκκινο) σε σχέση με την SDF

Το ποσοστιαίο όφελος, που έχει επιτύχει ο κάθε μηχανισμός, μετά και την προώθηση του τελευταίου πλαισίου είναι 72% ο ARCC, 53% ο PCARCC και 51% ο PC3.

Επόμενη πτυχή της σύγκρισης των τριών μηχανισμών είναι η ακρίβεια των δεδομένων, δηλαδή το σφάλμα ανακατασκευής του ληφθέντος, από τον κόμβο j , διανύσματος πλαισίου σε σχέση με το πραγματικό πλαίσιο που διήλθε/ανιχνεύτηκε και προωθήθηκε από τον κόμβο i .

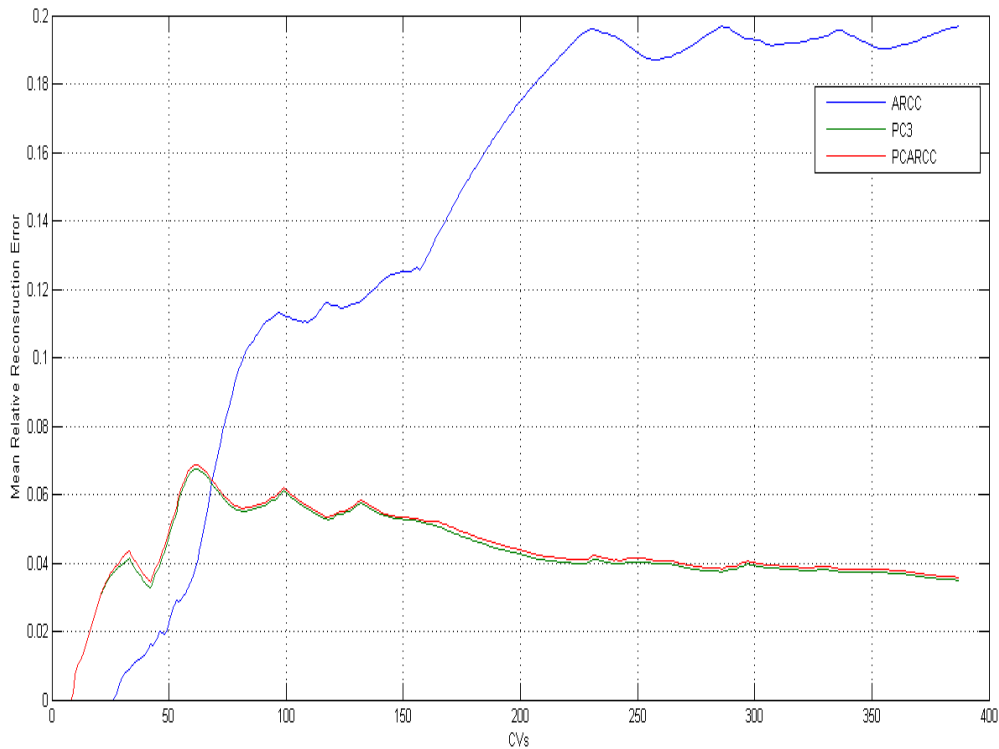
Οι μηχανισμοί, PC3 και PCARCC, όσον αφορά το κομμάτι της ακρίβειας δεδομένων, παρουσιάζουν παρόμοια συμπεριφορά στις φάσεις μάθησης (m) και υπό-μάθησης (l_1), αφού και στις δύο διαδικασίες προωθούνται τα ίδια ακριβώς δεδομένα στον επόμενο κόμβο, η διαφοροποίηση έγκειται στην φάση υπό-συμπύεσης (l_2), όπου υπό όρους ο PCARCC, προωθεί ένα σήμα ανακατασκευής. Συνεπώς η όποια διαφοροποίηση υπάρχει σχετικά με το σφάλμα ανακατασκευής των δυο διαδικασιών θα υπάρχει σε αυτή την τελευταία φάση. Αυτό είναι ευδιάκριτο και από την γραφική αναπαράσταση των σφαλμάτων ανακατασκευής, όπως φαίνεται παρακάτω, η πράσινη (PC3) και η μπλε (PCARCC) καμπύλη ταυτίζονται στο μεγαλύτερο μέρος τους ($m+l_1$).



Σχήμα 16: Το σχετικό σφάλμα ανακατασκευής σε κάθε ένα δάνυσμα πλαισίου, στους μηχανισμούς ARCC (κόκκινο), PCARCC (μπλε) και PC3 (πράσινο)

Το μέσο σχετικό σφάλμα ανακατασκευής, που αποδίδει αντιπροσωπευτικότερα την ακρίβεια των δεδομένων, είναι ως εξής : 3.50% για PC3, 3.58% για τον PCARCC και 19.68% για τον ARCC.

Η εξέλιξη του μέσου σχετικού σφάλματος ανακατασκευής και στους τρεις μηχανισμούς, φαίνεται στο παρακάτω σχήμα, διακρίνουμε πως PC3 και PCARCC, έχουν σχεδόν ταυτόσημη συμπεριφορά, και περιορίζουν την τιμή του μέσου σφάλματος σε χαμηλά-αποδεκτά επίπεδα.

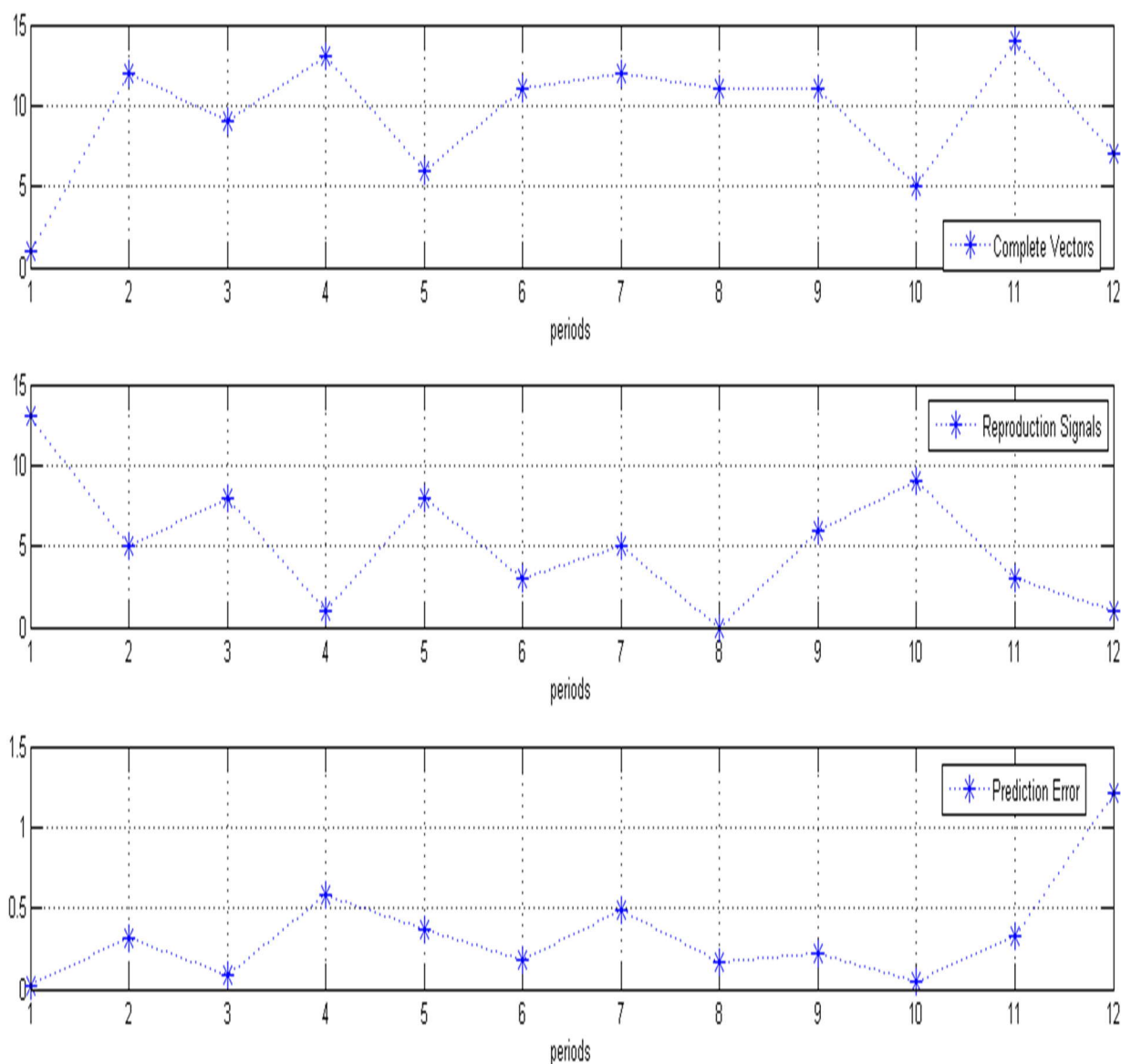


Σχήμα 17: Εξέλιξη του μέσου σχετικού σφάλματος ανακατασκευής στους μηχανισμούς PC3(πρασινό) ,PCARCC (κόκκινο) και ARCC (μπλε)

Σε αυτό το σημείο αξίζει να αναλύσουμε λίγο παραπάνω την συμπεριφορά του μηχανισμού PCARCC, και πιο συγκεκριμένα στο διάστημα συμπίεσης (υπό-μάθηση & υπό-συμπύεση) όπως αυτό περιγράφηκε παραπάνω και αναπαρίσταται στην εικόνα 7.

Σύμφωνα με τα αποτελέσματα της προσομοίωσης που εκτελέσαμε η διαδικασία προώθησης όλων των πληροφοριών πλαισίου από τον κόμβο i στον επόμενο j διαρκεί 12 περιόδους. Σε κάθε μια αυτές τις περιόδους στο διάστημα συμπίεσης εκτελείται ο μηχανισμός ARCC, όχι στα πραγματικά διανύσματα πλαισίου αλλά στις κύριες συνιστώσες αυτών όπως αυτές διαμορφώνονται από την φάση μάθησης της κάθε περιόδου. Στα 12 διαστήματα συμπίεσης συνολικά, σύμφωνα με τα στοιχεία της προσομοίωσης, στον κόμβο i βρίσκονται προς προώθηση 291 διανύσματα (βασικές συνιστώσες), από αυτά μόνο τα 62, δηλ. ποσοστό 21% αποστέλλεται μέσω του σήματος ανακατασκευής, το υπόλοιπο 79% (229) των διανυσμάτων προωθούνται αυτούσια στον επόμενο κόμβο. Στον μηχανισμό ARCC, όταν αυτός εκτελούταν για όλα τα διανύσματα πλαισίου, το ποσοστό των αυτούσιων προωθούμενων πλαισίων ήταν 10% επί του συνόλου αυτών. Βλέπουμε λοιπόν πως υπάρχει μια ιδιαίτερως σημαντική διαφοροποίηση, που επηρεάζει βεβαίως και την συνολική απόδοση του PCARCC, στο σκέλος την ενέργειας. Αυτήν την διαφοροποίηση μπορούμε να την αποδώσουμε σε δύο παράγοντες: ο πρώτος είναι το γεγονός πως έχουμε 12 φάσεις υπό-μάθησης, στην διάρκεια των οποίων, εξ' ορισμού, τα πλαίσια προωθούνται αυτούσια. Αφαιρώντας λοιπόν αυτά τα πλαίσια, των φάσεων υπό-μάθησης, από το συνολικό αριθμό απομένουν 191. Συνεπώς το ποσοστό των αυτούσια απεσταλμένων πλαισίων είναι 62%. Παρατηρούμε λοιπόν, πως ακόμη και μετά από αυτή την διευκρίνιση, η πλειοψηφία των σημάτων αποστέλλονται ακριβώς όπως είναι στον επόμενο κόμβο. Ο δεύτερος και σημαντικότερος παράγοντας, έχει να κάνει με την συσχέτιση που υπάρχει

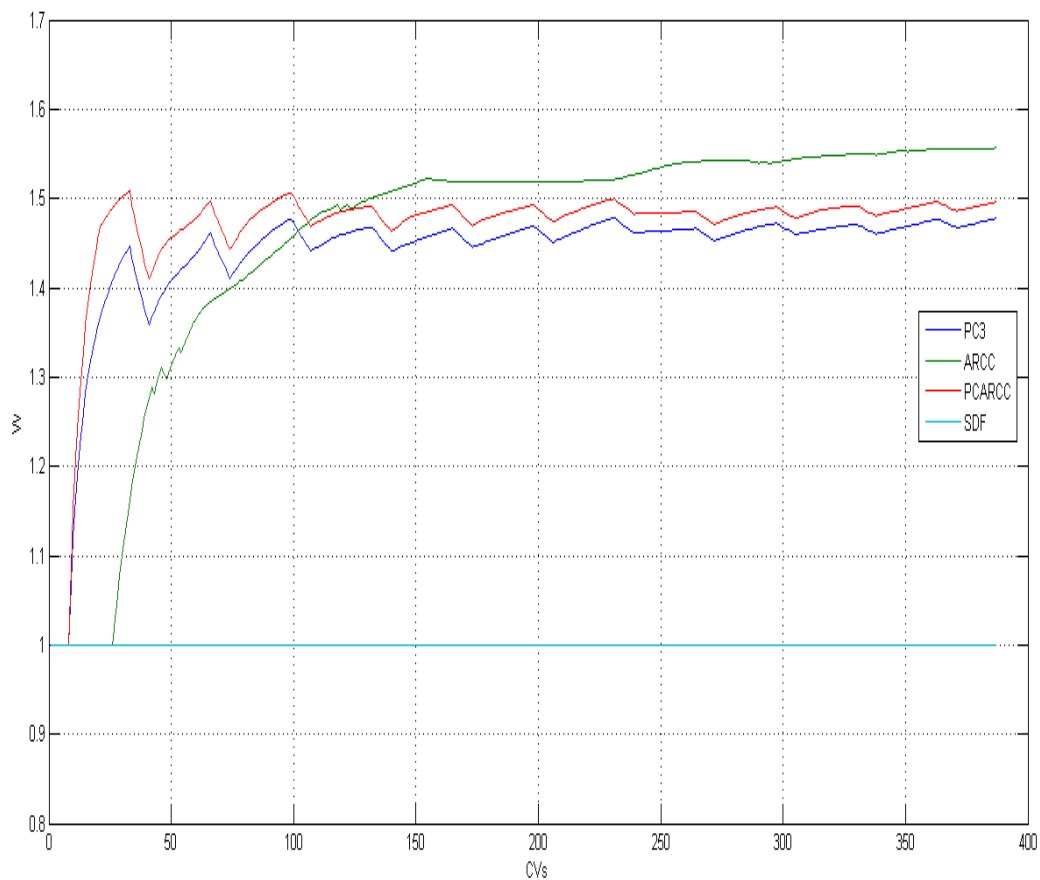
μεταξύ των προς αποστολή διανυσμάτων. Όπως αναφέραμε και παραπάνω οι διαδικασίες VAR και PCA, στις οποίες βασίζονται οι μηχανισμοί PCARCC, ARCC και PC3, οφείλουν την αποτελεσματικότητά τους, σε μεγάλο βαθμό, στο γεγονός πως τα δεδομένα που λαμβάνουν ως είσοδο είναι εξαρτημένα μεταξύ τους. Κάτι τέτοιο λοιπόν, συμβαίνει όντως με τα αρχικά διανύσματα πλαισίου του παραδείγματος μας, αλλά και γενικότερα με κάθε είδους πραγματική πληροφορία πλαισίου. Στον μηχανισμό PCARCC όμως τα δεδομένα εισόδου της διαδικασίας VAR, είναι οι κύριες συνιστώσες, οι οποίες δεν έχουν μεγάλη συσχέτιση μεταξύ τους συνεπώς η πρόβλεψη που πραγματοποιείται δεν πληροί τα κριτήρια που τίθενται τις περισσότερες φορές, οπότε δεν προωθείται σήμα ανακατασκευής στον επόμενο κόμβο αλλά ολόκληρη η πληροφορία. Αριθμητικά αυτό καταδεικνύεται από το γεγονός πως ενώ στον ARCC το σφάλμα τα πρόβλεψης που γίνεται στο κόμβο i , για το επόμενο διάνυσμα με βάση τα προηγούμενα, είναι κατά μέσο όρο στο 2%, το αντίστοιχο μέσο σφάλμα πρόβλεψης και των 12 φάσεων υπό-συμπύεσης στον PCARCC είναι 33%. Παρακάτω βλέπουμε και γραφικά αυτό που μόλις περιγράψαμε, σε κάθε μια από τις 12 φάσεις υπό-συμπύεσης απεικονίζεται το μέσο σφάλμα πρόβλεψης, καθώς και το πόσα πλαίσια προωθήθηκαν αυτούσια και πόσα αντικαταστάθηκαν από σήματα ανακατασκευής.



Σχήμα 18: Πλήθος αυτούσιων προωθούμενων διανυσμάτων, σημάτων ανακατασκευής και μέσο σφάλμα πρόβλεψης σε κάθε μια από τις περιόδους του PCARCC

Έχουμε διαπιστώσει, λοιπόν, από την μέχρι στιγμής σύγκριση, από την οπτική του ενεργειακού κόστους και της ακρίβειας των δεδομένων, πως το PCARCC επιτυγχάνει καλύτερη απόδοση όσον αφορά το ενεργειακό κόστος από το PC3, και επίσης είναι αποτελεσματικότερο όσον αφορά την ακρίβεια των δεδομένων από το ARCC. Ο πρωταρχικός μας στόχος δηλαδή επετεύχθη, υλοποιήσαμε έναν μηχανισμό (PCARCC) που απολαμβάνει τα θετικά στοιχεία των δύο προηγούμενων (ARCC & PC3).

Ωστόσο για να είμαστε περισσότερο συγκεκριμένοι, ιδίως όσον αφορά το ποσοτικό κομμάτι, θα πρέπει να εξετάσουμε και τους τρεις μηχανισμούς, ως προς την μετρική w , η οποία λαμβάνει υπόψη της ταυτόχρονα τις παραμέτρους ενεργειακού κόστους και ακρίβειας δεδομένων, με αποτέλεσμα να έχουμε μια ολοκληρωμένη και σφαιρική προσέγγιση στο θέμα της εκτίμησης-σύγκρισης των μοντέλων.

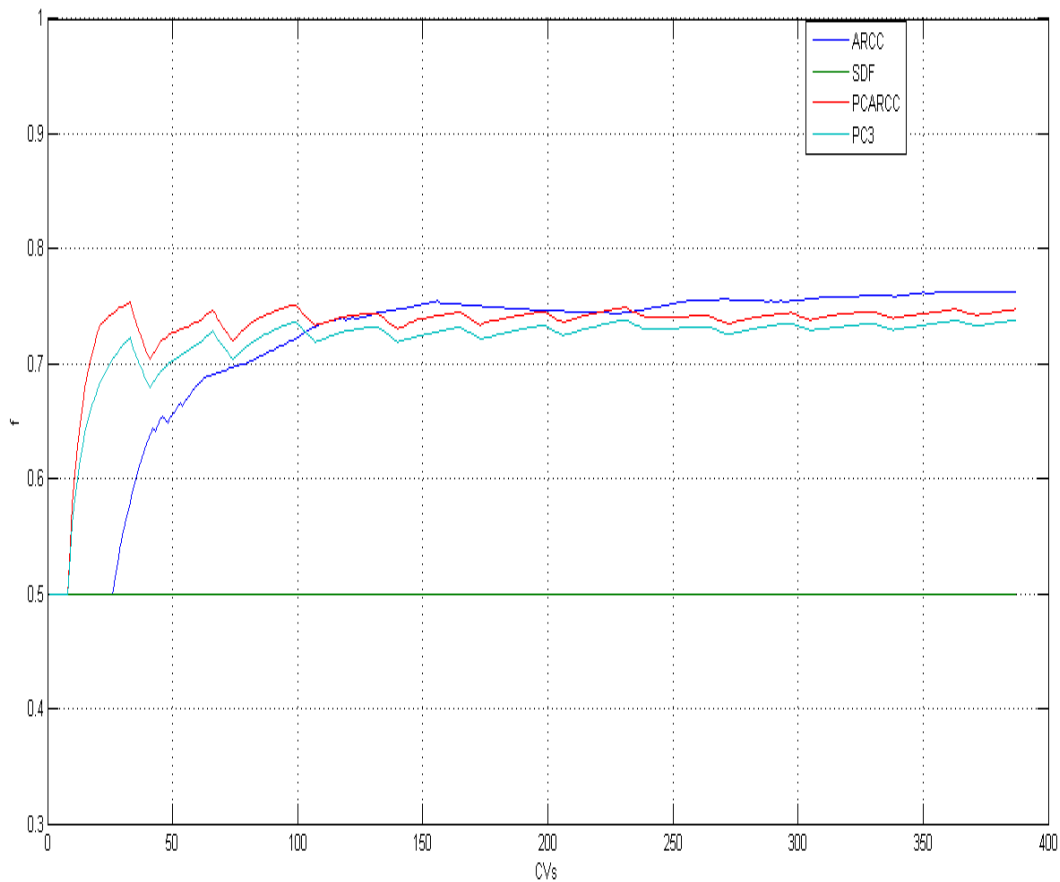


Σχήμα 19: Συντελεστής απόδοσης w για τους μηχανισμούς ARCC (πράσινο), PC3 (μπλε), PCARCC (κόκκινο) και SDF (θαλασσί)

Στην παραπάνω παράσταση (σχήμα 19), παρατηρούμε την εξέλιξη που παρουσιάζει η μετρική w , καθώς προωθούνται τα πλαίσια από το έναν κόμβο στο άλλον, έως ότου προωθηθεί και το τελευταίο. Βλέπουμε λοιπόν ότι στο τέλος της συγκεκριμένης διαδικασίας, ο PCARCC αποδίδει $w=1.50$, ο PC3 $w=1.48$ και ο ARCC $w=1.56$. Παρατηρούμε από την εικόνα πως ο PC3 και PCARCC παρουσιάζουν μια πριονωτή συμπεριφορά, η οποία εν μέρει οφείλεται στην συνεχή διαδοχή των φάσεων μάθησης και συμπύεσης. Από την άλλη, η ARCC παρουσιάζει μια διαρκής αύξηση στην απόδοση της, με αυξητικό ρυθμό στην αρχή ενώ στη συνέχεια με μειούμενο. Άρα στην

παρούσα πειραματική διαδικασία, με τα συγκεκριμένα διανύσματα πλαισίου, ο ARCC μας προσφέρει καλύτερη συνολική απόδοση, δεδομένου της μετρικής w που έχουμε ορίσει.

Όσον αφορά την μετρική f , οι τιμές που μας δίνει για κάθε μηχανισμό, αφού του έχει ολοκληρωθεί η προώθηση όλων των ανταλλασσόμενων πληροφοριών πλαισίου, έχει ως εξής: για τον ARCC $f = 0.762$, για τον PCARCC $f = 0.747$ και τέλος για τον PC3 $f = 0.738$, στο σχεδιάγραμμα που ακολουθεί φαίνεται η εξέλιξη του συντελεστή σε κάθε έναν μηχανισμό.



Σχήμα 20: Συντελεστής απόδοσης f για τους μηχανισμούς ARCC (μπλε), PC3 (θαλασσί), PCARCC (κόκκινο) και SDF (πράσινο)

Σε αυτό το σημείο, αξίζει να τονίσουμε πως τα τελευταία αποτελέσματα, που αφορούν τις μετρικές w και f , δεν καθορίζουν μια εκ των προτέρων υπεροχή της ARCC, υπό οποιεσδήποτε συνθήκες, αλλά μια εκ των πραγμάτων καλύτερη απόδοση υπό τις συγκεκριμένες προϋποθέσεις, κάτω από τις οποίες εκτελέστηκε η παρούσα πειραματική διαδικασία προώθησης των πλαισίων. Επίσης, ανάλογα με το ύψος και τις απαιτήσεις, που έχει η κάθε εφαρμογή επίγνωσης πλαισίου, μπορεί να σταθμίζονται κάθε φορά με δυσανάλογο τρόπο οι παράμετροι που καθορίζουν την απόδοση του κάθε μηχανισμού,

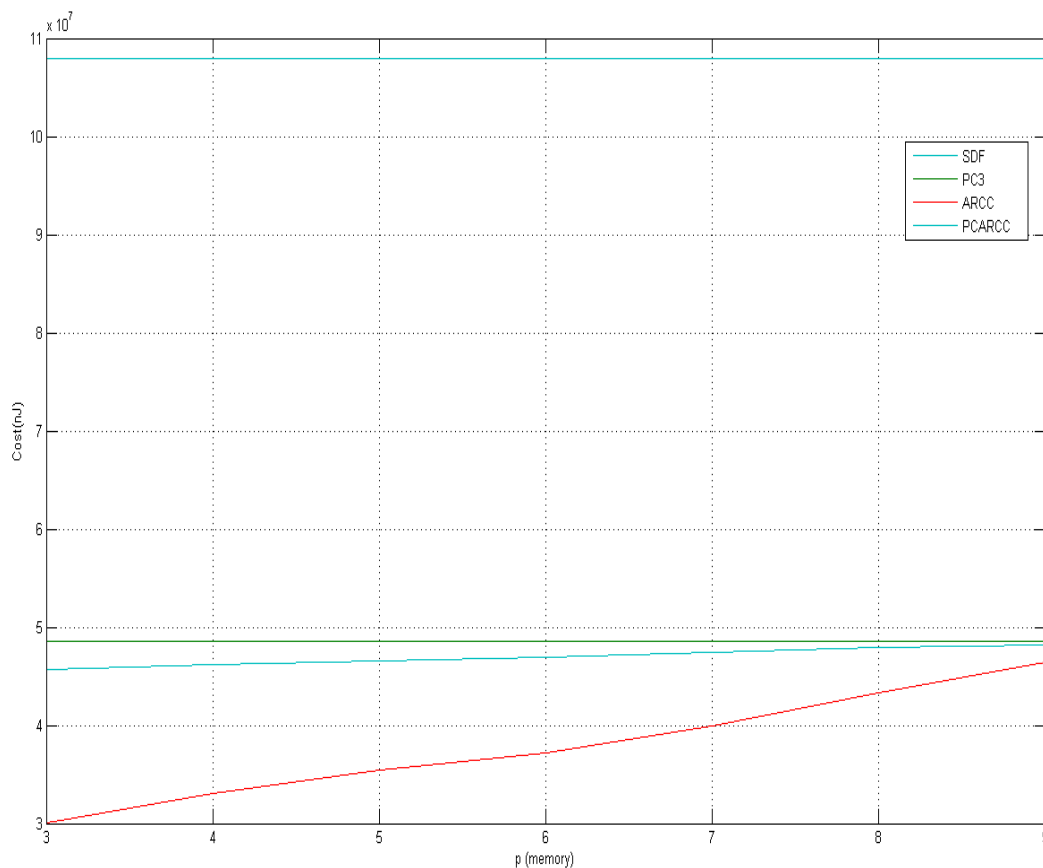
και όχι να λαμβάνονται ισότιμα υπόψη όπως συμβαίνει στις μετρικές ,που ορίσαμε. Για παράδειγμα κάποια εφαρμογή πιθανώς να είναι περισσότερο ευαίσθητη στην ακρίβεια των δεδομένων και συνεπώς η αντίστοιχη μετρική που θα πρέπει να χρησιμοποιηθεί να δίνει μεγαλύτερο βάρος στο σφάλμα ανακατασκευής από ότι στο ενεργειακό κόστος. Εμείς, στο τρόπο καθορισμού των μετρικών ακολουθήσαμε μια μέση προσέγγιση , που καλύπτει εν μέρει τις περισσότερες εφαρμογές και λαμβάνει υπόψη της, τους περιορισμένους πόρους των κόμβων ανίχνευσης (μνήμη, ενέργεια) και του ασύρματου δικτύου (εύρος ζώνης).

5 ΣΥΜΠΕΡΙΦΟΡΑ ΤΩΝ ΜΗΧΑΝΙΣΜΩΝ ARCC, PC3 ΚΑΙ PCARCC ΜΕΤΑΒΑΛΛΟΝΤΑΣ ΤΙΣ ΣΤΑΘΕΡΕΣ (p, m, l, MAX_{error})

Μέχρι τώρα στην εκτέλεση όλων των πειραματικών διαδικασιών λάβαμε κάποιες τιμές ως σταθερές. Στη διαδικασία VAR, που εκτελέστηκε στον μηχανισμό ARCC και PCARCC, είχαμε ορίσει εξ' αρχής σαν μνήμη της διαδικασίας την τιμή $p=3$ και σαν μέγιστο επιτρεπτό σφάλμα πρόβλεψης $MAX_{error}=10\%$, στους μηχανισμούς PC3 και PCARCC λάβαμε σταθερά διαστήματα μάθησης και συμπίεσης ($m=8, l=25$). Σε αυτό το εδάφιο θα παρατηρήσουμε και θα αναλύσουμε την συμπεριφορά των προαναφερόμενων μηχανισμών, όσον αφορά το καταναλισκόμενο ενεργειακό κόστος και το σφάλμα ανακατασκευής τους, μεταβάλλοντας τις μέχρι πρότινος σταθερές.

5.1 Μεταβλητή μνήμη διαδικασίας VAR (p)

5.1.1 Μνήμη διαδικασίας VAR (p) – Ενεργειακό κόστος



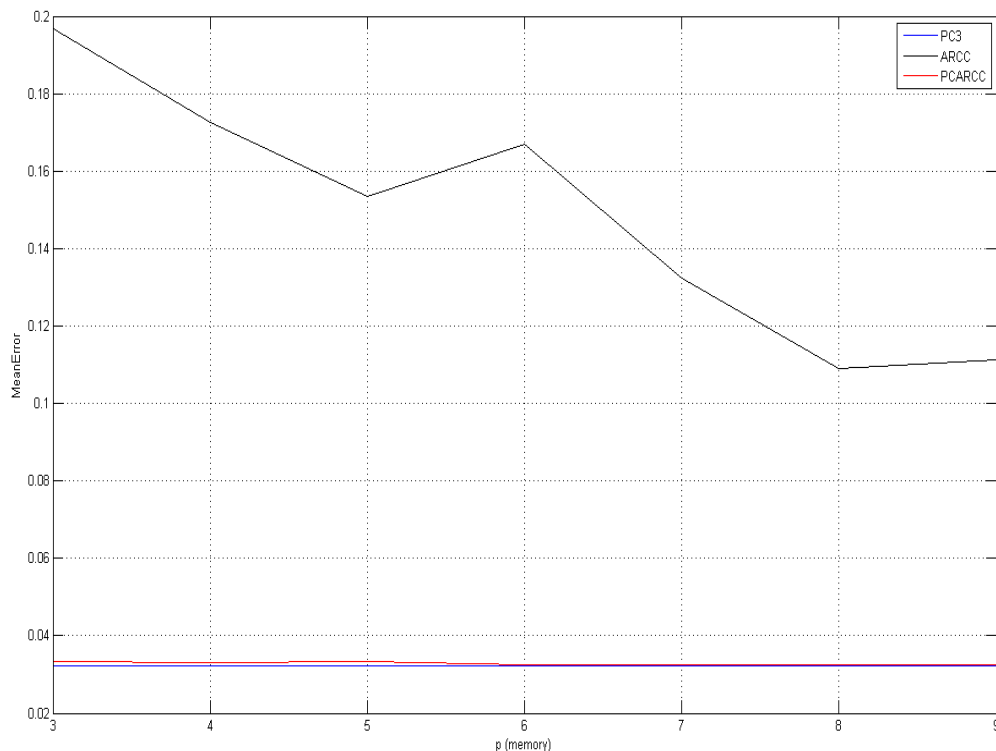
Σχήμα 21: Σχέση μεταξύ μνήμης διαδικασίας VAR και Ενεργειακού κόστους για τους μηχανισμούς PCARCC(θαλασσί), PC3(πράσινο), SDF (μπλε) και ARCC (κόκκινο)

Στην γραφική παράσταση (σχήμα 21), που απεικονίζει την σχέση μεταξύ μνήμης της διαδικασία αυτοπαλίνδρομου υποδείγματος και ενεργειακού κόστους, διακρίνουμε πως καθώς αυξάνεται η μνήμη, έχουμε και αύξηση του ενεργειακού κόστους που απαιτείται για την ολοκλήρωση της διαδικασίας στους μηχανισμούς, ARCC και PCARCC. Αυτό

είναι αναμενόμενο καθώς η αύξηση του μεγέθους της μνήμης της διαδικασίας VAR, συνεπάγεται πως περισσότερα δείγματα προωθούνται αυτούσια - χωρίς να υποστούν κάποιο είδος συμπίεσης - από τον έναν κόμβο στον επόμενο, άρα καταναλώνεται περισσότερη ενέργεια.

Στους μηχανισμούς, PC3 και SDF, οι οποίοι δεν κάνουν χρήση της VAR, όπως είναι φυσικό το κόστος παραμένει σταθερό ανεξαρτήτου του μεταβαλλόμενου μεγέθους. - Υπάρχουν στην παράσταση για λόγους σύγκρισης των μεγεθών.

5.1.2 Μνήμη διαδικασίας VAR (p) – Μέσο Σφάλμα ανακατασκευής

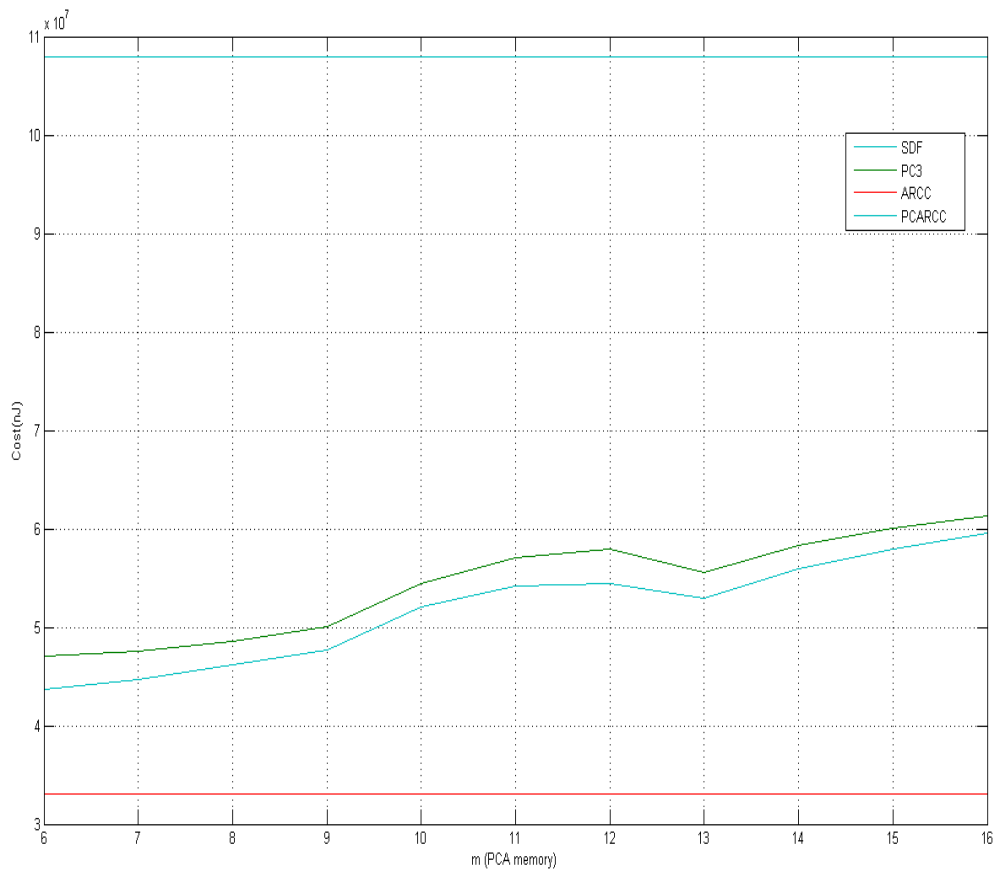


Σχήμα 22: Σχέση μεταξύ μνήμης διαδικασίας VAR (p) και μέσου σφάλματος ανακατασκευής για τους μηχανισμούς PCARCC(κόκκινο), PC3(μπλε), και ARCC (πράσινο)

Στην γραφική παράσταση (σχήμα 22), που απεικονίζει την σχέση μεταξύ μνήμης της διαδικασία αυτοπαλίνδρομου υποδείγματος και του μέσου σφάλματος ανακατασκευής, παρατηρούμε την σημαντική διαφοροποίηση του μέσου σφάλματος που παρατηρείται στον μηχανισμό ARCC, καθώς μεταβάλλεται η μνήμη της διαδικασίας VAR. Η καμπύλη παρουσιάζει μια πτωτική τάση, που δεν είναι όμως γνησίως φθίνουσα. Ο PCARCC, παρουσιάζει επίσης πτωτική τροχιά αλλά είναι τέτοιου βαθμού, που κάλλιστα μπορεί να θεωρηθεί αμελητέα. Ο PC3 μένει ανεπηρέαστος.

5.2 Μεταβλητή φάση μάθησης (m)

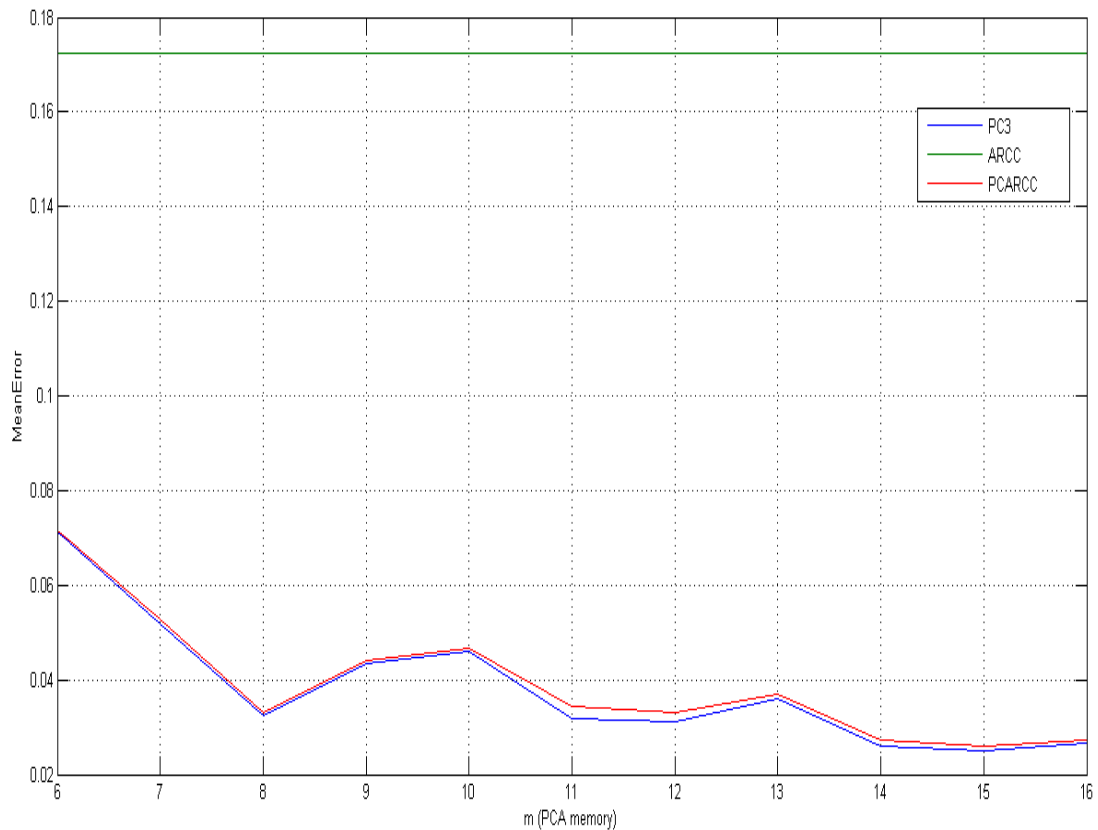
5.2.1 Φάση μάθησης (m) - Ενεργειακό κόστος



Σχήμα 23: Σχέση μεταξύ φάσης μάθησης και ενεργειακού κόστους για τους μηχανισμούς PCARCC(θαλασσί), PC3(πράσινο), SDF (μπλε) και ARCC (κόκκινο)

Στην εικόνα (σχήμα 23) , φαίνεται η σχέση μεταξύ ενεργειακού κόστους που δαπανάται, σε κάθε έναν από τους τέσσερις μηχανισμούς, και μεγέθους της φάσης μάθησης m , που υπάρχει στους μηχανισμούς PC3 και PCARCC. Σε γενικές γραμμές η αύξηση του διαστήματος μάθησης έχει σαν συνέπεια και την αύξηση του ενεργειακού κόστους, στους PC3 και PCARCC, κάτι που είναι φυσικό αφού αυξάνεται ο λόγος $\frac{m}{l}$ και συνεπώς μειώνεται το συνολικό διάστημα της φάσης συμπίεσης. Οι άλλοι δυο μηχανισμοί παραμένουν αμετάβλητοι, αφού δεν επηρεάζονται από την φάση μάθησης.

5.2.2 Φάση μάθησης (m) - Μέσο Σφάλμα ανακατασκευής

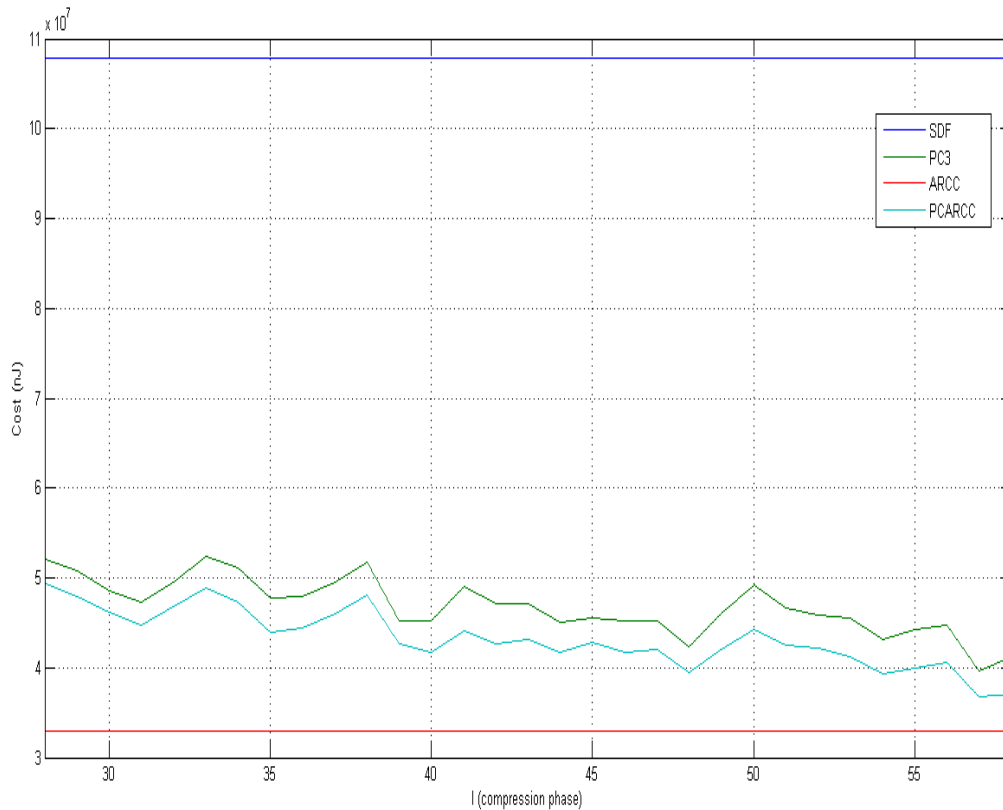


Σχήμα 24: Σχέση μεταξύ φάσης μάθησης m και μέσου σφάλματος ανακατασκευής για τους μηχανισμούς PCARCC(κόκκινο), PC3(μπλε), και ARCC (πράσινο)

Στην γραφική παράσταση (σχήμα 24), απεικονίζεται η σχέση που υπάρχει μεταξύ διάρκειας φάσης μάθησης m και μέσου σφάλματος ανακατασκευής. Όπως είναι αναμενόμενο ο μηχανισμός ARCC μένει ανεπηρέαστος, αφού δεν σχετίζεται με το m . Όσον αφορά τώρα τους δύο άλλους μηχανισμούς PC3 και PCARCC, παρατηρούμε αρχικά, πως και σε αυτή την σχέση η μεταβολή τους ακολουθεί παρόμοια εξέλιξη και σε γενικές γραμμές η αύξηση του διαστήματος μάθησης επιφέρει μια πτώση στο συνολικό μέσο σφάλμα ανακατασκευής. Εξάρτηση η οποία θεωρείται αναμενόμενη, καθώς το μεγαλύτερο διάστημα μάθησης σημαίνει και την αποστολή περισσότερων αυτούσιων πληροφοριών πλαισίου από τον έναν κόμβο στον επόμενο.

5.3 Μεταβλητή φάση συμπίεσης (l)

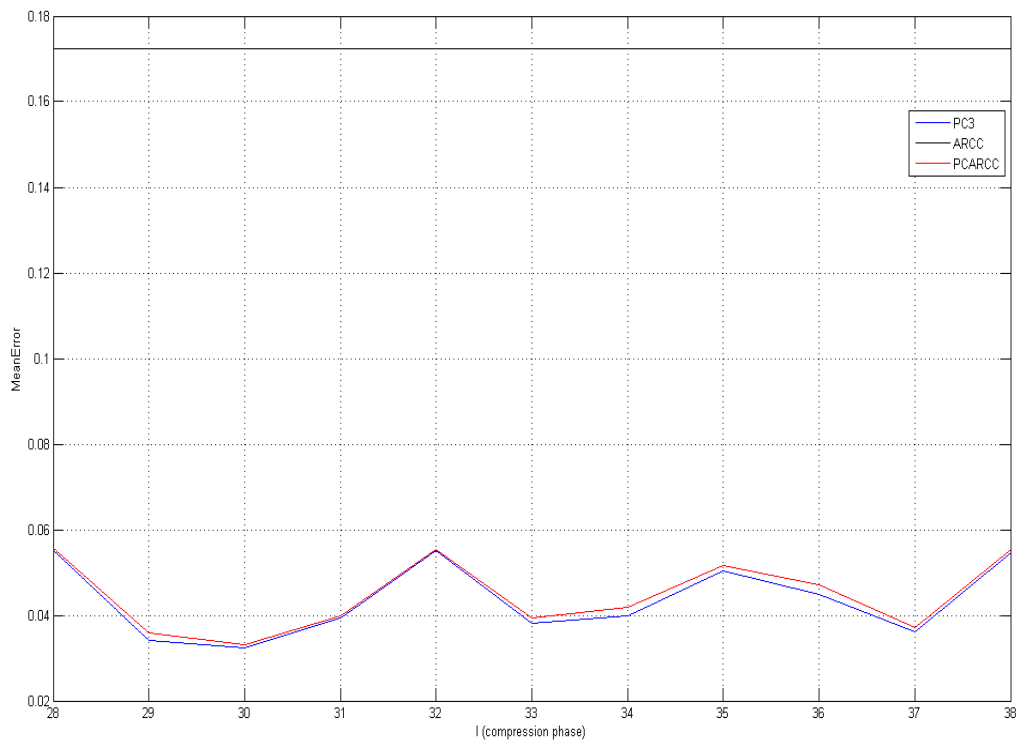
5.3.1 Φάση συμπίεσης (l) - Ενεργειακό κόστος



Σχήμα 25: Σχέση μεταξύ φάσης συμπίεσης και ενεργειακού κόστους για τους μηχανισμούς PCARCC(θαλασσί), PC3(πράσινο), SDF (μπλε) και ARCC (κόκκινο)

Στην εικόνα (σχήμα 25) , φαίνεται η σχέση μεταξύ ενεργειακού κόστους που δαπανάται, σε κάθε έναν από τους τέσσερις μηχανισμούς, και του μεγέθους της φάσης συμπίεσης l , που υπάρχει στους μηχανισμούς PC3 και PCARCC. Σε γενικές γραμμές η αύξηση του διαστήματος συμπίεσης έχει σαν συνέπεια και την μείωση του ενεργειακού κόστους, στους PC3 και PCARCC, κάτι που είναι φυσικό αφού μειώνεται ο λόγος $\frac{m}{l}$ και συνεπώς αυξάνεται το συνολικό διάστημα της φάσης συμπίεσης. Η συμπεριφορά όπως παρατηρούμε είναι ακριβώς η αντίστροφη, από την σχέση φάση μάθησης και ενεργειακού κόστους, που αναλύσαμε πρωτύτερα. Οι άλλοι δυο μηχανισμοί παραμένουν αμετάβλητοι, αφού δεν επηρεάζονται από την φάση συμπίεσης.

5.3.2 Φάση συμπίεσης (l) - Μέσο Σφάλμα ανακατασκευής

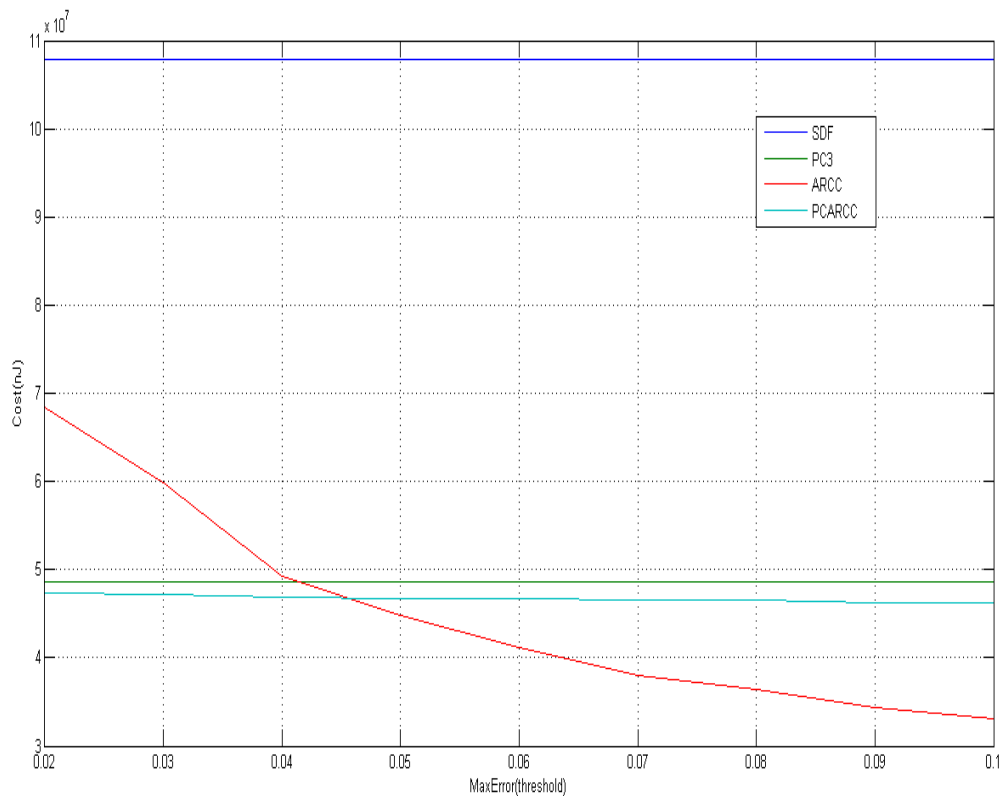


Σχήμα 26: Σχέση μεταξύ φάσης συμπίεσης l και μέσου σφάλματος ανακατασκευής για τους μηχανισμούς PCARCC(κόκκινο), PC3(μπλε), και ARCC (πράσινο)

Στην γραφική παράσταση (σχήμα 26), απεικονίζεται η σχέση που υπάρχει μεταξύ διάρκειας φάσης συμπίεσης l και μέσου σφάλματος ανακατασκευής. Όπως είναι αναμενόμενο ο μηχανισμός ARCC μένει ανεπηρέαστος, αφού δεν σχετίζεται με το l . Τώρα όσον αφορά τους μηχανισμούς PC3 και PCARCC, μεταβάλλονται ανομοιόμορφα καθώς το l αυξάνεται, χωρίς να μπορούμε να εκφράσουμε έναν γενικό κανόνα που να συγκεντρώνει όλες τις επιμέρους περιπτώσεις.

5.4 Μεταβλητό σφάλμα πρόβλεψης

5.4.1 Μέγιστο σφάλμα πρόβλεψης (MAX_{error}) - Ενεργειακό κόστος

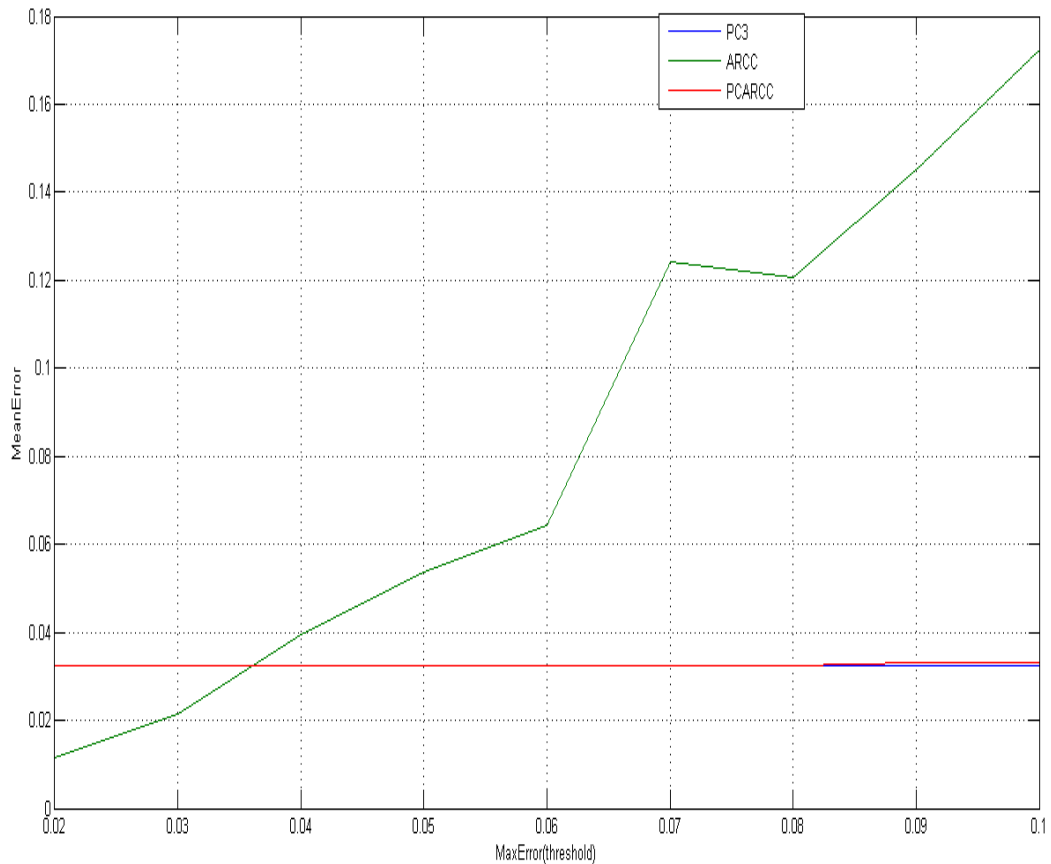


Σχήμα 27: Σχέση μεταξύ μέγιστου σφάλματος πρόβλεψης και ενεργειακού κόστους για τους μηχανισμούς PCARCC(θαλασσί), PC3(πράσινο), SDF (μπλε) και ARCC (κόκκινο)

Στην γραφική παράσταση (σχήμα 27), απεικονίζεται η σχέση μεταξύ μέγιστου σφάλματος πρόβλεψης, όπως αυτό τίθεται στις ARCC και PCARCC, και ενεργειακού κόστους. Διακρίνουμε πως καθώς αυξάνουμε το μέγιστο επιτρεπόμενο σφάλμα πρόβλεψης, έχουμε μείωση της συνολικά καταναλισκόμενης ενέργειας, που απαιτείται για την ολοκλήρωση της διαδικασίας στους μηχανισμούς, ARCC και PCARCC. Αυτό είναι αναμενόμενο, καθώς ένα πιο «αυστηρό» όριο έχει σαν αποτέλεσμα, να αυξηθούν οι φορές κατά τις οποίες η συνθήκη απόφασης δηλ. $e_{i-i}(t) < Max_{error}$, στον κόμβο αποστολής, δεν ικανοποιείται, αυτό έχει σαν επακόλουθο την αποστολή ολοκλήρου του διανύσματος πλαισίου, συνεπώς αύξηση στο ενεργειακό κόστος του μηχανισμού, άρα μικρότερη διάρκεια ζωής του κόμβου.

Η αύξηση αυτή συμβαίνει σε πολύ μεγαλύτερο βαθμό στην ARCC από ότι στην PCARCC. Στους μηχανισμούς, PC3 και SDF, οι οποίοι δεν κάνουν χρήση της VAR, όπως είναι φυσικό το κόστος παραμένει σταθερό ανεξαρτήτου του μεταβαλλόμενου μεγέθους, υπάρχουν στην παράσταση για λόγους σύγκρισης των μεγεθών.

5.4.2 Μέγιστο σφάλμα πρόβλεψης (MAX_{error}) - Μέσο Σφάλμα ανακατασκευής

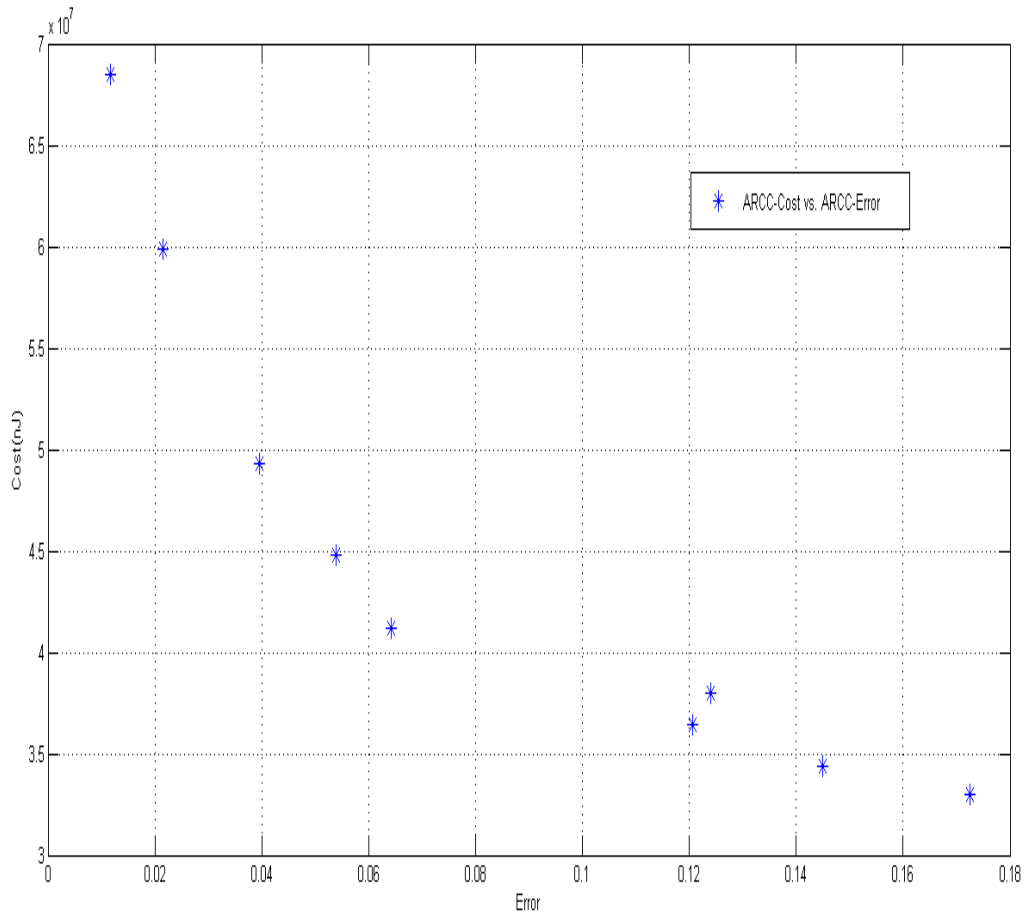


Σχήμα 28: Σχέση μεταξύ μέγιστου σφάλματος πρόβλεψης και μέσου σφάλματος ανακατασκευής για τους μηχανισμούς PCARCC(κόκκινο), PC3(μπλε), και ARCC (πράσινο)

Στην γραφική παράσταση (σχήμα 28), που απεικονίζει την σχέση μεταξύ μέγιστου σφάλματος πρόβλεψης και μέσου σφάλματος ανακατασκευής, παρατηρούμε την σημαντική διαφοροποίηση του μέσου σφάλματος που παρατηρείται στον μηχανισμό ARCC, καθώς μεταβάλλεται το μέγιστο επιτρεπτό σφάλμα πρόβλεψης. Η καμπύλη παρουσιάζει μια αυξητική τάση, σε γενικές γραμμές, υποδεικνύοντας πως ένα πιο «χαλαρό» όριο για την πρόβλεψη, που γίνεται στον κόμβο προώθησης, συνεπάγεται και μεγαλύτερο σφάλμα κατά την αναπαραγωγή του πλαισίου από τον κόμβο λήψης. Συνεπώς ελαττώνεται η ακρίβεια των δεδομένων. Ο PCARCC, παρουσιάζει επίσης αυξητική τροχιά αλλά είναι τέτοιου βαθμού, που κάλλιστα μπορεί να θεωρηθεί αμελητέα. Ο PC3 μένει ανεπηρέαστος.

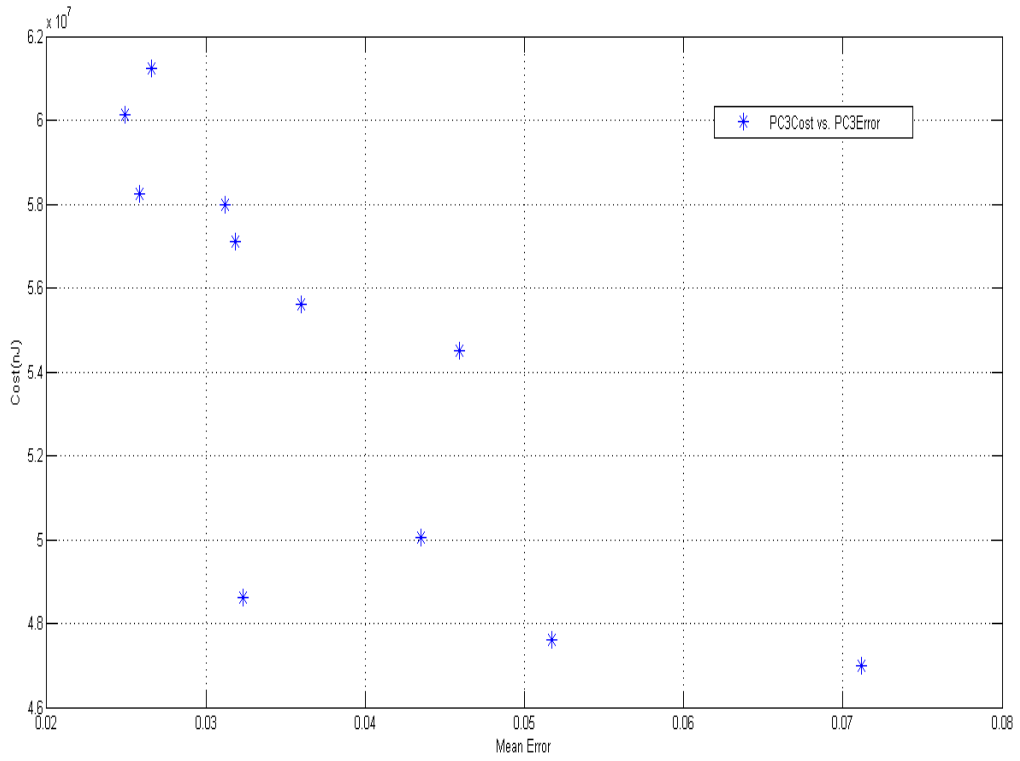
5.5 Ενεργειακό Κόστος – Σφάλμα ανακατασκευής

Τέλος παρουσιάζει ενδιαφέρον, να δούμε πως απεικονίζεται γραφικά η σχέση ανάμεσα στο ενεργειακό κόστος που καταναλίσκεται και στο σφάλμα ανακατασκευής που παρουσιάζεται κατά την διαδικασία προώθησης των διανυσμάτων πλαισίου, όταν εφαρμόζουμε έναν από τους παραπάνω μηχανισμούς.

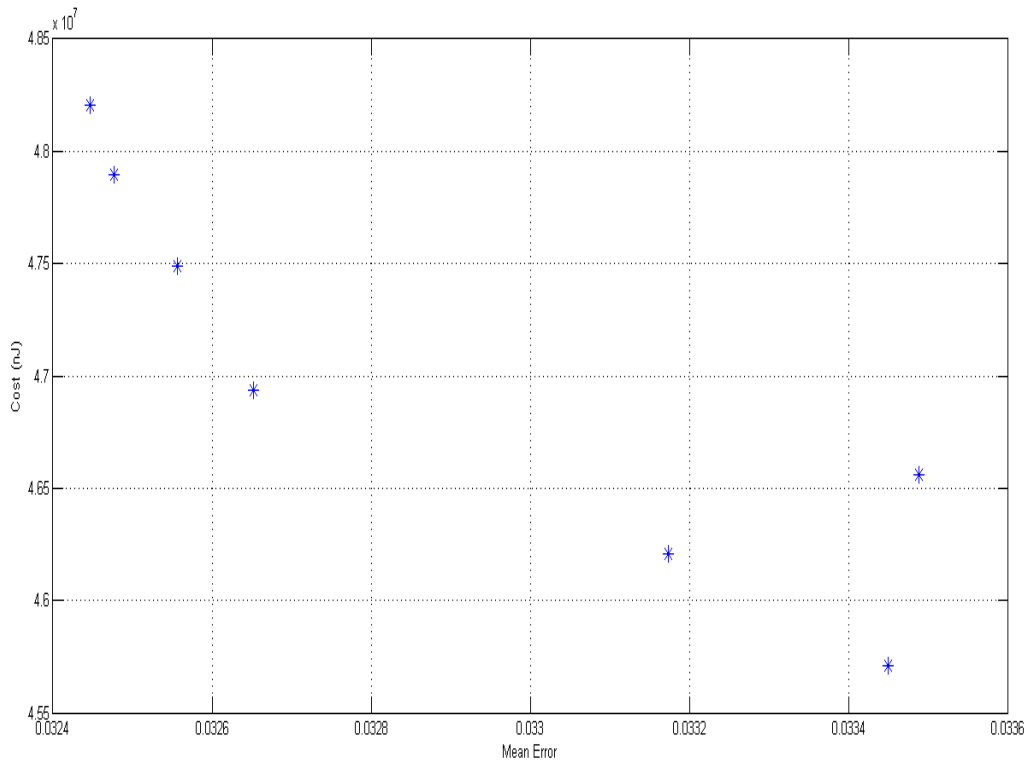


Σχήμα 29: Σχέση εξάρτησης ανάμεσα στο κόστος και στο σφάλμα ανακατασκευής στον ARCC

Στο σχήμα 29, φαίνεται μια τέτοια απεικόνιση. Στην παρούσα παράσταση απεικονίζουμε την σχέση κόστους-σφάλματος που υπάρχει στην διαδικασία ARCC για εκτελέσεις με διαφορετικές τιμές των βασικών μεταβλητών εισόδου (p, m, l, MAX_{error}) που έχουν σαν αποτέλεσμα επίσης διαφορετικές τιμές στο συνολικό κόστος και στο σφάλμα ανακατασκευής. Κάθε αστερίσκος στην εικόνα, αντιπροσωπεύει και μια εκτέλεση με τις διαφορετικές τιμές των βασικών μεταβλητών εισόδου. Είναι εμφανές λοιπόν, πως η επιδίωξη μας για μικρότερη κατανάλωση ενέργειας οδηγεί σε ολοένα και μεγαλύτερα σφάλματα ανακατασκευής, δηλ. μειώνεται η ακρίβεια των δεδομένων που υπάρχει στο ασύρματο δίκτυο αισθητήρων. Αυτή η αντίστροφη συσχέτιση είναι πολύ έντονη όσον αφορά την ARCC. Σε παρόμοια αποτελέσματα καταλήγουμε και από τις εκτελέσεις των άλλων μηχανισμών, ωστόσο όπως βλέπουμε και στις παρακάτω εικόνες, στον PC3, δεν είναι τόσο αυστηρή αυτή η σχέση όσο στον ARCC.



Σχήμα 30: Σχέση εξάρτησης ανάμεσα στο κόστος και στο σφάλμα ανακατασκευής στον PC3



Σχήμα 31: Σχέση εξάρτησης ανάμεσα στο κόστος και στο σφάλμα ανακατασκευής στον PCARCC

6 ΣΥΜΠΕΡΑΣΜΑΤΑ

Σε αυτή την εργασία προτείναμε και υλοποιήσαμε δυο νέους μηχανισμούς συμπύεσης και προώθησης πληροφορίας πλαισίου σε ασύρματα δίκτυα αισθητήρων. Απώτερος σκοπός των δυο αυτών μηχανισμών, είναι η ελαχιστοποίηση της καταναλισκόμενης ενέργειας, κατά την διαδικασία προώθησης, διατηρώντας όμως ταυτόχρονα σε αποδεκτά επίπεδα την ακρίβεια των δεδομένων ανάμεσα στους κόμβους αυτού του δικτύου. Οι δυο αυτοί μηχανισμοί βασίστηκαν, ο ένας (ARCC) εξολοκλήρου και ο άλλος (PCARCC) εν μέρει στο διανυσματικό αυτοπαλίνδρομο υπόδειγμα VAR και εκμεταλλεύονται την πολυμεταβλητή φύση και την στατιστική εξάρτηση, που χαρακτηρίζει την πληροφορία πλαισίου σε πραγματικές εφαρμογές ασύρματων δικτύων αισθητήρων. Μέσω αυτού του μηχανισμού δίνεται η δυνατότητα σε κάθε κόμβο να «προβλέψει» το επόμενο πλαίσιο, το οποίο επρόκειτο να λάβει από τον προηγούμενο κόμβο, χωρίς στη πραγματικότητα να το λάβει ποτέ, άρα εξοικονομείται η ενέργεια που θα σπαταλούταν για την αποστολή και τη λήψη αυτού του πλαισίου. Ωστόσο αυτή η διαδικασία προώθησης, όπως είναι αναμενόμενο προκαλεί κάποια απόκλιση στις τιμές των δεδομένων που έχει στην κατοχή του ο κάθε κόμβος σε σχέση με τα πραγματικά δεδομένα, αυτή η απόκλιση επιθυμούμε να είναι όσο το δυνατό μικρότερη.

Στη συνέχεια προχωρήσαμε στην εκτίμηση της απόδοσης, που επιτυγχάνουν αυτοί οι δυο νέοι μηχανισμοί, μέσω προσομοίωσης της διαδικασίας προώθησης με πραγματικά διανύσματα πλαισίου. Συγκρίναμε την απόδοση αυτών, με τον μηχανισμό της συμβατικής προώθησης όλων των δεδομένων πλαισίου SDF, καθώς επίσης και με έναν ήδη υπάρχον μηχανισμό που βασίζεται στη ανάλυση κυρίων συνιστωσών τον PC3, την λειτουργία του όποιου περιγράψαμε συνοπτικά, καθώς αποτελεί μέρος της λειτουργίας του PCARCC.

Αποδείξαμε λοιπόν, καταρχήν, πως οι δυο αυτοί μηχανισμοί εκπληρώνουν τον πρωταρχικό τους στόχο και παρατείνουν την διάρκεια ζωής του δικτύου. Πιο συγκεκριμένα, ο ARCC επιτυγχάνει αξιοσημείωτη μείωση του απαιτούμενου ενεργειακού κόστους, όχι μόνο σε σχέση με τον SDF (72%) αλλά και με τον PC3 (43%). Ωστόσο, επιφέρει ένα σχετικά μεγάλο σφάλμα ανακατασκευής, κατά πολύ μεγαλύτερο από τον PC3. Ο PCARCC, σύνθετος μηχανισμός συναποτελούμενος από τον PC3 και τον ARCC, επιτυγχάνει παρόμοιο σφάλμα ανακατασκευής με τον PC3, έχοντας (έστω και ελάχιστα) καλύτερη ενεργειακή απόδοση. Προκειμένου, να έχουμε μια περισσότερο σφαιρική εκτίμηση της απόδοσης των μηχανισμών, ορίσαμε δυο συντελεστές, που συνδυάζουν το όφελος από τη μείωση του κόστους μαζί με τη ζημία από το προκύπτον σφάλμα. Τα αποτελέσματα δείχνουν πως από τους μηχανισμούς συμπύεσης, ο ARCC έχει τον μεγαλύτερο συντελεστή και συνεπώς την καλύτερη συνολική απόδοση, μετά ακολουθεί ο PCARCC και έπειτα ο PC3.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός Όρος
Autoregression-based Context Compression	Συμπίεση Πλαισίου βασισμένη στο Αυτοπαλίνδρομο Υπόδειγμα
Context	Πλαίσιο
Context Aware-System	Σύστημα επίγνωσης πλαισίου
Contextual information	Πληροφορία πλαισίου
Data flow	Ροή πληροφορίας
Error threshold	Μέγιστο όριο σφάλματος
Pervasive computing	Διάχυτος υπολογισμός
Principal Component-Autoregression-based Context Compression	Συμπίεση Πλαισίου βασισμένη στην Ανάλυση Κύριων Συνιστωσών και στο Αυτοπαλίνδρομο υπόδειγμα
Principal Components-based Context Compression	Συμπίεση Πλαισίου βασισμένη στην Ανάλυση Κύριων Συνιστωσών
Relay node	Κόμβος Μεταφοράς
Sensor node	Κόμβος ανίχνευσης
Simple Data Forwarding	Μηχανισμός Απλής Προώθηση Δεδομένων
Sink node	Κόμβος συλλέκτης
Upstream node	Επόμενος κόμβος
Vector Autoregression Model	Υπόδειγμα Αυτοπαλίνδρομου Διανύσματος

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

ARCC	Autoregression-based Context Compression
CAS	Context-Aware Systems
PC3	Principal Components-based Context Compression
PCARCC	Principal Component-Autoregression-based Context Compression
SDF	Simple Data Forwarding
VAR	Vector Autoregressive Model
WSN	Wireless Sensor Networks

ΠΑΡΑΡΤΗΜΑ Ι

Το ARfit είναι μια συλλογή από συναρτήσεις Matlab, με στόχο την μοντελοποίηση και ανάλυση πολυμεταβλητών χρονοσειρών κάνοντας χρήση αυτοπαλίνδρομων (autoregressive - AR) μοντέλων [8]. Το ARfit περιλαμβάνει συναρτήσεις για εφαρμογή αυτοπαλινδρομούμενων μοντέλων σε συγκεκριμένες χρονοσειρές δεδομένων, για την ανάλυση ιδιομορφών ενός εφαρμοσμένου μοντέλου, και για την προσομοίωση διαδικασιών αυτοπαλινδρόμησης.

Το ARfit υπολογίζει τις παραμέτρους των αυτοπαλινδρομούμενων μοντέλων από συγκεκριμένες χρονοσειρές με έναν κλιμακωτό αλγόριθμο ελαχίστων τετραγώνων, ο οποίος είναι υπολογιστικά αποδοτικός, ιδίως όταν τα δεδομένα είναι μεγάλων διαστάσεων.

Οι συναρτήσεις του ARfit κατασκευάζουν, κατά προσέγγιση, σταθερά χρονικά διαστήματα για τις εκτιμώμενες παραμέτρους και υπολογίζουν στατιστικά στοιχεία με τα οποία η επάρκεια ενός εφαρμοσμένου μοντέλου μπορεί να αξιολογηθεί. Τα δυναμικά χαρακτηριστικά μιας μοντελοποιημένης χρονοσειράς μπορεί να εξεταστούν, μέσω της ανάλυσης της , σε ιδιόμορφες και τις σχετιζόμενες περιόδους ταλάντωσης, χρόνους απόσβεσης και διεγέρσεις.

ΓΕΝΙΚΑ

Το στοχαστικό υπόδειγμα, που «κρύβεται πίσω» από τις συναρτήσεις της Arfit, είναι το m – μεταβλητό αυτοπαλίνδρομο υπόδειγμα, με παράγων υστέρησης ή τάξης p .

$$u_v = w + \sum_{l=1}^p A_l u_{v-l} + \varepsilon_v, \quad \varepsilon_v = \text{noise}(C)$$

Ένα υπόδειγμα για μια στατική χρονοσειρά, αποτελούμενη από διανύσματα καταστάσεων u_v , m – διαστάσεων, τα οποία έχουν παρατηρηθεί σε ισαπέχουσες στιγμές v .

Οι πίνακες $A_1, \dots, A_p \in R^{m \times m}$, αποτελούν τους πίνακες συντελεστών του αυτοπαλίνδρομου μοντέλου, και τα m – διαστάσεων διανύσματα $\varepsilon_v = \text{noise}(C)$ είναι ασυσχέιστα τυχαία διανύσματα με μέση τιμή μηδέν και πίνακα συνδιασποράς $C \in R^{m \times m}$.

Το m - διαστάσεων διάνυσμα w , είναι ένα διάνυσμα που καθιστά δυνατό, την μη-μηδενική μέση τιμή της χρονοσειράς.

Το ARfit περιλαμβάνει συναρτήσεις (i) για τον υπολογισμό, από έναν δείγμα χρονοσειρών u_v , της τάξης p , του διανύσματος w , των πινάκων συντελεστών A_1, \dots, A_p και του πίνακα συνδιασποράς C , (ii) εκτιμάται η επάρκεια του εφαρμοζόμενου αυτοπαλίνδρομου μοντέλου, (iii) αναλύεται σε ιδιομορφές και (iv) προσομοιώνεται η διαδικασία αυτοπαλινδρόμησης.

ΥΠΟΛΟΓΙΣΜΟΣ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΣΕ ΕΝΑ ΑΥΤΟΠΑΛΙΝΔΡΟΜΟ ΜΟΝΤΕΛΟ

Για την επιλογή της τάξης p , ενός αυτοπαλίνδρομου μοντέλου και τον υπολογισμό των παραμέτρων w , A_1, \dots, A_p και C , ο κλιμακωτός αλγόριθμος ελαχίστων τετραγώνων που περιγράφεται από τους Neumaier και Schneider [9], υλοποιείται στο ARfit.

Λαμβάνοντας σαν είσοδο μια χρονοσειρά, αποτελούμενη από $N+p$ διανύσματα κατάστασης u_v , ($v = 1-p, \dots, N$) και ενός κάτω p_{\min} και άνω p_{\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 $p_{\min}, \dots, p_{\max}$ και υπολογίζει τις παραμέτρους w , A_1, \dots, A_p και C για ένα μοντέλο βέλτιστης τάξης p_{opt} .

Σαν βέλτιστη τάξη p_{opt} , ενός αυτοπαλίνδρομου μοντέλου επιλέγεται αυτή που βελτιστοποιεί έναν κριτήριο επιλογής τάξης [10]. Τα κριτήρια επιλογής τάξης που υλοποιούνται στο ARfit, είναι: κριτήριο τελευταίου λάθους πρόβλεψης (Final Prediction Error – FPE), [11] και Schwarz's - Bayesian κριτήριο (SBC) [12]. Ο Lutkepohl [13] 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.

Στον κλιμακωτό αλγόριθμο ελαχίστων τετραγώνων, τα κριτήρια επιλογής τάξης αξιολογούνται για μοντέλα τάξης $p_{\min}, \dots, p_{\max}$, μέσω της κλιμακωτής αφαίρεσης της πρώτης σειράς ενός κανονικοποιημένου, QR παραγοντοποιημένου πίνακα δεδομένων για ένα μοντέλο μέγιστης τάξης p_{\max} . Από την παραγοντοποίηση QR, για ένα μοντέλο τάξης p_{\max} , υπολογίζονται κατά προσέγγιση, από την μέθοδο των ελαχίστων τετραγώνων, οι παράμετροι w , A_1, \dots, A_p και C για το μοντέλο τάξης p_{opt} , η οποία τάξη, βελτιστοποιεί το κριτήριο επιλογής τάξης.

Ο κλιμακωτός υπολογισμός ελαχίστων τετραγώνων είναι υπολογιστικά αποδοτικός, κυρίως όταν οι χρονοσειρές u_v είναι μεγάλων διαστάσεων.

ΥΠΟΛΟΓΙΣΤΙΚΗ ΠΟΛΥΠΛΟΚΟΤΗΤΑ ΤΟΥ ΚΛΙΜΑΚΩΤΟΥ ΑΛΓΟΡΙΘΜΟΥ ΕΛΑΧΙΣΤΩΝ ΤΕΤΡΑΓΩΝΩΝ

Ο πίνακας των δεδομένων του οποίου η QR παραγοντοποίηση πρόκειται να υπολογιστεί είναι μεγέθους $N'x(n_{p_{\max}} + m)$, όπου ο αριθμός των σειρών N' του πίνακα είναι ίσος με το μέγεθος του δείγματος N , αν οι εκτιμήσεις των ελαχίστων τετραγώνων δεν είναι κανονικοποιημένες, ή ίσος με $N + n_{p_{\max}} + m$, εάν οι εκτιμήσεις ελαχίστων τετραγώνων είναι κανονικοποιημένες, υπολογίζοντας την παραγοντοποίηση QR ενός προσαυξημένου πίνακα δεδομένων. Υπολογίζοντας την QR παραγοντοποίηση χρειάζονται, $O(N' p_{\max}^2 m^2)$ πράξεις, m ο αριθμός των μεταβλητών της χρονοσειράς [9].

Στους παραδοσιακούς αλγορίθμους για την εκτίμηση των παραμέτρων ενός αυτοπαλίνδρομου μοντέλου, μια ξεχωριστή παραγοντοποίηση θα έπρεπε να είχε υπολογισθεί για κάθε τάξη p , για την οποία έναν κριτήριο επιλογής τάξης θα έπρεπε να υπολογιστεί.

Στον κλιμακωτό αλγόριθμο ελαχίστων τετραγώνων, οι κλιμακωτές αφαιρέσεις της πρώτης σειράς, που εφαρμόζονται, απαιτούν $O(m^3)$ πράξεις, για κάθε τάξη p , για την οποία έναν κριτήριο επιλογής τάξης θα έπρεπε να υπολογισθεί.

Από την στιγμή που $N' \geq m$, η κλιμακωτή αφαίρεση της πρώτης σειράς για κάθε τάξη p απαιτεί λιγότερες πράξεις από μια νέα QR παραγοντοποίηση. Εάν ο αριθμός των σειρών N' , των πινάκων δεδομένων, των οποίων η QR παραγοντοποίηση πρέπει να υπολογιστεί, είναι κατά πολύ μεγαλύτερη από την διάσταση m , των διανυσμάτων κατάστασης, ο αριθμός των πράξεων που απαιτούνται για τη κλιμακωτή αφαίρεση της πρώτης σειράς των πινάκων γίνεται αμελητέος σε σύγκριση με τον αριθμό των πράξεων που απαιτούνται για την QR παραγοντοποίηση.

Με τον κλιμακωτό αλγόριθμο ελαχίστων τετραγώνων, συνεπώς, η τάξη και οι παράμετροι ενός αυτοπαλίνδρομου μοντέλου μπορούν να υπολογιστούν περίπου $(p_{\max} - p_{\min} + 1)$ -φορές ταχύτερα από ότι με τους παραδοσιακούς αλγορίθμους ελαχίστων τετραγώνων, που απαιτούν $(p_{\max} - p_{\min} + 1)$ ξεχωριστές QR παραγοντοποιήσεις.

Δεδομένου ότι η διαγραφή στηλών ενός πίνακα δεν μειώνει την μικρότερη ιδιάζουσα τιμή του πίνακα, ο κλιμακωτός αλγόριθμος ελαχίστων τετραγώνων είναι μια αριθμητικά σταθερή διαδικασία [14].

ΠΑΡΑΡΤΗΜΑ II

Τα πρώτα από τα 387 πραγματικά διανύσματα πλαισίου που χρησιμοποιήθηκαν σε όλες τις πειραματικές διαδικασίες. Κάθε εγγραφή είναι και ένα πλαίσιο που προωθείται σε συγκεκριμένη χρονική στιγμή t .

temp1	hum1	temp2	hum2	temp3	hum3	wind
14,4	64,2	17,4	55,5	15	65,8	0,33
15,2	59,8	20,7	44,9	15,2	61,7	0,42
15,7	56,2	17,9	50,7	15,3	63,2	0,6
15,8	61	16,3	53,5	15,5	59,2	1,04
15,5	56,1	16,3	53,5	16,3	55	0,45
15,5	55,2	18,9	46,1	15,8	50,9	0,77
15,6	56,8	16,6	50,3	15,5	52,7	0,43
15,3	60	16	55,8	14,8	61,5	0,57
15,2	61,3	15,2	62,3	14,8	61,5	0,36
15,9	55,3	14,9	58,1	14,1	62,1	0,03
13,8	65,6	14,9	58,1	14,1	62,1	0,01
12,9	70,9	13,5	69,6	12,3	76	0,01
12,1	74,6	12,4	73,5	11,2	80,9	0,04
11,4	76,1	11,7	75,8	10,7	79,7	0,12
11,3	76,8	11,9	73,6	10,8	80,5	0,4
10,8	81	11,2	78,3	10,2	84	0,18
10,5	83,3	11	81,4	9,9	85,8	0,1
9,9	84,6	10,2	83,8	10	83,8	0,25
10,6	83,9	10,2	83,8	10,4	82,6	0,17
10,7	83,3	10,8	83,4	10,5	83,6	0,39
10,6	86	10,9	84	10,5	85,5	0,49
10,7	85,6	11	83,8	10,5	85,5	0,33
10,6	84,6	10,7	83,9	10,5	85,5	0,04
10,5	84,1	10,6	83,9	10	86,1	0,5
10,5	83,9	10,5	84,5	10,2	85,1	0,49
10,6	82,6	10,5	83,4	9,9	85	0,34
10,2	80,9	9,9	82,9	10	81,5	0,62
10,4	78,4	10,1	81,5	10	79,6	0,43
10,4	77,8	10,4	78,8	9,9	80,6	0,57
10,4	77,3	10,4	79,1	10,1	79,2	0,65
10,4	79,4	10,2	80,1	10,2	80,1	0,27
10,1	81,2	10,2	80,1	10	81,5	0,34
10,2	82	9,9	83,7	9,7	83,3	0,32
10,5	80,4	10,3	82,1	10,2	81,2	0,32
10,5	79,7	10,4	81,9	10,4	81,3	0,71

ΠΑΡΑΡΤΗΜΑ ΙΙΙ

Ο κώδικας των υλοποιημένων συναρτήσεων (PCARCC.m, ARCC.m, predict.m) στο Matlab, με βάση τις οποίες έγινε η προσομοίωση των μηχανισμών που προτείναμε-περιγράψαμε στην εργασία, καθώς και το εκτελέσιμο αρχείο που κάνει χρήση αυτών (exe.m)

PCARCC.m

```
% Ylopoihs mhxaniasmwn PC3 kai PCARCC
% Ekteleitai sta CVs pou emperiexontai sto Y
% m & l einai antistoixa to mikos twn diasthmatwn ma8isis kai
simpieshs tis diadikasias PC3
% VarMemoryLength & VarMaxError einai antistoixa to diasthma
ma8isis kai to megisto sflama provlepis tis ARCC
% Epistrefei ta dianismata pou deixnoun tin exelikh toy kostoys
stous mixanismoys PC3 (xCostPC3), SDF(xCostSDF) kai PCARCC
(xCostPCARCC)
% Ka8ws epishs kai to sfalma anakataskeyhs poy parousiazetai
stous PC3(PC3totErrPlot)kai PCARCC (PCARCCtotErrPlot)

function[xCostPC3,xCostSDF,xCostPCARCC,PC3totErrPlot,PCARCCtotEr
rPlot ] = PCARCC (Y, m , l, VarMemoryLength, VarMaxError)
traceMatrix=Y;
xCostPC3 = [];
xCostSDF = [];
costNo = 0;
flag=1;
xCostPCARCC=[];
costVar=0;
pcarccReceiveNode = []; % sto pinaka pcarccReceiveNode einai ta
deigmata poy exei sthn katoxh toy o ReceiveNode meta to telos
toy mixanismoys

[traceLength n] = size(traceMatrix);
p=0.9;% PC3
mpointer=1;

for pp=1:1:traceLength
    PCARCCtotErrPlot (pp)=0;
```

```

    PC3totErrPlot (pp)=0;
    sendDataPlot (pp)=0;
end
lend=0;
cost=0; %COST
costOfBitTrans=720; %COST (nJoule)
costOfBitRec=110; %COST (nJoule)
costOfInstruction=4; %COST (nJoule)
while (lend==0)
    mpointer;
    %Stelnei -m*n- times opws akriwvs einai (h diadikasia einai
    paromoia kai stouw dio mixanismsous)
    for j=1:1:m
        H(j,:) = traceMatrix(mpointer+j-1,:); %Kataskevi History
        matrix gia PCA

        pcarccReceiveNode = [pcarccReceiveNode
        ;traceMatrix(mpointer+j-1,:)];
        cost = cost + transmitValuesCost(n, costOfBitTrans);
        %COST of PC3 (send-node)
        cost = cost + receiveValuesCost(n, costOfBitRec); %COST
        of PC3 (receive-node)

        %*****
        %*****

        costVar = costVar + transmitValuesCost(n,
        costOfBitTrans); %COST of PCARCC (send-node) *****
        costVar = costVar + receiveValuesCost(n, costOfBitRec);
        %COST of PCARCC (receive-node) *****

        %*****
        %*****

        costNo = costNo + transmitValuesCost(n, costOfBitTrans);
        %COST of SDF (send-node)
        costNo = costNo + receiveValuesCost(n, costOfBitRec);
        %COST of SDF (receive-node)
        %my statistics
        xCostPC3 = [xCostPC3 cost];
        xCostSDF = [xCostSDF costNo];
        xCostPCARCC = [xCostPCARCC costVar];
        %*****
    end
end

```

```

    %mpointer+j-1
end

    %KAI pompos KAI dektis xrisimopoioun tis prohgoumenes -m-
metriseis ws memory gia PCA.

    %Kanoun PCA kai kratan tis prwtes -q- PCs wste h
metavlitotita na einai -p%-

    [coeff, meanH, q] = InHistorymatrixOutPcacoef2 (H,p);
    %kostos ipologismou twv PCs
    %send-node
    cost = cost + createPcaCoeffCost(n,m,costOfInstruction);
    %COST of PC3
    %receive-node
    cost = cost + createPcaCoeffCost(n,m,costOfInstruction);
    %COST of PC3

    %kostos ipologismou twv PCs
    %send-
node*****
*****
    costVar = costVar +
createPcaCoeffCost(n,m,costOfInstruction); %COST of PCARCC
    %receive-
node*****
*****
    costVar = costVar +
createPcaCoeffCost(n,m,costOfInstruction); %COST of PCARCC

%*****
*****

    %Oles oi metriseis tou traceMatrix apo -mpointer+m- ws -
mpointer+m+1-1- tha ipostoun
    %PCA, me vasi ton -coeff-. Etsi, kathe mia apo aftes tis
metriseis tha exei pleon -q- PCs.

    %Oles aftes oi PCA times tha apothikeftoun ston pinaka -
lPCAMeasurements-.

    lPCAMeasurements = [
(traceMatrix((mpointer+m):(mpointer+m+1-1),:))
 repmat(meanH,[1,1]))*coeff(:,1:q) ];

```

```

% O pinakas lPCAMEauserements me ta simpiesmena CVs
diastasews q, 8a

% perasei apo ton mixanismo ARCC, oson afora ton mixanismo
PCARCC

% etsi 8a exoume ta diastimata ypo-ma8isisis kai ypo-
simpieshs opws

% ayta orizontai stin 8ewria

[onebit_counter,fullcv_counter,
error_i_j,C_VAR,limit,RecNodeSamples]=ARCC(lPCAMEauserements(:,1
:q),VarMemoryLength,VarMaxError);

%Ypologismos twn errors gia tis metriseis sto diastima m ews
m+1, oson

%afora ton mixanismo PC3

lPcaMinusRealMeauserementsNorm = sqrt(sum( ( (
lPCAMEauserements(:, 1:q)*coeff' + repmat(meanH,[1,1]) ) -
traceMatrix((mpointer+m):(mpointer+m+1-1),:).^2 ,2) ));

lRealMeauserementsNorm = sqrt(sum(
traceMatrix((mpointer+m):(mpointer+m+1-1),:).^2 ,2) );

for zd=1:1:1 %Prostasia, gia na mhn ginei diairesis me to
mhden

if (lRealMeauserementsNorm(zd)==0)
lRealMeauserementsNorm(zd)=0.0000000000000001;
end
end

lPCAMEauserementsError = lPcaMinusRealMeauserementsNorm ./
lRealMeauserementsNorm;

RecNode=(RecNodeSamples*coeff' + repmat(meanH,[1,1]));
pcarccReceiveNode=[pcarccReceiveNode ; RecNode];

%
*****
*****

%Ypologismos twn errors gia tis metriseis sto diastima m
ews m+1 oson afora to mixanismo PCARCC

%apo m ews m+limit (ypo-ma8isis)exoume idio sfalma me to
PC3, enw apo m+limit ews l (ypo-simpiesi)

%exartatai apo to an stelnetai onebit 'h fullcv ston
receive-Node(RecNodeSamples)

```

```

lVarMinusRealMeasuserementsNorm = sqrt(sum( ( (
RecNodeSamples*coeff' + repmat(meanH,[1,1]) ) -
traceMatrix((mpointer+m):(mpointer+m+1-1),:) ).^2 ,2) );

lRealMeasuserementsNorm = sqrt(sum(
traceMatrix((mpointer+m):(mpointer+m+1-1),:) .^2 ,2) );

for zd=1:1:1 %Prostasia, gia na mhn ginei diairesh me to
mhden

    if (lRealMeasuserementsNorm(zd)==0)
        lRealMeasuserementsNorm(zd)=0.000000000000000001;
    end

end

lVARMeasuserementsError= lVarMinusRealMeasuserementsNorm./
lRealMeasuserementsNorm;

%
*****
*****

for j=1:1:1
    PC3totErrPlot(mpointer+m+j-1)=lPCAMeasuserementsError(j);
    PCARCCtotErrPlot(mpointer+m+j-
1)=lVARMeasuserementsError(j);

    cost = cost + PCsCompressCost(n, q, costOfInstruction);
%COST of PC3 (compression)

    cost = cost + transmitValuesCost(q, costOfBitTrans);
%COST of PC3 (transmission)

    cost = cost + receiveValuesCost(q, costOfBitRec); %COST
of PC3 (reception)

    cost = cost + PCsUncompressCost(n, q, costOfInstruction);
%COST of PC3 (uncompression)

    costNo = costNo + transmitValuesCost(n, costOfBitTrans);
%COST of SDF

    costNo = costNo + receiveValuesCost(n, costOfBitRec);
%COST of SDF

    xCostPC3 = [xCostPC3 cost];
    xCostSDF = [xCostSDF costNo];

end

%Sto xroniko diasthma apo m ews m+1 , to kostos tou PCARCC
exartatai apo ta posa flag-bits kai

%autousia CVs stal8uikan, sinepws

```



```

    for i=1:onebit_counter %kostos apo m+limit ews l (l2) otan
aposteletai flag
        costVar      =      costVar      +      PCsCompressCost(n,      q,
costOfInstruction); %COST of PCARCC
        costVar      =      costVar      +      PCsUncompressCost(n,      q,
costOfInstruction); %COST of PCARCC
        costVar      =      costVar      +      transmitValuesCost((1/4),
costOfBitTrans);%COST of PCARCC
        costVar = costVar + receiveValuesCost((1/4), costOfBitRec);
%COST of PCARCC
        costVar = costVar + costOfInstruction * limit^2; %COST of
PCARCC (VAR cost calculation (send-node))
        costVar = costVar + costOfInstruction * limit^2; %COST of
PCARCC (VAR cost calculation (receive-node))
        xCostPCARCC = [xCostPCARCC costVar];
    end

    for i=1:fullcv_counter %kostos apo m+limit ews l (l2), otan
aposteletai olo to CV aytousio
        costVar      =      costVar      +      PCsCompressCost(n,      q,
costOfInstruction); %COST of PCARCC
        costVar      =      costVar      +      PCsUncompressCost(n,      q,
costOfInstruction); %COST of PCARCC
        costVar      =      costVar      +      transmitValuesCost(q,
costOfBitTrans);%COST of PCARCC
        costVar = costVar + receiveValuesCost(q, costOfBitRec);
%COST of PCARCC
        costVar = costVar + costOfInstruction * limit^2; %COST of
PCARCC (VAR cost calculation (send-node))
        xCostPCARCC = [xCostPCARCC costVar];
    end

    for i=1:(l-onebit_counter-fullcv_counter) %kostos apo m ews
m+limit (ypo-ma8isis)
        costVar      =      costVar      +      PCsCompressCost(n,      q,
costOfInstruction); %COST of PCARCC
        costVar      =      costVar      +      PCsUncompressCost(n,      q,
costOfInstruction); %COST of PCARCC
        costVar      =      costVar      +      transmitValuesCost(q,
costOfBitTrans);%COST of PCARCC
        costVar = costVar + receiveValuesCost(q, costOfBitRec);
%COST of PCARCC

```

```

xCostPCARCC = [xCostPCARCC costVar];
end

%*****
*****

mpointer = mpointer + m + 1;

if ( (mpointer+m+1-1)>traceLength )
    lend=1;
end

end %end of while loop

%%Oi teleftaies metriseis pou den einai arketes gia loop
%%%%%%%%%

remainingMeasurementsNo = traceLength-mpointer+1;

if (remainingMeasurementsNo<=m)
    %Stelnei -remainingMeasurementsNo- (<=m) metriseis opws
    akriwvs einai
    for j=1:1:remainingMeasurementsNo

        pcarccReceiveNode = [pcarccReceiveNode
;traceMatrix(mpointer+j-1,:)];

        cost = cost + transmitValuesCost(n, costOfBitTrans);
        %COST of PC3 (send-node)

        cost = cost + receiveValuesCost(n, costOfBitRec); %COST
of PC3 (receive-node)

%*****
*****

        costVar = costVar + transmitValuesCost(n,
costOfBitTrans); %COST of PCARCC (send-node) *****

        costVar = costVar + receiveValuesCost(n, costOfBitRec);
        %COST of PCARCC (receive-node) *****

%*****
*****

```

```

        costNo = costNo + transmitValuesCost(n, costOfBitTrans);
% COST of SDF

        costNo = costNo + receiveValuesCost(n, costOfBitRec); %
COST of SDF

        %my statistics

        xCostPC3= [xCostPC3 cost];
        xCostSDF = [xCostSDF costNo];
        xCostPCARCC = [xCostPCARCC costVar];

    end

end %END of: if (remainingMeasuresNo<=m)

if (remainingMeasuresNo>m)
    %Stelnei -m*n- times opws akriws einai kai stoys 2
    mixanismoys
    for j=1:1:m
        H(j,:) = traceMatrix(mpointer+j-1,:); %Kataskevi History
matrix gia PCA
        pcarccReceiveNode = [pcarccReceiveNode
;traceMatrix(mpointer+j-1,)];

        cost = cost + transmitValuesCost(n, costOfBitTrans);
%COST of PC3 (send-node)
        cost = cost + receiveValuesCost(n, costOfBitRec); %COST
of PC3 (send-node)
%*****
%*****

        costVar = costVar + transmitValuesCost(n,
costOfBitTrans); %COST of PCARCC (send-node) *****
        costVar = costVar + receiveValuesCost(n, costOfBitRec);
%COST of PCARCC (receive-node)*****

%*****
%*****

        costNo = costNo + transmitValuesCost(n, costOfBitTrans);
%COST of SDF
        costNo = costNo + receiveValuesCost(n, costOfBitRec);
%COST of SDF

```

```

    %my statistics

    xCostPC3= [xCostPC3 cost];
    xCostSDF = [xCostSDF costNo];
    xCostPCARCC = [xCostPCARCC costVar];

end

    %KAI pompos KAI dektis, xrisimopoioun tis prohgoumenes -m-
    metrises ws memory gia PCA. Kanoun
    %PCA kai kratan tis prwtes -q- PCs wste h metavlitosita na
    einai -p%-
    [coeff, meanH, q] = InHistorymatrixOutPcacoef2 (H,p);
    %send-node
    cost = cost + createPcaCoeffCost(n,m,costOfInstruction);
    %COST of PC3
    %receive-node
    cost = cost + createPcaCoeffCost(n,m,costOfInstruction);
    %COST of PC3

    %*****
    %*****

    %send-node
    /*****

    costVar = costVar +
    createPcaCoeffCost(n,m,costOfInstruction); %COST of PCARCC
    %receive-
    node*****
    costVar = costVar +
    createPcaCoeffCost(n,m,costOfInstruction); %COST of PCARCC

    %*****
    %*****

    %Oles oi metrises tou traceMatrix apo -mpointer+m- ws -
    mpointer+m+remainingMeasurmentsNo-1- tha ipostoun
    %PCA, me vasi ton -coeff-. Etsi, kathe mia apo aftes tis
    metrises tha exei pleon -q- PCs.
    %Oles aftes oi PCA times tha apothikeftoun ston pinaka -
    finalPCAMEasurments-.

```

```

    finalPCAMeasurements = [
(traceMatrix((mpointer+m):(traceLength),:))
repmat(meanH, [(remainingMeasurementsNo-m),1]))*coeff(:,1:q)
];

    if( (remainingMeasurementsNo-m) >
(VarMemoryLength*(q+1)+2) ) % αν iparxei o aparaithtos ari8mos
deigmatwn gia mia nea VAR diadikasia

        % O pinakas finalPCAMeasurements me ta simpiesmena CVs
diastasews q, 8a
        % perasei apo ton mixanismo ARCC, oson afora ton
mixanismo PCARCC
        % etsi 8a exoume ta diastimata ypo-ma8isisis kai ypo-
simpieshs opws
        % ayta orizontai stin 8ewria
        [onebit_counter,fullcv_counter,
error_i_j,C_VAR,limit,RecNodeSamples]=ARCC(finalPCAMeasurements
(:,1:q),VarMemoryLength,VarMaxError);

        else % H diadikasia VAR den ekteleitai afou den arkoun ta
deigmata gia na kataskeyastei mnhmh.
            % Ta dedomena pernane ston receive Node opws akriwvs
kai sthn PC3
            flag=0;
            RecNodeSamples=finalPCAMeasurements(:, 1:q);
            fprintf('VAR memory cannot be constructed.\nCVs will be
forwarding through PC3 mechanism \n');

        end

        %Ypologismos twv PCA errors gia tis metriseis sto diastima m
ews m+remainingMeasurementsNo, oson
        %afora ton mixanismo PC3
        finalPcaMinusRealMeasurementsNorm = sqrt(sum( ( (
finalPCAMeasurements(:, 1:q)*coeff' +
repmat(meanH, [(remainingMeasurementsNo-m),1]) ) -
traceMatrix((mpointer+m):(traceLength),:)) .^2 ,2) );
        finalRealMeasurementsNorm = sqrt(sum(
traceMatrix((mpointer+m):(traceLength),:) .^2 ,2) );
        for zd=1:1:(remainingMeasurementsNo-m) %Prostasia, gia na
mhn ginei diairesis me to mhden

```

```

        if (finalRealMeauserementsNorm(zd)==0)
            lRealMeauserementsNorm(zd)=0.0000000000000001;
        end
    end

    finalPCAMEauserementsError =
    finalPcaMinusRealMeauserementsNorm ./
    finalRealMeauserementsNorm;

    %
    *****
    *****

    %Ypologismos twn VAR errors gia tis metriseis sto diastima
    m ews m+1 onon afora to mixanismo PCARCC

    %apo m ews m+limit (ypo-ma8isis)exoume idio sfalma me to
    PC3, enw apo m+limit ews remainingMeauserementsNo (ypo-simpiesi)

    %exartatai apo to an stelnetai onebit 'h fullcv ston
    receive-Node(RecNodeSamples)

    finalVarMinusRealMeauserementsNorm = sqrt(sum( ( (
    RecNodeSamples*coeff' + repmat(meanH, [(remainingMeauserementsNo-
    m),1]) ) - traceMatrix((mpointer+m):(traceLength),:) ).^2 ,2)
    );

    finalRealMeauserementsNorm = sqrt(sum(
    traceMatrix((mpointer+m):(traceLength),:) .^2 ,2) );

    for zd=1:1:(remainingMeauserementsNo-m) %Prostasia, gia na
    mhn ginei diairesh me to mhden
        if (finalRealMeauserementsNorm(zd)==0)
            lRealMeauserementsNorm(zd)=0.0000000000000001;
        end
    end

    finalVarMeauserementsError =
    finalVarMinusRealMeauserementsNorm ./
    finalRealMeauserementsNorm;

    %
    *****
    *****

    RecNode=(
    RecNodeSamples*coeff' +
    repmat(meanH, [(remainingMeauserementsNo-m),1]));
    pcarccReceiveNode=[pcarccReceiveNode ; RecNode];

    for j=1:1:(remainingMeauserementsNo-m)

```

```

    PC3totErrPlot (mpointer+m+j-
1)=finalPCAMEasurementsError (j);

    PCARCCtotErrPlot (mpointer+m+j-
1)=finalVarMeasurementsError (j);

    cost = cost + PCsCompressCost (n, q, costOfInstruction);
%COST of PC3
    cost = cost + transmitValuesCost (q, costOfBitTrans);
%COST of PC3
    cost = cost + receiveValuesCost (q, costOfBitRec); %COST
of PC3
    cost = cost + PCsUncompressCost (n, q, costOfInstruction);
%COST of PC3

    costNo = costNo + transmitValuesCost (n, costOfBitTrans);
%COST of SDF
    costNo = costNo + receiveValuesCost (n, costOfBitRec);
%COST of SDF

    if(flag==0) %An ta stoixeia perasoun ston upstream komvo
opws akriwvs einai epeidh den ektelesthke eh diadikasia VAR,
        %logw mi dinatothtas kataskeyhs mnimis, to kostos
einai idio
        %opws kai stin PC3
        costVar = costVar + PCsCompressCost (n, q,
costOfInstruction); %COST of PCARCC
        costVar = costVar + transmitValuesCost (q,
costOfBitTrans); %COST of PCARCC
        costVar = costVar + receiveValuesCost (q,
costOfBitRec); %COST of PCARCC
        costVar = costVar + PCsUncompressCost (n, q,
costOfInstruction); %COST of PCARCC
        xCostPCARCC = [xCostPCARCC costVar];
    end

    %my statistics

    xCostPC3= [xCostPC3 cost];

```

```

        xCostSDF = [xCostSDF costNo];
    end

    %Sto xroniko diasthma apo m ews m+remainingMeauserementsNo
    , to kostos tou PCARCC exartatai apo ta posa flag-bits kai
    %autousia CVs stal8uikan, sinepws

    if(flag==1)
        for i=1:onebit_counter %kostos apo m+limit ews l
        (l2) otan aposteletai flag
            costVar = costVar + PCsCompressCost(n, q,
            costOfInstruction); %COST of PCARCC
            costVar = costVar + PCsUncompressCost(n, q,
            costOfInstruction); %COST of PCARCC
            costVar = costVar + transmitValuesCost((1/4),
            costOfBitTrans);%COST of PCARCC
            costVar = costVar + receiveValuesCost((1/4),
            costOfBitRec); %COST of PCARCC
            costVar = costVar + costOfInstruction *
            limit^2; %COST of PCARCC (VAR cost calculation (send-node))
            costVar = costVar + costOfInstruction *
            limit^2; %COST of PCARCC (VAR cost calculation (receive-node))
            xCostPCARCC = [xCostPCARCC costVar];
        end

        for i=1:fullcv_counter %kostos apo m+limit ews l
        (l2), otan aposteletai olo to CV aytousio
            costVar = costVar + PCsCompressCost(n, q,
            costOfInstruction); %COST of PCARCC
            costVar = costVar + PCsUncompressCost(n, q,
            costOfInstruction); %COST of PCARCC
            costVar = costVar + transmitValuesCost(q,
            costOfBitTrans);%COST of PCARCC
            costVar = costVar + receiveValuesCost(q,
            costOfBitRec); %COST of PCARCC
            costVar = costVar + costOfInstruction *
            limit^2; %COST of PCARCC (VAR cost calculation (send-node))
            xCostPCARCC = [xCostPCARCC costVar];
        end
    end

```



```

        for i=1:((remainingMeauserementsNo-m)-
onebit_counter-fullcv_counter) %kostos apo m ews m+limit (ypo-
ma8isis)

            costVar = costVar + PCsCompressCost(n, q,
costOfInstruction); %COST of PCARCC

            costVar = costVar + PCsUncompressCost(n, q,
costOfInstruction); %COST of PCARCC

            costVar = costVar + transmitValuesCost(q,
costOfBitTrans);%COST of PCARCC

            costVar = costVar + receiveValuesCost(q,
costOfBitRec); %COST of PCARCC

            xCostPCARCC = [xCostPCARCC costVar];

        end

    end

%*****
%*****

end %END of: if (remainingMeauserementsNo>m)

```

ARCC.m

```

% Ektelei ton mixanismo ARCC se mia seira apo CVs pou
perioxontai sto Y

% H mnhmh ths diadikasia VAR einai p

% To megisto sfalma provlepeis einai maxerror

% Epistrefei san exodo to kostos(xCostVar) kai to sfalma
(VARtotErrPlot)

% tou mixanismou ARCC, ta CVs poy katexei o receive node
(RecNode) meta thn oloklirwsh toy

% mixanismou. Metrhtes pou ipologizoyh posa CVs stal8ikan
aytousia

% (fullcv_counter) kai posa antikatasta8hkan apo flag bit
(onebit_counter)

% To limit anaferetai ston elaxisto ari8mo CVs pou xreiazontai
prokeimenou

% na pragmatopoih8ei h diadikasia VAR

function [onebit_counter,fullcv_counter,
VARtotErrPlot,xCostVar,limit,RecNode]=ARCC(Y,p,maxerror)

costOfBitTrans=720; %COST of Transmission (nJoule)

```

```
costOfBitRec=110; %COST of Reception(nJoule)
costOfInstruction=4; %COST of Instruction(nJoule)
HeaderLength= 7; % Το mikos se bytes tis epikefalidas se enan
paketo
PayloadLenght = 20; %To mikos se bytes toy wfelimou fortiou enow
paketou
VariableLength = 4; %Ta bytes pou xreiazontai gia na apostalei
ena reading enos CV
onebit_counter=0;
fullcv_counter=0;

[lines,columns]= size(Y); % Diakrinoume apo posa CV apoteleitai
h eisodos kai poses metavlhtes exei to ka8e ena CV
VARTotErrPlot =[]; % se ayton ton pinaka , 8a krateitai to
sfalma metaxy toy
                % CV pou katexei RecNode se sxesh me ton
                % pragmatiko pou exei o Send Node gia ka8e
ena CV

packets=fix((columns*VariableLength)/PayloadLenght); %ipologizei
posa "plhrhs" paketa 8a apostaloun wste na metado8ei ena CV
bytes=mod((columns*VariableLength),PayloadLenght); % ipologizei
ta bytes pou 8a prepei na staloun apo to epipleon "mh plhrhs"
paketo
bits=          packets*(PayloadLenght+HeaderLength)*8          +
(bytes+HeaderLength)*8; %ipologizei posa bits 8a stelontai
ka8e fora pou stelnetai olo to cv
                %ka8e          paketo
exei PayloadLenght bytes wfelimo fortio kai HeaderLength bytes
epikefalida

cost_var = 0; % kostos arxikhs katastashes gia tin ARCC isoutai
me 0
xCostVar = []; %good
limit = p*(columns+1)+2; %ipologizei ton elaxisto ari8mo
deigmatwn(CV)
                %poy 8a prepei na perastoun ws eidodo
sth arfit.m , prokeimenoy na mporei
                %na eketeletsei

if (limit > lines )
```

```
error('Unable to construct memory for VAR procedure. Select  
a bigger value for l') %Den iparxei arketo disthma wste na  
"ftiaxtei" mnhmh gia na
```

```
%eketeletsei h diadikasia VAR
```

```
end
```

```
SendNode=Y;
```

```
RecNode=zeros(lines,columns);
```

```
RecNode(1:limit,:)=SendNode(1:limit,:);
```

```
VARtotErrPlot(:,1:limit)=0;
```

```
% Ta prwta limit context vectors ta pernaei o SendNode opws  
einai ston RecNode ,
```

```
% prokeimenou na dhmiourh8ei h mnhmh gia thn diadikasia VAR poy  
8a akolouthisei
```

```
% Opote h ARCC exei akriwvs to idio kostos me thn SDF
```

```
for i=1:limit
```

```
cost_var = cost_var + bits*costOfBitTrans +  
bits*costOfBitRec; %COST of ARCC
```

```
xCostVar = [xCostVar cost_var];
```

```
end
```

```
% Apo tin stigmh pou dhmiourghthei i mnhmh kai meta se ka8e enan  
CV o send
```

```
% node kanei provlepsi kai an einai mikrotero to sfalma tiw  
provlepsis apo
```

```
% to maxerror stelnei ena flag bit enw sthn alli periptwsh  
oloklhro to CV
```

```
for i=1:lines-limit;
```

```
[w,A,c]=arfit(SendNode(i:(i+limit-1),:), p, p); %eyresh  
twv parametrwn twv pinakwn A kai w me vash ta limit teleytaia  
deigmata
```

```
[pr_v]=predict(SendNode(i:(i+limit-1),:), p, A, w,  
columns,limit); % provlepsi toy epomenou CV, dld toy  
SendNode(i+limit, :)
```

```
error_i_i(i+limit)=norm(SendNode(i+limit,:)-  
pr_v')/norm(SendNode(i+limit,:)); % to sfalma tis provlepsis tou  
sendnode, gia to i+limit CV
```

```

    if ( error_i_i(i+limit) <= maxerror ) % an to sfalma tis
provlepsi einai mikrotero apo mia sigkekrimenh timh, o komvos
RecNode 8a provlepei to CV

        onebit_counter=onebit_counter+1; % 0
komvosSendNodestelnei ena bit ston RecNode

        [w,A,c]=arfit(RecNode(i:(i+limit-1),:), p, p);
%eyresh twn parametrwn twn pinakwn A kai w me vash ta limit
teleytaia deigmata,

%ekteleitai h provlepsi ston Recnode

        [pr_v]=predict(RecNode(i:(i+limit-1),:), p, A,
w,columns, limit); % provlepsi toy epomenou CV, dld toy
RecNode(i+limit, :)

        RecNode(i+limit,:)= pr_v';

        cost_var = cost_var +
((HeaderLength+1)*8)*costOfBitTrans +
((HeaderLength+1)*8)*costOfBitRec + costOfInstruction * limit^2;
% ka8e fora pou stelnertai ena flag bit ston Rec Node ,
apostelontai 64 bits

    else

        fullcv_counter=fullcv_counter+1; % 0 komvos
SendNode stelnei olokliro to CV ston RecNode

        RecNode(i+limit,:)= SendNode(i+limit,:);

        cost_var = cost_var + bits*costOfBitTrans +
bits*costOfBitRec + costOfInstruction * limit^2;

    end

    errorij=norm(SendNode(i+limit,:)-
RecNode(i+limit,:))/norm(SendNode(i+limit,:)); % Ipologizei to
sfalma anakataskeyis sto CV

    VARTotErrPlot = [VARTotErrPlot errorij]; % Simplirwnei ton
pinaka sfalmatwn anakataskeuis

    xCostVar = [xCostVar cost_var]; % COST of ARCC

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);
for(i=1:p)
    sum_=sum_+A(:,m*(i-1)+1:m*i)*v(lim-i,:)' ;
end
pr_v=sum_+w;
```

exe.m

```
load sensors_data_clear.mat

SendNode = variables7;
p = 3;
maxerror = 0.1;
m=8;
l=25;

[onebit_counter,fullcv_counter,totErrPlotARCC,xCostARCC,limit,Re
cNodeARCC]=ARCC(SendNode,p,maxerror);

[xCostPC3,xCostSDF,xCostPCARCC,totErrPlotPC3,totErrPlotPCARCC] =
PCARCC(SendNode, m , l , p, maxerror);

L=length(SendNode);

energyGainPC3_SDF = (xCostSDF(L)-xCostPC3(L))/xCostSDF(L);
energyGainPCARCC_SDF = (xCostSDF(L)-xCostPCARCC(L))/xCostSDF(L);
energyGainARCC_SDF = (xCostSDF(L)-xCostARCC(L))/xCostSDF(L);
energyGainARCC_PC3 = (xCostPC3(L)-xCostARCC(L))/xCostPC3(L);
energyGainPCARCC_PC3 = (xCostPC3(L)-xCostPCARCC(L))/xCostPC3(L);
```

```

energyGainARCC_PCARCC= (xCostPCARCC (L) -
xCostARCC (L) ) /xCostPCARCC (L) ;

errorRatePC3      = mean (totErrPlotPC3) ;
errorRatePCARCC   = mean (totErrPlotPCARCC) ;
errorRateARCC     = mean (totErrPlotARCC) ;

pcgARCC=(xCostSDF-xCostARCC) ./xCostSDF;
pcgPC3=(xCostSDF-xCostPC3) ./xCostSDF;
pcgPCARCC=(xCostSDF-xCostPCARCC) ./xCostSDF;

cumPCARCCError=cumsum (totErrPlotPCARCC) ;
cumPC3Error=cumsum (totErrPlotPC3) ;
cumARCCError=cumsum (totErrPlotARCC) ;

for i=1:L
    meanPCARCCtotErrPlot (i)= cumPCARCCError (i) /i;
    meanPC3totErrPlot (i)= cumPC3Error (i) /i;
    meanARCCtotErrPlot (i)= cumARCCError (i) /i;
end

wARCC=pcgARCC+1./ (1+meanARCCtotErrPlot) ;
wPC3=pcgPC3+1./ (1+meanPC3totErrPlot) ;
wPCARCC=pcgPCARCC+1./ (1+meanPCARCCtotErrPlot) ;

fARCC=(0.5) .*pcgARCC + (0.5) .* (1-meanARCCtotErrPlot) ;
fPC3=(0.5) .*pcgPC3 + (0.5) .* (1-meanPC3totErrPlot) ;
fPCARCC=(0.5) .*pcgPCARCC + (0.5) .* (1-meanPCARCCtotErrPlot) ;

```

ΑΝΑΦΟΡΕΣ

- [1] Dey and G. Abowd, “*Towards a Better Understanding of Context and Context-Awareness*”, *Proc. CHI’00*, The Hague, The Netherlands, 2000.
- [2] P. Prekop and M. Burnett, “*Activities, Context and Ubiquitous Computing*”, *Computer Communications*, Vol. 26, pp. 1168–1176, 2003 .
- [3] 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.
- [4] C. Perkins, “*Ad Hoc Networks*”, Addison-Wesley, Reading, MA, 2000.
- [5] E. Zivot and J. Wang, “*Modeling Financial Time Series with S-PLUS*”, Springer, New York (2003), pp 383-411.
- [6] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, B. Krogh, “*Energy-efficient surveillance system using wireless sensor networks*”, in *Proc. of 2nd Intl. Conf. on Mobile Systems, Applications, and Services, (ACM MobiSys ’04)*, pp.270–283, June 2004.
- [7] 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.
- [8] T. Schneider, A. Neumaier, “*Algorithm 808: Arfit-a Matlab package for the estimation of parameters and eigenmodes of multivariate autoregressive models*”, *ACM T. Math. Software* 27 58–65, (March 2001).
- [9] A. Neumaier, T. Schneider, “*Estimation of parameters and eigenmodes of multivariate autoregressive models*”, *ACM T Math. Software* 27 (March 2001) 27–57.
- [10] Lutkepohl, H.” *Introduction to Multiple Time Series Analysis (2nd ed.)*”. Springer-Verlag, Berlin, 1993.
- [11] Akaike, H. “*Autoregressive model fitting for control*” *Ann. Inst. Statist. Math.* 23,163-180. 1971
- [12] Schwarz, G.. “*Estimating the dimension of a model*”. *Ann. Statist.* 6, 461- 464, 1978.
- [13] 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.
- [14] Bjorck A. “*Numerical Methods for Least Squares Problems. Society for Industrial and Applied Mathematics*”, Philadelphia, PA, 1996.