Πολυτεχνειο Κρητης
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Πληροφορικης

Πρόβλεψη Μετοχών στο Apache Flink

Διπλωματικη Εργασια

της

ΜΑΝΑΡΑ ΧΡΙΣΤΙΝΑΣ

Εξεταστική επιτροπή: Αντώνιος Δεληγιαννάκης (επιβλέπων), Καθηγητής
Μίνως Γαροφαλάκης, Καθηγητής
Βασίλειος Σαμολαδάς, Αναπληρωτής Καθηγητής

Χανιά, Σεπτέμβριος 2021

Περίληψη

Η σύγχρονη εποχή χαρακτηρίζεται και ως εποχή των Μεγάλων Δεδομένων (Big Data), λόγω της πρωτοφανούς κλίμακας δεδομένων που παράγονται σε καθημερινή βάση και της ανάγκης ανάλυσης και εξαγωγής χρήσιμων αποτελεσμάτων σε μια πληθώρα διάφορων τομέων. Επιτακτική είναι η ανάγκη για την παρακολούθηση χιλιάδων ροών δεδομένων προκειμένου να ληφθούν αποφάσεις. Στον χρηματιστηριακό τομέα, ένας επενδυτής επιθυμεί να εντοπίσει τις εν δυνάμει ευκαιρίες, γεγονός που προσδίδει μεγάλη σημασία στην ενασχόληση με αυτό τον τομέα, καθώς η ορθή και αποδοτική επεξεργασία χρηματιστηριακών δεδομένων καθίσταται καθοριστική για την οικονομική ευημερία μιας χώρας. Στην περίπτωση του χρηματιστηριακού τομέα, οι ροές δεδομένων-μετοχές είναι συνεχείς και μακροσκελείς. Η παρούσα διπλωματική εργασία επεξεργάζεται κατανεμημένα και παράλληλα χιλιάδες χρηματιστηριακές μετοχές, μέσω της εύρεσης υψηλών συσχετίσεων που αφορούν σύνολα δύο μετοχών. Η διαδικασία αυτή πραγματοποιείται σε πραγματικό χρόνο και στοχεύει στην εύρεση μετοχών, των k πιο όμοιων, οι οποίες είναι ζωτικής σημασίας για την πρόβλεψη άλλων, οι οποίες δίδονται ως είσοδος, προκειμένου να προβλεφθούν. Αναπόφευκτη και ουσιαστική είναι η ανάγκη για την εκτέλεση της προσέγγισης σε εύλογα χρονικά πλαίσια, στα οποία αποδίδονται οι επιθυμητές απαντήσεις με την ταυτόχρονη αύξηση του πλήθους των δεδομένων στην είσοδο.

Το ζητούμενο ικανοποιείται (α) με την υλοποίηση και τη διαχείριση μιας σύνοψης στο σύστημα Synopsis Data Engine (SDE) (β) την εφαρμογή του αλγορίθμου Discrete Fourier Transform (DFT) που αποσκοπεί στη μείωση του απαιτούμενου αριθμού υποψήφιων όμοιων μετοχών (γ) την εφαρμογή του Multiple Linear Regression (MLR) μοντέλου για την πρόβλεψη των μετοχών. Για την εξαγωγή της πειραματικής διαδικασίας, ο αλγόριθμος ελέγχεται τόσο τοπικά όσο και απομακρυσμένα, πετυχαίνοντας ικανοποιητικά αποτελέσματα.

Λέξεις Κλειδιά

Σύνοψη, Discrete Fourier Transform (DFT), Multiple Linear Regression (MLR), μετοχές, συσχέτιση, ανάλυση δεδομένων, χρονοσειρές

Abstract

The modern age is also characterized as the age of Big Data, due to the unprecedented scale of data produced on a daily basis and the need to analyze and extract useful results in a variety of different fields. The need to monitor thousands of data streams in order to make decisions is imperative. In the stock market, an investor wants to identify potential opportunities, which is very important in dealing with this sector, as the correct and efficient processing of stock market data becomes crucial for a country's economic prosperity. In the case of the stock market, the stock-data flows are continuous and long. This dissertation processes thousands of stock market shares distributed and simultaneously, by finding high correlations that concern sets of two shares. This process is done in real time and aims to find shares of the k most similar, which are vital to the prediction of others, which are given as input, in order to be predicted. Inevitable and essential is the need to perform the approach in a reasonable time frame, to which the desired answers are attributed while increasing the amount of data at the input.

The request is satisfied by (a) implementing and managing a synopsis in the system Synopsis Data Engine (SDE) (b) the application of the Discrete Fourier Transform (DFT), which aims to reduce the required number of candidate similar stocks (c) the application of the Multiple Linear Regression (MLR) model for stock forecasting. For the extraction of the experimental process, the algorithm is checked both locally and in a computing cluster, achieving satisfactory results.

Keywords

Synopsis, Discrete Fourier Transform (DFT), Multiple Linear Regression (MLR), stocks, correlation, data analysis, time series

στην οικογένειά μου

Ευχαριστίες

Θα ήθελα καταρχάς, να ευχαριστήσω θερμά τον καθηγητή κ.Αντώνιο Δεληγιαννάκη για την επίβλεψη αυτής της διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο εργαστήριο Διανεμημένων Πληροφοριακών Συστημάτων και Εφαρμογών. Επίσης, ευχαριστώ ιδιαίτερα τον Αντώνιο Κονταξάκη για την καθοδήγησή του και την εξαιρετική συνεργασία που είχαμε. Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου και τις επιστήθιες φίλες μου, την Ήρα Κατάρα, τη Μαρία Μαραγκάκη για τις όμορφες στιγμές που μου χάρισαν, την καθοδήγηση και την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

Στη σύγχρονη εποχή, η μεγαλύτερη πρόκληση των υπολογιστικών συστημάτων, έγκειται στην αποδοτική αποθήκευση και ανάκτηση μεγάλου όγκου δεδομένων. Η ανάγκη αυτή προκύπτει με αφορμή τη ραγδαία αύξηση των δεδομένων που παρατηρείται στο Διαδίκτυο και αποκτά ολοένα και μεγαλύτερη σημασία λόγω του εύρους πληροφοριών που είναι δυνατόν να αντληθούν. Ο όρος *«Μεγάλα Δεδομένα»* (Big Data) χρησιμοποιείται για να αναπαρασταθεί ο τεράστιος όγκος δομημένων και αδόμητων δεδομένων, τόσο μεγάλος, όπου η επεξεργασία του χρησιμοποιώντας παραδοσιακές τεχνικές επεξεργασίας δεδομένων και λογισμικού, είναι πολύ δύσκολη. Επομένως, τα *«Μεγάλα Δεδομένα»* προβάλλουν αρκετές προκλήσεις εξαιτίας του όγκου τους, της ταχύτητας με την οποία δημιουργούνται, της ποικιλομορφίας τους, καθώς επίσης και της πολυπλοκότητάς τους. Με την πρόοδο της τεχνολογίας και του ολοένα αυξανόμενου όγκου των δεδομένων, καθίσταται επιτακτική ανάγκη της ταχύτερης και αποδοτικότερης ανάλυσης τους. Νέα εργαλεία, μέθοδοι και αρχιτεκτονικές επιστρατεύονται με στόχο να αποθηκεύσουν και να διαχειριστούν τα νέα δεδομένα. Μερικά χαρακτηριστικά παραδείγματα τέτοιων εργαλείων είναι τα εξής: Hadoop [1], Map Reduce [2], Apache Spark [3], Storm [4]. Στην παρούσα διπλωματική εργασία πραγματοποιείται η χρήση του Apache Flink [5].

Συχνά, είναι επιτακτική η ανάγκη της επεξεργασίας και εξαγωγής χρήσιμων αποτελεσμάτων μεγάλου όγκου δεδομένων σε πραγματικό χρόνο, με απώτερο στόχο τη λήψη σημαντικών αποφάσεων. Ειδικότερα, στον χρηματιστηριακό τομέα, ζωτικής σημασίας είναι η πρόβλεψη της πορείας των μετοχών, η οποία στοχεύει στην όσο το δυνατό μειωμένη ανάληψη ρίσκου και κατ΄ επέκταση τη μεγιστοποίηση του κέρδους. Στην πραγματικότητα, ο εκάστοτε επενδυτής ενδιαφέρεται για την εφαρμογή της επενδυτικής στρατηγικής του σε κάποια μετοχή σε μια συγκεκριμένη χρονική στιγμή, η οποία θα επιφέρει τα καλύτερα δυνατά αποτελέσματα. Φυσικά, κατά την επίτευξη του στόχου αυτού τονώνεται η οικονομία και κατ΄ επέκταση αναπτύσσεται η χώρα που επιδίδεται στο χρηματιστήριο.

1.1 Αντικείμενο της διπλωματικής

Το αντικείμενο της παρούσας διπλωματικής εργασίας επικεντρώνεται στην αποδοτική πρόβλεψη των χρηματιστηριακών μετοχών. Σε πρώτο επίπεδο, πραγματοποιείται η αποτελεσματική διαχείριση χιλιάδων μετοχών, προκειμένου να επιτευχθεί η εξαγωγή χρήσιμων αποτελεσμάτων, τα οποία θα αποτελέσουν ακρογωνιαίο λίθο για την υλοποίηση της πρόβλεψης. Με τη χρήση

της σύνοψης καθίσταται δυνατή η επεξεργασία χιλιάδων μετοχών, καθώς σε καθεμιά από αυτές εφαρμόζεται ο αλγόριθμος DFT σε συγκεκριμένα χρονικά παράθυρα, με στόχο την εύρεση ζευγών συσχετίσεων. Αυτό που επιδιώκεται είναι η εύρεση των συσχετίσεων, καθώς μέσω αυτών, είναι δυνατόν να προβλεφθεί η τάση των εκάστοτε επιθυμητών μετοχών. Στην ουσία, παρατηρώντας τις παρελθοντικές τιμές των συσχετισμένων μετοχών, επιτυγχάνεται η πρόβλεψη της τάσης των εκάστοτε επιθυμητών μετοχών.

Προκειμένου να υλοποιηθούν τα παραπάνω πραγματοποιείται η χρήση της μηχανής συνόψεων SDE και του Kafka. Σε πρώτη φάση, εισάγονται τα δεδομένα στο Kafka, τα οποία επεξεργάζονται ως streams από το SDE, το οποίο παρέχει τη δυνατότητα διαχείρισης και επεξεργασίας μεγάλου όγκου δεδομένων, κατανεμημένα και παράλληλα. Επιπλέον, εξετάζεται η αποδοτικότητα του αλγορίθμου μεταβάλλοντας διάφορες παραμέτρους, ώστε να αποδειχθεί αν διαθέτει τη δυνατότητα του scalability σε σχέση με τους πόρους, τα streams εισόδου και τη λίστα των επιθυμητών μετοχών που πρόκειται να προβλεφθούν.

1.2 Οργάνωση του τόμου

Στο Κεφάλαιο 2 αναλύεται το θεωρητικό υπόβαθρο, το οποίο είναι χρήσιμο για την κατανόηση του μαθηματικού υποβάθρου του αλγορίθμου DFT. Στο Κεφάλαιο 3 πραγματοποιείται η παρουσίαση των εργαλείων που χρησιμοποιούνται προκειμένου να επιτευχθεί το ζητούμενο. Στο Κεφάλαιο 4 αναλύεται λεπτομερώς η υλοποίηση και ο τρόπος που εφαρμόζεται ο αλγόριθμος. Στο Κεφάλαιο 5 παρουσιάζονται τα πειραματικά αποτελέσματα που λαμβάνονται, ώστε να αποδειχθεί η αποδοτικότητα του αλγορίθμου. Στο Κεφάλαιο 6 παρατίθενται τα συμπεράσματα, καθώς οι μελλοντικές προεκτάσεις-βελτιώσεις που είναι δυνατόν να εφαρμοστούν.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό παρουσιάζονται αναλυτικά βασικές έννοιες που χρησιμοποιούνται για την επίτευξη του στόχου της παρούσας διπλωματικής εργασίας.

2.1 Ροές Δεδομένων

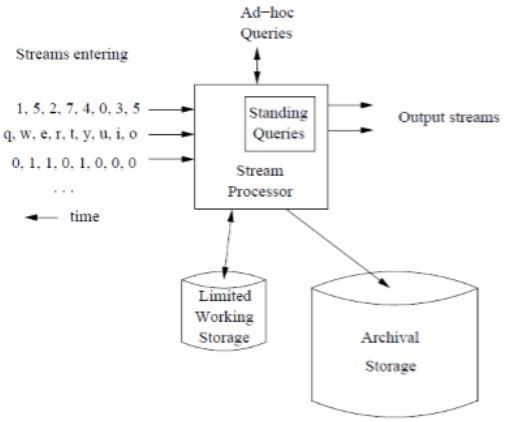
Καθώς το Διαδίκτυο έχει καταστήσει ρουτίνα την επικοινωνία μεταξύ των μηχανών, έχει αναπτυχθεί μια κατηγορία εφαρμογών που θέτουν σε δοκιμασία το παραδοσιακό πρότυπο των συστημάτων βάσεων δεδομένων. Αυτό το ζήτημα ανακύπτει καθώς τα δεδομένα εισόδου καταφθάνουν με ρυθμό που δεν μπορεί να ελεγχθεί από το Σύστημα Διαχείρισης Βάσεων Δεδομένων (Database Management System). Χαρακτηριστικά παραδείγματα ροών δεδομένων είναι οι ροές δεδομένων που προέρχονται από δορυφόρους και οι ροές που σχετίζονται με τα «χτυπήματα» που σχηματίζονται από τα αιτήματα που υποβάλλονται σε έναν ιστότοπο.

Ορισμός 1. *Μια ροή δεδομένων (Data Stream) είναι μια ακολουθία πλειάδων που καταφθάνουν σε κάποια τοποθεσία συνήθως με τόσο γρήγορο ρυθμό που η επεξεργασία και η αποθήκευσή τους στην ολότητά τους καθίσταται δύσκολη [6].*

Προκειμένου να πραγματοποιηθεί η αποτελεσματική διαχείριση των ροών δεδομένων εισάγεται η έννοια του Συστήματος Διαχείρισης Ροών Δεδομένων (Data Stream Management System) [6].

Ορισμός 2. *Ένα ΣΔΡΔ δέχεται δεδομένα υπό μορφή ροών. Τηρεί έναν αποθηκευτικό χώρο εργασίας και έναν χώρο μόνιμης αποθήκευσης. Ο αποθηκευτικός χώρος εργασίας είναι περιορισμένος, αν και μπορεί να περιλαμβάνει δίσκους.Ένα ΣΔΡΔ δέχεται είτε ad-hoc ερωτήματα είτε στάσιμα ερωτήματα, τα οποία αποθηκεύονται από το σύστημα και εκτελούνται κάθε στιγμή επί της ροής ή των ροών εισόδου [6]. Στο Σχήμα 2.1 αποδίδεται το ΣΔΡΔ.*

Τα ερωτήματα, είτε είναι ad-hoc είτε στάσιμα, που υποβάλλονται σε ένα ΣΔΡΔ, πρέπει να εκφραστούν με τέτοιο τρόπο ώστε να απαντηθούν με την χρήση περιορισμένων τμημάτων των ροών δεδομένων. Ένα ΣΔΡΔ δεν είναι εφικτό να αποθηκεύει και να υποβάλλει ερωτήματα σε ροές που επεκτείνονται επ΄ αόριστον στο παρελθόν. Προκειμένου να αντιμετωπιστεί αυτός



Σχήμα 2.1: *Σύστημα Διαχείρισης Βάσεων Δεδομένων*

ο περιορισμός εισάγεται η έννοια του *κυλιόμενου παραθύρου* (sliding window). Στην πραγματικότητα τα παράθυρα διατηρούν το πιο πρόσφατο τμήμα της ροής. Τα παράθυρα χωρίζονται σε δύο κατηγορίες ανάλογα με τι αναπαριστά το μέγεθος τους. Πιο συγκεκριμένα, υπάρχουν τα time-based παράθυρα, των οποίων το μέγεθος υποδηλώνει χρόνο και τα count-based, των οποίων το μέγεθος υποδηλώνει τον αριθμό των tuples.

2.2 Σύνοψη

Οι τεράστιες ροές δεδομένων δημιουργούν την ανάγκη για την ύπαρξη τόσο αλγορίθμων όσο και δομών δεδομένων, που θα παρέχουν ταχεία απάντηση σε ερωτήματα που τίθενται σε τέτοιου είδους ροές δεδομένων. Μια τέτοια δομή δεδομένων συνιστά και η *σύνοψη* (synopsis). Η τελευταία χρησιμοποιεί περιορισμένους υπολογιστικούς πόρους, παρέχει προσεγγιστικές απαντήσεις σε ερωτήματα και μόνη προϋπόθεση για την απάντηση των εκάστοτε ερωτημάτων είναι να πραγματοποιηθεί ένα πέρασμα στα δεδομένα. Στην ουσία μια σύνοψη αντιπροσωπεύει μια δομή δεδομένων, η οποία είναι μικρότερη από τα εκάστοτε σύνολα δεδομένων. Με αυτό τον τρόπο είναι δυνατόν να συμπιεστούν μεγάλα σύνολα δεδομένων σε λογαριθμική ή σταθερή χωρική πολυπλοκότητα.

Ορισμός 3. *Μία $f(n)$-σύνοψη δομή δεδομένων για μια κλάση Q ερωτημάτων είναι μια δομή δεδομένων, η οποία παρέχει ακριβείς ή προσεγγιστικές απαντήσεις στα ερωτήματα της κλάσης Q και χρησιμοποιεί $\mathcal{O}(f(n))$ χώρο για ένα σύνολο δεδομένων μεγέθους n όταν $f(n) = \mathcal{O}(n^e)$, για μια σταθερά $e < 1$ [7].*

2.2.1 Πλεονεκτήματα

Μια σύνοψη δομή δεδομένων παρέχει [7]:

- ταχύτητα επεξεργασίας: μια σύνοψη διατηρείται στην μνήμη, με αποτέλεσμα να παρέχει γρήγορες απαντήσεις σε ερωτήματα, αφού αποφεύγονται οι προσβάσεις στο δίσκο,

- χαμηλό κόστος: μια σύνοψη έχει μικρό αντίκτυπο στις συνολικές χωρικές απαιτήσεις του συστήματος,
- καλύτερη επίδοση συστήματος,
- γρήγορη μεταφορά: μια σύνοψη που διατηρείται στο δίσκο μπορεί εύκολα και ταχύτατα να μεταφερθεί στην μνήμη με λιγοστές προσβάσεις στο δίσκο και
- μια απεικόνιση των δεδομένων, όταν είναι αδύνατο να πραγματοποιηθεί πρόσβαση σε αυτά.

2.2.2 Εκτίμηση αξιοπιστίας σύνοψης

Μια σύνοψη μπορεί να εκτιμηθεί σύμφωνα με τις εξής πέντε μετρικές:

- την απόδοση: σχετίζεται με το εύρος και τη σημαντικότητα των ερωτημάτων,
- την ποιότητα απάντησης: ακρίβεια των απαντήσεων σε ερωτήματα που τίθενται,
- το αποτύπωμα στο χώρο,
- τον χρόνο εκτέλεσης του ερωτήματος,
- τον υπολογιστικό/update χρόνο.

2.3 Μετρικές Απόστασης

Στα μαθηματικά, μια μετρική απόστασης [8] είναι μια συνάρτηση, η οποία ορίζει μια απόσταση μεταξύ ενός ζευγαριού στοιχείων σε ένα n-διάστατο χώρο. Οποιαδήποτε μετρική απόστασης πρέπει να ικανοποιεί τα παρακάτω κριτήρια:

1. $d(x, y) \geq 0$, θετικές αποστάσεις
2. $d(x, y) = 0 \Leftrightarrow x = y$, μηδενικές αποστάσεις αντιστοιχούν σε σημεία που συμπίπτουν
3. $d(x, y) = d(y, x)$, συμμετρία απόστασης
4. $d(x, z) \leq d(x, y) + d(y, z)$, τριγωνική ανισότητα

Υπάρχουν πολλές διαφορετικές μετρικές απόστασης στη βιβλιογραφία, συμπεριλαμβανομένου της Ευκλείδειας, Hamming, Cosine, Chebyshev, Manhattan, κτλ. Στην παρούσα διπλωματική πραγματοποιείται η χρήση της Ευκλείδειας απόστασης.

2.3.1 Ευκλείδεια Απόσταση

Έστω δύο σημεία x και y. Το μήκος του ευθυγράμμου τμήματος που ενώνει τα δύο αυτά σημεία καθορίζει την Ευκλείδεια απόσταση. Αν $x = (x_1, x_2, \ldots, x_n)$ και $y = (y_1, y_2, \ldots, y_n)$

δύο σημεία στον Ευκλείδειο n-διάστατο χώρο, τότε η απόσταση των σημείων αυτών δίνεται σύμφωνα με την Εξίσωση 2.1, στον οποίο πραγματοποιείται η χρήση του Πυθαγορείου Θεωρήματος:

$$d(x,y) = d(y,x) = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2 + \cdots + (y_n - x_n)^2} = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2} \quad (2.1)$$

2.4 Pearson Correlation

Ο δείκτης συσχέτισης (correlation coefficient) μεταξύ δύο χρονοσειρών $X = (x_1, x_2, \ldots, x_n)$ και $Y = (y_1, y_2, \ldots, y_n)$ είναι ένα σημαντικό στατιστικό κριτήριο, το οποίο χρησιμοποιείται για τη διαπίστωση της αλληλεξάρτησης μεταξύ δύο μεταβλητών. Ο δείκτης συσχέτισης είναι αντιστρόφως ανάλογος με την μεταξύ τους απόσταση και κυμαίνεται μεταξύ του -1 και του 1, με τη μηδενική τιμή να υποδηλώνει ότι οι μεταβλητές είναι εντελώς ανεξάρτητες. Μια θετική τιμή σημαίνει ότι οι μεταβλητές X και Y συσχετίζονται, ενώ μια αρνητική τιμή υποδηλώνει αρνητική συσχέτιση. Ένας από τους διαφορετικούς τύπους δεικτών συσχέτισης είναι ο Pearson. Ο δείκτης συσχέτισης Pearson μεταξύ δύο μεταβλητών X και Y, ορίζεται ως εξής:

$$\mathrm{corr}(X,Y) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} \quad (2.2)$$

όπου τα $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ και $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, αντιπροσωπεύουν τις μέσες τιμές των X και Y αντίστοιχα.

Οι πληροφορίες που μπορεί να εξαχθούν από το δείκτη συσχέτισης Pearson σχετίζονται με το αν υπάρχει γραμμική συσχέτιση μεταξύ δύο μεταβλητών X και Y, το είδος (κατεύθυνση) της συσχέτισης, καθώς και το βαθμό στον οποίο συσχετίζονται. Αξίζει να σημειωθεί ότι δείκτης συσχέτισης είναι ένα στατιστικό κριτήριο που πληροφορεί μόνο για τη συμμεταβολή των δύο μεταβλητών που μελετώνται και όχι για το εάν υπάρχει αιτιώδης σχέση μεταξύ τους.

2.5 Correlation με βάση την Ευκλείδεια Απόσταση

Έστω δύο χρονοσειρές X και Y, όπου οι $\hat{X}$ και $\hat{Y}$ αντιπροσωπεύουν τις κανονικοποιημένες χρονοσειρές αντίστοιχα. Πιο συγκεκριμένα, κάθε στοιχείο της χρονοσειράς X, συμβολίζεται ως x_i και η κανονικοποιημένη τιμή του υπολογίζεται ως εξής:

$$\hat{x}_i = \frac{x_i - \bar{x}}{\sigma_x} \quad (2.3)$$

όπου το $\bar{x}$ αντιπροσωπεύει τη μέση τιμή της χρονοσειράς X και σ_x την τυπική απόκλιση ως εξής:

$$\sigma_x = \sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \quad (2.4)$$

Υποθέτοντας δύο ροές δεδομένων X και Y, η συσχέτιση αυτών μπορεί να υπολογιστεί μέσω του παρακάτω τύπου:

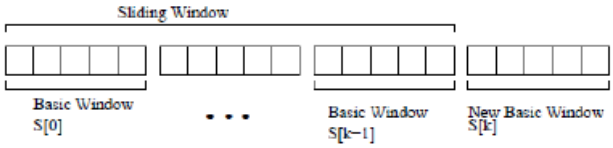
$$\text{corr}(X,Y) = 1 - \frac{1}{2}d^2(\hat{X},\hat{Y}) \tag{2.5}$$

όπου η ποσότητα $d(\hat{X},\hat{Y})$ συνιστά την Ευκλείδεια Απόσταση μεταξύ των κανονικοποιημένων χρονοσειρών $\hat{X}$ και $\hat{Y}$. Η απόδειξη αυτού παρατίθεται στο [9]. Ο μετασχηματισμός αυτός καθίσταται αρκετά σημαντικός λόγω της αποδοτικότητά του στην προσεγγιστική ομοιότητα streaming δεδομένων.

2.6 Sliding Window Model

Ένας αλγόριθμος που επεξεργάζεται ροές δεδομένων, δεν είναι εφικτό να αποθηκεύσει αυτές τις ροές, ώστε να τις αναλύσει αργότερα. Προκειμένου να αντιμετωπιστεί αυτός ο περιορισμός εισάγεται η έννοια του κυλιόμενου παραθύρου (sliding window) [9]. Στις χρηματιστηριακές εφαρμογές, ειδικά, είναι κατάλληλο για την επεξεργασία των ρευμάτων δεδομένων. Δεδομένου του μεγέθους του sliding window ως w και ενός τωρινού χρονικού σημείου t, η ανάλυση θα πραγματοποιηθεί στο υποσύνολο $s[t-w+1...t]$, [9]. Επιπλέον, λόγω του γεγονότος ότι το μέγεθος του παραθύρου αντιπροσωπεύει τον χρόνο, το παράθυρο είναι ένα time-based sliding window.

Στην πραγματικότητα ένα sliding window αποτελεί ένα χρονικό διάστημα ενδιαφέροντος, το οποίο καθορίζεται από τον χρήστη και χωρίζεται σε k ίσα μικρότερα χρονικά διαστήματα, τα οποία ονομάζονται basic windows και χρησιμεύουν στην αποτελεσματική εξάλειψη παλαιών δεδομένων και την ενσωμάτωση νέων, [9]. Αυτό σημαίνει ότι ισχύει b < w, όπου b το μέγεθος του basic window και w το μέγεθος του sliding window. Το σύστημα διατηρεί μια περίληψη των basic windows, όπως και ολόκληρου του sliding window. Μόλις συμπληρωθούν όλα τα basic windows ενός sliding window, τότε το παλαιότερο χρονικά basic window απορρίπτεται και αντ΄ αυτού στη συνεισφορά των basic windows προσμετράται το νεότερο χρονικά basic window, όπως φαίνεται στο Σχήμα 2.2.



Σχήμα 2.2: *Sliding Window Model [9]*

Πίνακας 1: Σύμβολα

w	μέγεθος sliding window
b	μέγεθος basic window
k	αριθμός basic windows που περιέχονται σε ένα sliding window

(2.6)

Για παράδειγμα, το άθροισμα πάνω σε ένα sliding window ανανεώνεται κατ΄ αυτόν τον

τρόπο:

$$\sum_{\text{new}}(s) = \sum_{\text{old}}(s) + \sum S[k] - \sum S[0] \tag{2.7}$$

όπου $S[0]$ το παράθυρο που πρόκειται να αφαιρεθεί - το παλαιότερο χρονικά, $S[k]$ το χρονικό παράθυρο που πρόκειται να προστεθεί - το νεότερο χρονικά.

2.7 Discrete Fourier Transform

Στην παρούσα διπλωματική εργασία, η υλοποίηση της εκτίμησης της συσχέτισης των εκάστοτε χρονοσειρών πραγματοποιείται με βάση τον αλγόριθμο Discrete Fourier Transform (DFT) και βασίζεται κυρίως στο σύστημα StatStream [9]. Πιο συγκεκριμένα, οι DFT συντελεστές (coefficients) παρέχουν τρεις αξιοσημείωτες ιδιότητες, οι οποίες είναι οι εξής:

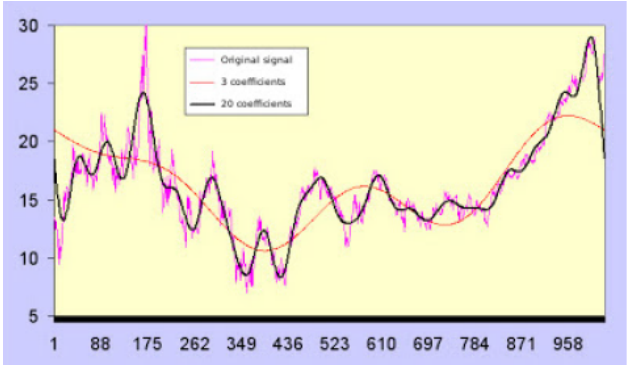
- διατηρούν την Ευκλείδεια Απόσταση, δηλαδή $d(X,Y) = d(DFT(X), DFT(Y))$,
- η πλειονότητα της ενέργειας του σήματος συγκεντρώνεται στους πρώτους συντελεστές και,
- διατηρούν την ποσότητα: $\text{corr}(x,y) \geq 1 - \epsilon^2 \Rightarrow d_n(\hat{X}, \hat{Y}) \leq \epsilon$, η οποία καταδεικνύει ότι έχει νόημα η αναζήτηση ομοιότητας των χρονοσειρών, όταν αυτή πραγματοποιείται ανά ζεύγη.

Οι DFT συντελεστές υπολογίζονται με βάση την Εξίσωση 2.8:

$$X_F = \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} x_i e^{\frac{-j2\pi Fi}{n}}, \quad F = 0, 1, \ldots, n-1 \tag{2.8}$$

όπου $j = \sqrt{-1}$, F είναι το πλήθος των συντελεστών και n ο αριθμός των δειγμάτων.

Η συμπίεση επιτυγχάνεται περιορίζοντας το F, στον παραπάνω τύπο, σε λίγους συντελεστές. Στην παρούσα εργασία χρησιμοποιούνται οι πρώτοι οκτώ(8) συντελεστές, προκειμένου να επιτευχθεί η σύγκριση των χρονοσειρών. Αυτή η επιλογή επικυρώνεται λόγω της συγκέντρωσης της ενέργειας του Fourier σήματος στους πρώτους συντελεστές [10].



Σχήμα 2.3: *Προσέγγιση σήματος DFT [11]*

Οι DFT συντελεστές ανανεώνονται σταδιακά κατά τη λειτουργία των sliding windows. Υποθέτοντας ένα παράθυρο μεγέθους w, με slide b, τότε για τον F-th συντελεστή προκύπτει η Εξίσωση 2.9.

$$X_F^{\text{new}} = e^{\frac{i2\pi bF}{w}} X_F^{\text{old}} + \frac{1}{N}\left(\sum_{k=0}^{b-1} x_{w+k} e^{\frac{i2\pi F(b-i)}{w}} - \sum_{k=0}^{b-1} x_k e^{\frac{i2\pi F(b-i)}{w}}\right) \tag{2.9}$$

Επομένως, προκειμένου να επιτευχθεί η ανανέωση των συντελεστών με βάση κάποιο χρονικό παράθυρο, καθώς καταφθάνουν νέα δεδομένα, χρειάζεται να αποθηκευτεί η ποσότητα, $\sum_{k=0}^{b-1} x_k e^{\frac{i2\pi F(b-i)}{w}}$ (για τον F-th συντελεστή), για κάθε συντελεστή που θα συμπεριληφθεί στην προσέγγιση [8].

2.7.1 Προσέγγιση time series με DFT συντελεστές

Το πλήθος των DFT συντελεστών που απαιτούνται, ώστε να αναπαρασταθούν στο πεδίο των συχνοτήτων, χωρίς απώλεια πληροφορίας, είναι $m = \frac{n}{2}$ (λόγω της συμμετρικής ιδιότητας), όπου m είναι το πλήθος των συντελεστών και n το πλήθος των data points. Καθώς, η πλειονότητα της ενέργειας του σήματος συγκεντρώνεται τυπικά στους πρώτους DFT συντελεστές, το πλήθος των οποίων είναι ικανό να προσεγγίσει με ακρίβεια τις αρχικές χρονοσειρές, κρατείται ένα μικρό σύνολο συντελεστών προκειμένου να προσεγγιστεί το αρχικό σήμα. Αναλυτικότερα, ένα πλήθος m συντελεστών είναι εφικτό να αναπαραστήσει μια χρονοσειρά με n στοιχεία, όπου $m << \frac{n}{2}$. Γενικά, η τιμή του m κυμαίνεται από 16...128. Το Σχήμα 2.3 αναπαριστά μια χρονοσειρά, η οποία αποτελείται από $n = 1000$ σημεία και προσεγγίζεται τόσο με $m = 3$ (κόκκινη καμπύλη) όσο και με $m = 20$ (μαύρη καμπύλη) συντελεστές. Είναι εμφανές ότι όσο λιγότεροι συντελεστές συμπεριλαμβάνονται στην προσέγγιση, τόσο πιο ομαλή είναι η καμπύλη, με λιγότερες διακυμάνσεις. Αυτό οφείλεται και στη φύση του σήματος, καθώς ένα αρχικά ομαλό σήμα, θα χρειαστεί λιγότερους συντελεστές προκειμένου να προσεγγιστεί, σε αντίθεση με ένα αρχικό σήμα, το οποίο παρουσιάζει αρκετές μεταβολές, που θα χρειαστεί περισσότερους συντελεστές.

Η χωρική πολυπλοκότητα της αποθήκευσης των χρονοσειρών είναι $O(m)$ αντί $O(n)$, καθώς χρειάζεται να αποθηκευτούν μόνο οι απαραίτητοι DFT συντελεστές και όχι ολόκληρες οι χρονοσειρές (αυτό μπορεί να υλοποιηθεί με ένα πέρασμα στα δεδομένα). Επιπρόσθετα, αξίζει να σημειωθεί ότι η συσχέτιση των χρονοσειρών μπορεί να εκτιμηθεί μέσω της προσέγγισης των πραγματικών χρονοσειρών ως εξής:

$$\text{corr}(X,Y) = 1 - \frac{1}{2}d^2(\hat{X},\hat{Y}) = 1 - \frac{1}{2}d^2(DFT(\hat{X}), DFT(\hat{Y})) \tag{2.10}$$

Ωστόσο, το υπολογιστικό κόστος του υπολογισμού της συσχέτισης μεταξύ δύο ροών δεδομένων, παραμένει $O(n)$, καθώς είναι απαραίτητο να πραγματοποιηθεί ένα πλήρες πέρασμα στα αρχικά δεδομένα [8].

2.7.2 Grid Structure

Ο απώτερος στόχος είναι η εύρεση ζευγαριών ρευμάτων δεδομένων με υψηλή συσχέτιση, η οποία εξαρτάται άμεσα από την Ευκλείδεια Απόσταση, γεγονός που υποδηλώνει ότι οι ροές

δεδομένων με μεγάλη απόσταση μεταξύ τους, αυτόματα απορρίπτονται ως εν δυνάμει υποψήφιοι ομοιότητας. Πιο συγκεκριμένα, η μελέτη επικεντρώνεται στην ανίχνευση της υπέρβασης του κατωφλίου, $t \in [0, 1]$ (ορίζεται από τον χρήστη), από τις εκάστοτε συσχετισμένες χρονοσειρές. Όπως αποδεικνύεται στο ερευνητικό άρθρο [9], προκειμένου η συσχέτιση να είναι μεγαλύτερη από την τιμή του κατωφλίου t, η Ευκλείδεια Απόσταση μεταξύ των εξεταζόμενων χρονοσειρών X και Y, πρέπει να είναι μικρότερη από την τιμή ϵ, με $t = 1 - \epsilon^2$.

Με την χρήση των κανονικοποιημένων τιμών των DFT συντελεστών, οι αρχικές χρονοσειρές χαρτογραφούνται σε έναν οριοθετημένο χώρο χαρακτηριστικών. Αποδεικνύεται [9] ότι η νόρμα κάθε συντελεστή κυμαίνεται από το $-\frac{\sqrt{2}}{2}$ έως το $\frac{\sqrt{2}}{2}$, με αποτέλεσμα ο DFT χώρος να καθορίζεται από ένα κύβο με διάμετρο $\sqrt{2}$. Η δομή του Grid είναι αποδοτική για την εύρεση των γειτονικών χρονοσειρών. Χρησιμοποιώντας τους πρώτους $\hat{m}$- κανονικοποιημένους συντελεστές, κατασκευάζεται ένα $\hat{m}$-διάστατο πλέγμα, το οποίο χωρίζεται σε ισοκατανεμημένα κελιά, διαμέτρου ϵ. Στην πραγματικότητα, υπάρχουν $\left(2\left\lceil\frac{\sqrt{2}}{2\epsilon}\right\rceil\right)^{\hat{m}}$ συνολικά κελιά. Τυπικά, οι τιμές $\hat{m} = 3$ ή $\hat{m} = 4$ είναι αποδοτικές. Κάθε ροή δεδομένων χαρτογραφείται σε ένα κελί στο πλέγμα. Υποθέτοντας ότι η ροή δεδομένων X κατακερματίζεται στο κελί, $(c1, c2, \ldots, c_{\hat{m}})$, προκειμένου να εντοπισθούν οι ροές δεδομένων που συσχετίζονται με το X (η συσχέτιση να μην υπερβαίνει το κατώφλι t), θα χρειαστεί να ελεγχθούν μόνο τα γειτονικά κελιά, καθώς μόνο σε αυτά υπάρχουν οι εν δυνάμει υποψήφιοι. Αυτές οι ροές δεδομένων συνιστούν ένα υπερσύνολο από πραγματικά συσχετισμένα streams. Είναι αναγκαίο να αναφερθεί ότι από την στιγμή που η διάμετρος των κελιών είναι ϵ και λόγω της χρήσης των $\hat{m}$ συντελεστών, είναι σίγουρο πως τα μη γειτονικά κελιά θα έχουν $d_{\hat{m}}(X, Y) > \epsilon$, με αποτέλεσμα να πιστοποιείται ότι η συσχέτιση δεν υπερβαίνει το t. Επομένως, με βάση αυτό το γεγονός δεν υπάρχουν false negatives στο σύστημα. Ωστόσο, η μέθοδος αυτή της ευρετηρίασης indexing είναι δυνατόν να παράγει false positives λόγω της συνθήκης: $d(X, Y) \geq d_{\hat{m}}(X, Y)$. Αυτό σημαίνει ότι η Ευκλείδεια Απόσταση όλων των σημείων των X και Y χρονοσειρών μπορεί να είναι μεγαλύτερη από την αντίστοιχη απόσταση που προκύπτει από τους πρώτους $\hat{m}$ συντελεστές, επομένως η συσχέτισή τους είναι μικρότερη από το κατώφλι t. Έτσι, μετά τον καθορισμό των υποψήφιων ζευγαριών από τα γειτονικά κελιά, απαιτείται ο υπολογισμός της συσχέτισής τους, η οποία βασίζεται στην χρήση των μη κανονικοποιημένων συντελεστών, ώστε να εξαλειφθούν οι false positives τιμές [8].

Η εκτίμηση της προσέγγισης των χρονοσειρών με βάση τους DFT συντελεστές υλοποιείται ως εξής: οι συντελεστές τοποθετούνται σε buckets, με στόχο όμοιες χρονοσειρές να ενταχθούν στο ίδιο ή σε κάποιο γειτονικό bucket, ενώ οι υπόλοιπες σε απομακρυσμένα, ώστε να μην πραγματοποιηθεί ποτέ σύγκριση για ομοιότητα με αυτές τις χρονοσειρές. Προηγουμένως, αξίζει να σημειωθεί ότι η παραπάνω διαδικασία πραγματοποιείται παράλληλα, που σημαίνει ότι τα buckets εκχωρούνται σε διαφορετικές μονάδες επεξεργασίας, οι οποίες αναλαμβάνουν το φορτίο των συγκρίσεων των χρονοσειρών ανά ζεύγη.

2.8 Πρόβλεψη Μετοχών

Στη σύγχρονη εποχή, η μεγαλύτερη πρόκληση των συστημάτων, έγκειται στην αποτελεσματική αποθήκευση και ανάκτηση μεγάλου όγκου δεδομένων. Συχνά, είναι επιτακτική η

ανάγκη της επεξεργασίας και εξαγωγής χρήσιμων αποτελεσμάτων μεγάλου όγκου δεδομένων σε πραγματικό χρόνο, με απώτερο στόχο τη λήψη σημαντικών αποφάσεων. Ειδικότερα, στον χρηματιστηριακό τομέα, ζωτικής σημασίας είναι η πρόβλεψη της πορείας των μετοχών, η οποία στοχεύει στην όσο το δυνατό μειωμένη ανάληψη ρίσκου και κατ΄ επέκταση τη μεγιστοποίηση του κέρδους.

Για όλους τους παραπάνω λόγους, στην παρούσα διπλωματική αναπτύσσεται ο αλγόριθμος του Multiple Linear Regression, ο οποίος θα χρησιμοποιηθεί ως ένα μοντέλο πρόβλεψης.

2.8.1 Multiple Linear Regression

Τα μοντέλα Regression χρησιμοποιούνται για να περιγράψουν τη σχέση μεταξύ των εκάστοτε μεταβλητών προσαρμόζοντας μια γραμμή πάνω στα παρατηρούμενα δεδομένα. Γενικά, αυτά τα μοντέλα επιτρέπουν την εκτίμηση της μεταβολής της εξαρτημένης μεταβλητής, μεταβάλλοντας την (ή τις) ανεξάρτητες μεταβλητές.

Εν προκειμένω, το μοντέλο Multiple Linear Regression χρησιμοποιείται για να εκτιμήσει τη σχέση δύο ή περισσότερων ανεξάρτητων μεταβλητών και μιας εξαρτημένης. Στην πραγματικότητα χρησιμοποιείται όταν είναι επιθυμητό:

1. να δειχθεί πόσο ισχυρή είναι η σχέση μεταξύ δύο ή περισσότερων ανεξάρτητων μεταβλητών και μιας εξαρτημένης μεταβλητής,

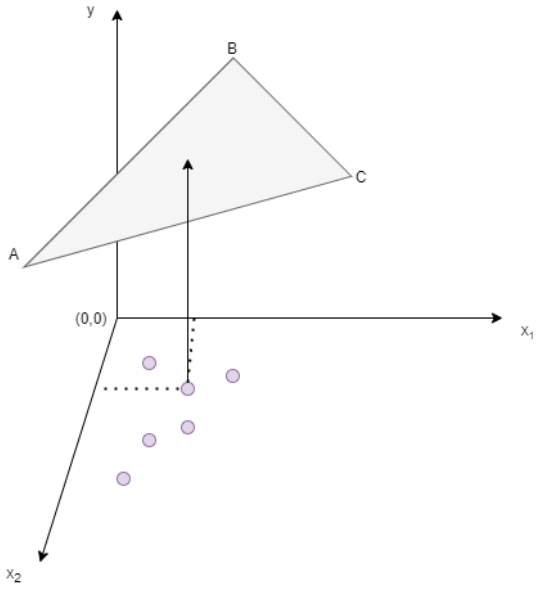
2. να δειχθεί η τιμή της εξαρτημένης μεταβλητής σε μια συγκεκριμένη τιμή των ανεξάρτητων μεταβλητών.

Στην περίπτωση της ύπαρξης πολλών ανεξάρτητων μεταβλητών δημιουργείται ένας χώρος, οι διαστάσεις του οποίου καθορίζονται από το πλήθος της εξαρτημένης και των ανεξάρτητων μεταβλητών. Επί παραδείγματι, στο Σχήμα 2.4 απεικονίζεται ένας τρισδιάστατος χώρος που απαρτίζεται από την εξαρτημένη μεταβλητή και από τις δύο ανεξάρτητες μεταβλητές. Πιο συγκεκριμένα, όπως φαίνεται, συνδυάζεται η αιτία x_1 και x_2, ώστε να παρθεί το σημείο που προκύπτει από τις διακεκομμένες γραμμές. Επομένως, με βάση αυτό το σημείο ενυπάρχει ακόμα ένα σημείο, το οποίο είναι το αποτέλεσμα της δράσης των x_1 και x_2 και βρίσκεται στο επίπεδο που ορίζεται από τα σημεία (ABC). Στην ουσία, ο στόχος είναι η εύρεση ενός υπερεπιπέδου στο χώρο με p διαστάσεις, που εφαρμόζει καλύτερα στα σημεία του χώρου αυτού.

2.8.2 Προϋποθέσεις μοντέλου

Πριν την εφαρμογή του μοντέλου είναι σημαντικό να ελεγχθούν οι παρακάτω προϋποθέσεις, ώστε να προκύψει ένα αξιόπιστο αποτέλεσμα στην πρόβλεψη.

1. Πολυσυγγραμμικότητα (multicollinearity): στο Multiple Linear Regression οι ανεξάρτητες μεταβλητές είναι πιθανό να έχουν μεγάλη συσχέτιση μεταξύ τους κι επομένως είναι απαραίτητο να ελεγχθεί αν συμβαίνει το φαινόμενο αυτό πριν την ανάπτυξη του μοντέλου. Σε αυτή την περίπτωση είναι δύσκολο να γίνει αντιληπτή η αξία της κάθε μεταβλητής, ώστε να ληφθεί υπόψη εκείνη που προκαλεί μεγαλύτερη επίδραση στην εξαρτημένη. Ο δείκτης ανεκτικότητας θα βοηθήσει για την επίλυση αυτού του προβλήματος, του οποίου οι τιμές πρέπει να είναι υψηλές.



Σχήμα 2.4: *Multiple Linear Regression*

2. Γραμμική σχέση: άλλη μια προϋπόθεση που πρέπει να ισχύει είναι η γραμμική σχέση μεταξύ ανεξάρτητων και εξαρτημένης μεταβλητής. Η γραμμικότητα είναι δυνατό να ελεγχθεί μέσω των scatterplots.

3. Ομοιόμορφη κατανομή καταλοίπων: η ανάλυση μέσω του μοντέλου Multiple Linear Regression απαιτεί την ομοιόμορφη κατανομή των σφαλμάτων, τα οποία προκύπτουν από τη διαφορά μεταξύ των παρατηρούμενων και των προβλεπόμενων τιμών. Αυτή η προϋπόθεση μπορεί να ελεγχθεί μέσω των ιστογραμμάτων.

4. Ομοσκεδαστικότητα: η τελευταία προϋπόθεση που πρέπει να ισχύει είναι η ύπαρξη της ομοσκεδαστικότητας των καταλοίπων, που σημαίνει ότι η διασπορά τους πρέπει να είναι σταθερή. Στην ουσία είναι επιθυμητό τα κατάλοιπα να είναι ομοιόμορφα κατανεμημένα, ώστε να μην παρουσιάζουν κάποιο μοτίβο.

Ο τύπος του Multiple Linear Regression είναι ο εξής:

$$y = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n \tag{2.11}$$

όπου y είναι η προβλεπόμενη τιμή της εξαρτημένης μεταβλητής, η τιμή β_0 είναι η τιμή του y όταν όλες οι παράμετροι τεθούν ίσοι με το μηδέν και $X_1, X_2, ..., X_n$ οι ανεξάρτητες μεταβλητές, β_i με $i = 1, 2, ..., k$ είναι η κλίση της επιφάνειας ως προς την αντίστοιχη ανεξάρτητη μεταβλητή X_i. Ως ε ορίζεται το κατάλοιπο (residual), δηλαδή η απόκλιση ενός συγκεκριμένο σημείου από το επίπεδο της παλινδρόμησης.

Η Εξίσωση ;; μπορεί να αναπαρασταθεί και με τη μορφή πινάκων. Αυτή η αναπαράσταση καθιστά δυνατή την χρήση της Γραμμικής Άλγεβρας, η οποία προσδίδει ευκολία στον τρόπο που υλοποιούνται οι διαδικασίες στο επίπεδο του κώδικα.

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad X = \begin{bmatrix} \mathbf{x}_0^\top \\ \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}, \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix} \tag{2.12}$$

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon \tag{2.13}$$

2.8.3 Εκτίμηση παραμέτρων $\beta_0, \beta_1, ..., \beta_{p-1}$

Το τυχαίο διάνυσμα ε αποτελείται από n ανεξάρτητες τυχαίες μεταβλητές που ακολουθούν την πολυδιάστατη κανονική κατανομή (Εξίσωση 2.14), δηλαδή έχουν την από κοινού συνάρτηση πυκνότητας πιθανότητας $\mathrm{N}\left(\mathbf{0}, \sigma^2\mathbf{I}_n\right)$, όπου $\mathbf{I}_n$ αναπαρίσταται ο μοναδιαίος πίνακας διάστασης n.

$$f\left(x_1, x_2, \ldots, x_k\right) = \frac{1}{(2\pi)^{k/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})}, \tag{2.14}$$

όπου,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix}, \boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_k \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{ik} \\ \sigma_{21} & \sigma_{22} & & \sigma_{2k} \\ \vdots & & & \vdots \\ \sigma_{k1} & \sigma_{k2} & \cdots & \sigma_{kk} \end{bmatrix} \tag{2.15}$$

Επομένως, το τυχαίο διάνυσμα $\mathbf{y}$ θα ακολουθεί την πολυδιάστατη κανονική κατανομή, $\mathrm{N}\left(\mathbf{Xb}, \sigma^2\mathbf{I}_n\right)$, με συνάρτηση πιθανοφάνειας:

$$L\left(\mathbf{b}, \sigma^2\right) = f\left(y_1, y_2, \ldots, y_n; \mathbf{b}, \sigma^2\right) = \frac{1}{(2\pi)^{n/2}\left(\sigma^2\right)^{n/2}} e^{-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{Xb})^T(\mathbf{y}-\mathbf{Xb})} \tag{2.16}$$

Η παραπάνω συνάρτηση μεγιστοποιείται ως προς το διάνυσμα $\mathbf{b}$, ώστε να εκτιμηθεί η μέγιστη πιθανοφάνεια, όταν ελαχιστοποιείται η παρακάτω ποσότητα:

$$(\mathbf{Y} - \mathbf{Xb})^T(\mathbf{Y} - \mathbf{Xb}) = \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} = \sum_{i=1}^{n} \varepsilon_i^2 \tag{2.17}$$

Αναλυτικότερα προκύπτει,

$$\left(\mathbf{Y}^T - \mathbf{b}^T\mathbf{X}^T\right)(\mathbf{Y} - \mathbf{Xb}) = \mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{Xb} - \mathbf{b}^T\mathbf{X}^T\mathbf{Y} + \mathbf{b}^T\mathbf{X}^T\mathbf{Xb} \tag{2.18}$$

και παραγωγίζοντας ως προς $\mathbf{b}$, η ποσότητα 2.18 γίνεται:

$$\frac{\mathrm{d}f}{\mathrm{d}\mathbf{b}} = \left(\frac{\partial f}{\partial b_0}, \ldots, \frac{\partial f}{\partial b_{p-1}}\right) \Rightarrow \tag{2.19}$$

$$\frac{\mathrm{d}}{\mathrm{db}}(\mathbf{Y}-\mathbf{Xb})^T(\mathbf{Y}-\mathbf{Xb}) = -2\mathbf{X}^T\mathbf{Y} + 2\mathbf{X}^T\mathbf{Xb} = 0 \Rightarrow \mathbf{X}^T\mathbf{Xb} = \mathbf{X}^T\mathbf{Y} \tag{2.20}$$

Το παραπάνω σύστημα εξισώσεων, με p αγνώστους τις τιμές του διανύσματος $\mathbf{b}$, έχει μοναδική λύση όταν υπάρχει ο αντίστροφος του $\mathbf{X}^T\mathbf{X}$ και επομένως προκύπτει ότι οι εκτιμήτριες μέγιστης πιθανοφάνειας για το διάνυσμα $\mathbf{b}$ θα είναι:

$$\hat{\mathbf{b}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y} \tag{2.21}$$

Όσον αφορά στον υπολογισμό των καταλοίπων ή εκτιμώμενων σφαλμάτων αρκεί να αποτυπωθούν οι διαφορές μεταξύ των παρατηρούμενων σε σχέση με τις προβλεπόμενες τιμές, δηλαδή:

$$\hat{\varepsilon}_i = Y_i - \hat{Y}_i, i = 1, 2, \ldots, n \tag{2.22}$$

ή με τη μορφή πινάκων,

$$\hat{\boldsymbol{\varepsilon}} = \mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{X}\hat{\mathbf{b}} = \mathbf{Y} - \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y} = \left(\mathbf{I}_n - \mathbf{P}\right)\mathbf{Y} \tag{2.23}$$

όπου,

$$\mathbf{P} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T \tag{2.24}$$

Ο πίνακας P είναι γνωστός ως πίνακας *ορθής προβολής*.

Κεφάλαιο 3

Χρησιμοποιούμενα Συστήματα

Στο κεφάλαιο αυτό περιγράφονται τα βασικά συστήματα-εργαλεία, η γνώση των οποίων είναι αναγκαία για την εκπόνηση της παρούσας διπλωματικής εργασίας.

3.1 Βασικά Συστήματα

3.1.1 SDE

Το SDE (Synopses Data Engine) είναι μια μηχανή παράλληλης επεξεργασίας και ανάλυσης δεδομένων μεγάλης κλίμακας [11]. Το SDE είναι χτισμένο πάνω στο Apache Flink και εφαρμόζει το παράδειγμα της σύνοψης (synopsis-as-a-service). Με αυτό τον τρόπο επιτυγχάνει την:

1. ταυτόχρονη διατήρηση χιλιάδων συνόψεων διαφόρων τύπων για χιλιάδες ροές δεδομένων κατ΄ απαίτηση,

2. επαναχρησιμοποίηση συντηρημένων συνόψεων μεταξύ διαφόρων ταυτόχρονων ροών εργασίας,

3. παροχή συνόψισης δεδομένων ακόμη και για (Big Data) ροές εργασίας,

4. δυνατότητα μεταφοράς νέων συνόψεων εν κινήσει,

5. βελτιστοποίηση εκτέλεσης ροής εργασίας.

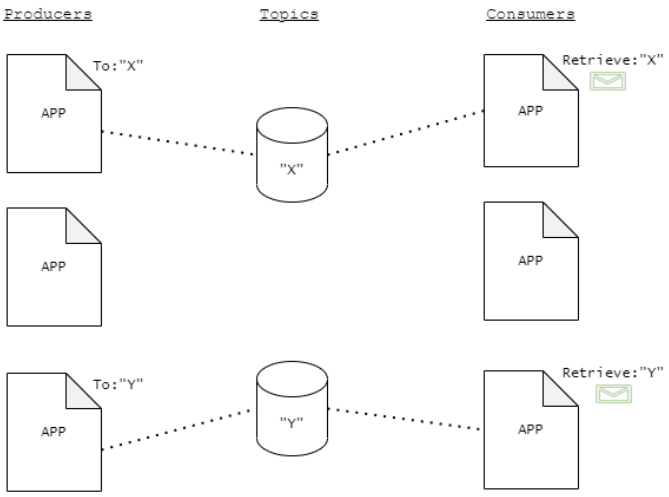
Το SDE είναι ιδανικό για ανάλυση δεδομένων μεγάλης κλίμακας, διότι επιτρέπει την οριζόντια επεκτασιμότητα, δηλαδή, όχι μόνο κλιμάκωση του υπολογισμού σε ορισμένες μονάδες επεξεργασίας, που είναι διαθέσιμες σε ένα σύμπλεγμα υπολογιστών, αλλά και αξιοποίηση του φορτίου επεξεργασίας που έχει εκχωρηθεί σε κάθε κόμβο. Επιπλέον, παρέχει κατακόρυφη επεκτασιμότητα, δηλαδή κλιμάκωση του υπολογισμού σε πολύ υψηλό αριθμό επεξεργασμένων ροών και ενοποιημένη επεκτασιμότητα, δηλαδή κλιμάκωση του υπολογισμού πέρα από single clusters και clouds, ελέγχοντας την επικοινωνία που απαιτείται για την απάντηση σε καθολικά ερωτήματα που τίθενται από δυνητικά γεωσπαρμένους clusters. Στην παρούσα εργασία, η σύνοψη που υλοποιείται, εντάσσεται στο SDE προκειμένου να επιτευχθεί η αποδοτικότερη και ταχύτερη απάντηση των εκάστοτε ερωτημάτων.

3.1.2 Apache Kafka

Το Apache Kafka συνιστά μια ανοιχτού κώδικα πλατφόρμα λογισμικού, η οποία αποσκοπεί στον αποδοτικό και άμεσο χειρισμό ροών δεδομένων σε πραγματικό χρόνο. Ουσιαστικά, πρόκειται για ένα κατανεμημένο σύστημα ανταλλαγής μηνυμάτων, το οποίο έχει σχεδιαστεί με στόχο τη μεταφορά δεδομένων σε μεγάλους όγκους. Το Apache Kafka είναι γραμμένο σε Scala και Java, ενώ αποτελεί δημιούργημα της εταιρείας που βρίσκεται πίσω από το κοινωνικό δίκτυο LinkedIn. Οι βασικές αρχές σχεδιασμού του Apache Kafka διαμορφώθηκαν με βάση την αυξανόμενη ανάγκη για αρχιτεκτονικές υψηλής απόδοσης, οι οποίες είναι εύκολα επεκτάσιμες και παρέχουν τη δυνατότητα αποθήκευσης, επεξεργασίας και επανεπεξεργασίας ροών δεδομένων.

Υποδομή

Πιο τεχνικά, το Apache Kafka βασίζεται σε ένα σύστημα δημοσίευσης-εγγραφής (publish-subscribe) μηνυμάτων. Στην πραγματικότητα, μόλις καταφθάνει ένα νέο μήνυμα, δηλαδή παράγεται από τον Producer, αποστέλλεται σε μια συγκεκριμένη τοποθεσία. Η τελευταία αναφέρεται ως *topic* και αποτελεί μια συλλογή ή ομάδα μηνυμάτων. Κάθε ένα topic χαρακτηρίζεται από ένα όνομα, το οποίο είναι μοναδικό και μπορεί να καθοριστεί εκ των προτέρων ή κατ΄ απαίτηση, εφόσον οι Producers γνωρίζουν το όνομα του topic και κατέχουν άδεια για να το αποστείλουν. Αξίζει να σημειωθεί ότι ένα topic διαθέτει ένα ή περισσότερα φυσικά αρχεία logs, τα οποία ονομάζονται *partitions* και βρίσκονται εντός κάποιου topic. Εν συνεχεία, οι Consumers αντλούν-καταναλώνουν τα μηνύματα του topic που τους ενδιαφέρει. Η παραπάνω διαδικασία μπορεί να αποτυπωθεί στο Σχήμα 3.1.



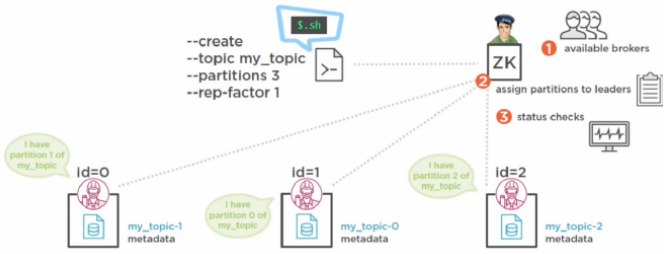
Σχήμα 3.1: *Apache Kafka as a Messaging System*

Τα μηνύματα, καθώς επίσης και τα αντίστοιχα topics τους, είναι αναγκαίο να διατηρηθούν σε κάποια τοποθεσία. Προκειμένου να καλυφθεί αυτή η ανάγκη εισάγεται η έννοια του *broker*. Στην πραγματικότητα, ένας broker είναι μια διεργασία λογισμικού, η οποία «τρέχει» σε ένα φυσικό ή εικονικό μηχάνημα, ενώ διαθέτει πρόσβαση στους πόρους του μηχανήματος, όπως για παράδειγμα το file system, το οποίο χρησιμοποιεί για να αποθηκεύσει μηνύματα που κατηγοριοποιεί ως topics.

Αρχιτεκτονική

Το Kafka cluster αποτελείται από έναν ή περισσότερους servers ή αλλιώς brokers, οι οποίοι «τρέχουν» τον Kafka. Το ρόλο του ενορχηστρωτή κατέχει ο Zookeeper, ο οποίος αποτελεί ένα κατανεμημένο σύστημα που στοχεύει στην αποτελεσματική διαχείριση των πόρων, προκειμένου να επιτευχθεί ένας κοινός στόχος.

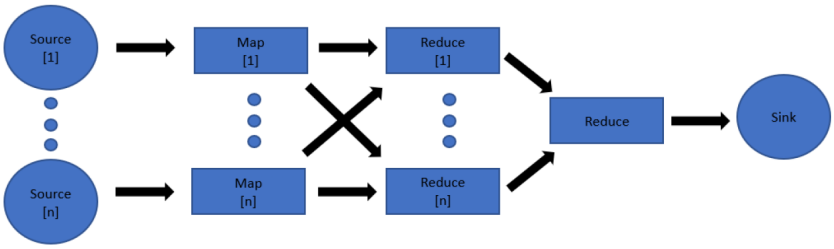
Όταν αποστέλλεται μια νέα εντολή δημιουργίας ενός topic, ο Zookeeper αναλαμβάνει την εκτέλεσή του. Συγκεκριμένα, αναζητά τους διαθέσιμους brokers και αποφασίζει ποιός από αυτούς είναι κατάλληλος για να διαχειριστεί ένα partition εντός κάποιου topic (Σχήμα 3.2). Μετά το πέρας αυτής της ανάθεσης, ο εκάστοτε Kafka broker δημιουργεί ένα αρχείο log, το οποίο αντιστοιχεί στο νέο partition. Όταν ο Producer είναι έτοιμος να δημοσιεύσει ένα νέο μήνυμα χρειάζεται να γνωρίζει τουλάχιστον έναν broker. Μόλις το μήνυμα σταλεί, ο Consumer θα καταναλώσει το μήνυμα.



Σχήμα 3.2: *Apache Zookeeper & Kafka*

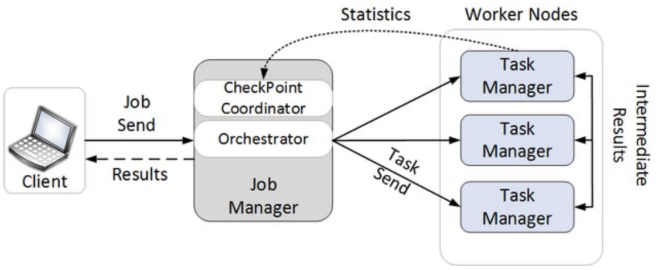
3.1.3 Apache Flink

Ο πυρήνας του Apache Flink είναι μια κατανεμημένη μηχανή που διαχειρίζεται τόσο bounded όσο και unbounded ροές δεδομένων [5]. Το Apache Flink είναι σχεδιασμένο προκειμένου να εκτελείται σε cluster περιβάλλοντα (δηλαδή ένα σύνολο από servers που συνδέονται μεταξύ τους, για την επίτευξη κοινού στόχου) και να επιτελεί υπολογισμούς με μεγάλη ταχύτητα και σε οποιαδήποτε κλίμακα. Γενικά, το Flink εφαρμόζει ένα μοντέλο ροής δεδομένων, στο οποίο τα δεδομένα ρέουν συνεχώς μέσω ενός δικτύου οντοτήτων μετασχηματισμού, δημιουργώντας ένα κατευθυνόμενο γράφημα, τις περισσότερες φορές ένα DAG (κατευθυνόμενο ακυκλικό γράφημα), που μπορεί να εκτελεστεί από ένα μόνον «αγωγό» ή από ένα κατανεμημένο παράλληλο σύστημα.



Σχήμα 3.3: *Flink Map-Reduce Data-flow*

Η παραπάνω ροή δεδομένων (Σχήμα 3.3) αποστέλλεται στον cluster του Flink, προκειμένου να εκτελεστεί. Ο cluster του Flink αποτελείται από ένα Master κόμβο (τουλάχιστον ένας) και από έναν αριθμό από Worker κόμβους. Ο κόμβος Master ενορχηστρώνει την εκτέλεση μιας κατανεμημένης εργασίας (JobManager), ενώ οι Worker κόμβοι αναλαμβάνουν την εκτέλεσής της (TaskManager), όπως απεικονίζεται στο Σχήμα 3.4. Κάθε Worker κόμβος παρέχει έναν αριθμό από διαθέσιμους φυσικούς πόρους, οι οποίοι θα αναλάβουν να εκτελέσουν την εκάστοτε εργασία. Οι Workers εκτελούν ένα στιγμιότυπο του κώδικα, με τη διαφορά ότι λαμβάνουν στην είσοδό τους διαφορετικά τμήματα των δεδομένων. Αξίζει να σημειωθεί ότι το Flink διαθέτει DataStream API για τη διαχείριση των ρευμάτων δεδομένων, τα οποία δημιουργούνται για παράδειγμα μέσω του Kafka. Το Flink παρέχει υψηλή απόδοση, ταχύτητα και ανοχή σε σφάλματα.



Σχήμα 3.4: *Architecture of Flink system*

3.2 Σχετικές εργασίες

Η πρόβλεψη των χρηματιστηριακών μετοχών αποτελεί ένα πεδίο πάνω στο οποίο έχουν πραγματοποιηθεί αρκετές προσπάθειες, ώστε να επιτευχθεί η ζητούμενη ανάγκη και να καθοριστεί η μελλοντική πορεία των μετοχών, η οποία είναι ιδιαίτερα χρήσιμη για τους επενδυτές.

Σε πρώτη φάση η παρούσα εργασία βασίζεται στην εκτίμηση της συσχέτισης των μετοχών, μέσω του Pearson Correlation, όπως αυτή αναλύεται στο StatStream, [9]. Στη συγκεκριμένη περίπτωση το StatStream έχει σχεδιαστεί για κεντρικοποιημένη εκτέλεση, ενώ η συμβολή της παρούσας διπλωματικής έγκειται στην προσπάθεια της εφαρμογής του εκτιμητή σε ένα κλιμακούμενο και παράλληλο σύστημα, το οποίο παρέχει τη δυνατότητα υιοθέτησης διαφορετικών σεναρίων ανάλογα με τις απαιτήσεις του χρήστη. Η προσπάθεια αυτή επιτυγχάνεται μέσω της ανάπτυξης της σύνοψης στο σύστημα του SDE, [11], στο οποίο προηγουμένως αναπτύσσεται ο DFT με τη διαφορά ότι δεν λαμβάνονται υπόψη οι παρελθοντικές τιμές και ταυτόχρονα δεν πραγματοποιείται η πρόβλεψη των μετοχών. Συγχρόνως, η διατριβή [8] εισάγει αποδοτικούς αλγόριθμους και αρχιτεκτονικές για την αντιμετώπιση του προβλήματος της παρακολούθησης της συσχέτισης χιλιάδων ροών δεδομένων (ύπαρξη συσχέτισης ανά ζεύγη) και εισάγει το framework, T-Storm, το οποίο μπορεί να χρησιμοποιηθεί για να γίνεται εύκολα και αποδοτικά η κλιμάκωση και ανάπτυξη εφαρμογών ανάλυσης ροών δεδομένων μεγάλης κλίμακας.

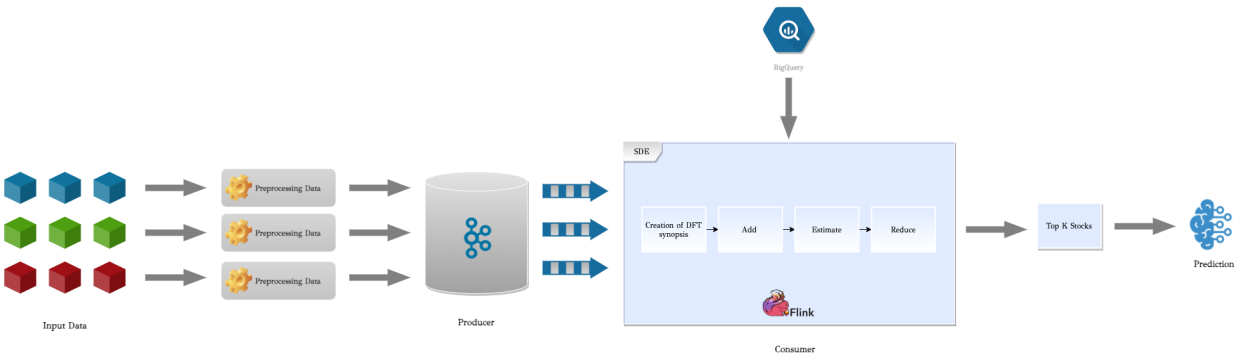
Κεφάλαιο 4

Υλοποίηση

Στο κεφάλαιο αυτό περιγράφεται η διασύνδεση των επιμέρους υποσυστημάτων που χρησιμοποιούνται, καθώς και η υλοποίηση της σύνοψης. Αρχικά, το σύστημα της Εικόνας 4.1 αποδομείται και αναλύεται περαιτέρω, προκειμένου να γίνουν αντιληπτά όλα τα στάδια της υλοποίησης.

4.1 Αρχιτεκτονική Συστήματος

Σε πρώτο επίπεδο, το Σχήμα 4.1 απεικονίζει την πορεία που ακολουθούν τα χρηματιστηριακά δεδομένα, καθώς και τον τρόπο επεξεργασίας τους, προκειμένου να αναλυθούν και να εξαχθούν τα επιθυμητά αποτελέσματα.



Σχήμα 4.1: *Αρχιτεκτονική Συστήματος*

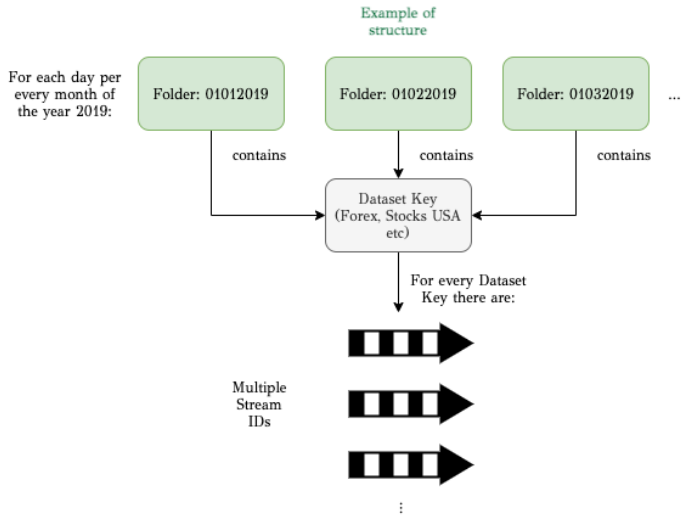
4.1.1 Dataset

Τα χρηματιστηριακά δεδομένα που χρησιμοποιούνται ως είσοδος στο σύστημα του Kafka, αντλούνται από την πηγή: [12] και αφορούν μετρήσεις των τιμών των μετοχών για το έτος 2019. Τα δεδομένα αυτά ακολουθούν μια συγκεκριμένη δομή, η οποία αναλύεται ακολούθως.

Δομή

Το dataset συνίσταται από τριακόσιους εξήντα πέντε (365) φακέλους, καθένας από τους οποίους αντιπροσωπεύει και μια μέρα του χρόνου. Πιο συγκεκριμένα, κάθε φάκελος εμπεριέχει

μετρήσεις διαφορετικών μετοχών για μια συγκεκριμένη μέρα. Το Σχήμα 4.2 αποσαφηνίζει τη δομή των δεδομένων.



Σχήμα 4.2: *Δομή των δεδομένων*

Αναλυτικότερα, οι μετρήσεις καταγράφονται σε αρχεία της μορφής .txt, το όνομα των οποίων καταγράφεται ως το όνομα της εκάστοτε μετοχής (StreamID). Το αρχείο αυτό διατηρεί πληροφορίες που σχετίζονται με την ημερομηνία, την ώρα (ανά δευτερόλεπτο), την τιμή και τον όγκο των συναλλαγών (ο αριθμός των μετοχών που διακινούνται σε μια συγκεκριμένη χρονική περίοδο) των μετοχών. Η Εικόνα 4.3 απεικονίζει ένα παράδειγμα του τρόπου δόμησης ενός αρχείου μιας συγκεκριμένης μετοχής για μια συγκεκριμένη μέρα ενός μήνα του έτους. Αξίζει να σημειωθεί ότι το πλήθος των μετρήσεων και ταυτόχρονα το πλήθος των χρονικών στιγμών είναι διαφορετικό για κάθε μετοχή. Αυτό σημαίνει ότι τα δεδομένα απαιτούν μια προ-επεξεργασία, προτού σταλούν προς ανάλυση, με στόχο την εξαγωγή αποτελεσμάτων.

```
01/09/2019,00:00:01,.94755,1
01/09/2019,00:00:01,.9476,1
01/09/2019,00:00:01,.94761,1
01/09/2019,00:00:02,.94761,1
01/09/2019,00:00:02,.94761,1
```

Σχήμα 4.3: *Παράδειγμα αρχείου της μετοχής "Forex·AUDCAD·NoExpiry"*

Προεπεξεργασία

Παρατηρώντας ένα αρχείο δεδομένων είναι δυνατόν να βρεθούν μετοχές στις οποίες εκλείπουν κάποιες χρονικές στιγμές ή για την ίδια χρονική στιγμή είναι πιθανό να έχουν καταγραφεί πολλαπλές χρονικές στιγμές, όπως φαίνεται στο Σχήμα 4.3. Προκειμένου να αντιμετωπιστούν αυτά τα πιθανά προβλήματα, ακολουθείται η εξής πολιτική:

- *για τα δεδομένα που εκλείπουν*: λαμβάνεται υπόψιν η τιμή της τελευταίας χρονικής στιγμής που έχει ληφθεί και με βάση αυτή τη τιμή «γεμίζουν» οι αντίστοιχες εκλειπόμενες.
- *για τα δεδομένα που έχουν καταγραφεί πολλαπλές τιμές ανά λεπτό*: λαμβάνεται υπόψιν η τιμή της τελευταίας χρονικής στιγμής, για την οποία έχει ληφθεί μέτρηση.

Μέρος της προ-επεξεργασίας των δεδομένων αποτελεί και η τροποποίηση του τρέχοντος dataset, ώστε να προκύψει ένα πιο δομημένο, που θα επιφέρει καλύτερα αποτελέσματα. Πιο συγκεκριμένα, κατά την εισαγωγή των δεδομένων στο topic του Kafka στην ουσία εφαρμόζεται ένα φίλτρο πάνω σε αυτά. Ειδικότερα, για κάθε ένα αρχείο μιας μετοχής κάθε ημέρας του χρόνου για κάθε μια τιμή της μετοχής διατηρούνται μόνο οι τιμές των μετοχών οι οποίες διαιρούνται ακριβώς με το 10. Αμέσως μετά, όταν τα δεδομένα διαβάζονται από το topic του Kafka, ώστε να εισαχθούν στη σύνοψη, εφαρμόζεται η παραπάνω πολιτική και προκύπτουν οι τιμές των μετοχών για κάθε λεπτό.

4.2 Λεπτομέρειες υλοποίησης

4.2.1 Kafka & SDE

Ακολουθώντας την πορεία του Σχήματος 4.1, τα επεξεργασμένα δεδομένα αποστέλλονται σε κάποιο topic του Kafka, από το οποίο καταναλώνονται μέσω του συστήματος SDE. Παράλληλα, αποστέλλονται αιτήματα εισαγωγής δεδομένων στο SDE, προκειμένου να ληφθούν τα δεδομένα που θα εισαχθούν στη δημιουργηθείσα σύνοψη. Πιο συγκεκριμένα, δημιουργείται ένα αίτημα, όπως αποτυπώνεται στη γραμμή κώδικα του πλαισίου παρακάτω, το οποίο δέχεται ως είσοδο:

1. key: το κλειδί-όνομα του topic στο Kafka, από το οποίο αντλούνται τα δεδομένα,
2. requestID: το είδος του αιτήματος (add, delete, estimate),
3. synopsisID: ο αναγνωριστικός αριθμός της σύνοψης, όπου στην περίπτωση αυτή είναι το 29,
4. uID: ο αναγνωριστικός αριθμός κάθε αιτήματος,
5. streamID: ο αναγνωριστικός αριθμός του εκάστοτε stream,
6. parameters: περιλαμβάνουν το όνομα της εκάστοτε μετοχής, τη τιμή της, τη χρονική στιγμή της που πραγματοποιείται η μέτρηση, τον όγκο των συναλλαγών, το μήκος του basic window DFT, το μήκος του sliding window DFT, το πλήθος των DFT συντελεστών που λαμβάνονται υπόψη, καθώς επίσης και το sliding window που πραγματοποιείται στην εισαγωγή δεδομένων,
7. noOfP: αριθμός παραλληλισμού του συστήματος.

Επιπλέον, είναι σημαντικό να αναφερθεί ότι το σύστημα του SDE είναι δυνατό να εκτελέσει την πράξη του add, της εισαγωγής δεδομένων στη σύνοψη και την πράξη του estimate, δηλαδή την πράξη της εύρεσης των συσχετίσεων. Το requestID για την πράξη του add λαμβάνει την τιμή 1, ενώ για την πράξη του estimate την τιμή 3.

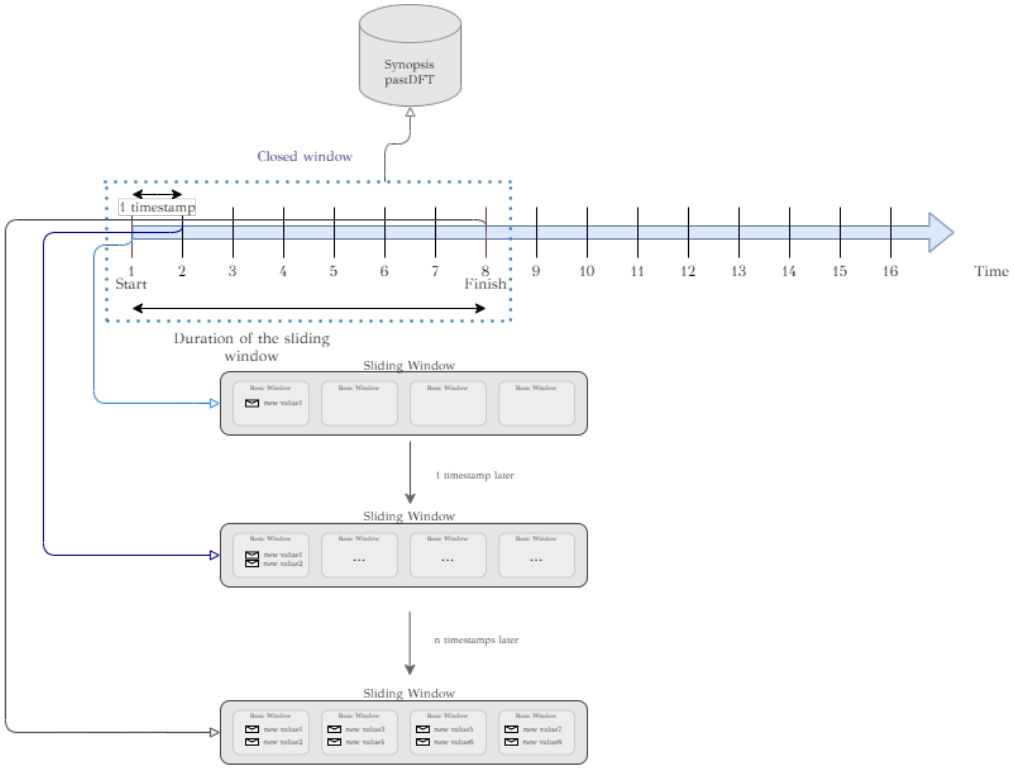
```
Request rq = new Request(key, requestID, synopsisID, uID, streamID, parameters, noOfP);
```

Από τη στιγμή που λαμβάνονται τα εκάστοτε δεδομένα, ακολουθεί η διαδικασία του add. Παράλληλα, μόλις ληφθεί ένα αντίστοιχο αίτημα για την εκτέλεση του estimate και κατ' επέκταση του reduce, λαμβάνονται στην έξοδο τα αντίστοιχα αποτελέσματα με τα k πιο όμοια

ζευγάρια. Το αίτημα που αφορά στην εκτέλεση του estimate ακολουθεί τη δομή του αιτήματος, όπως περιγράφεται παραπάνω, με τη διαφορά ότι οι παράμετροι είναι τροποποιημένοι. Αναλυτικότερα, ορίζεται το threshold, ο αριθμός του k (εξαγωγή πιο όμοιων ζευγαριών), το T ως η μέγιστη χρονική απόσταση μεταξύ δύο μετοχών, η λίστα των επιθυμητών μετοχών για τις οποίες αναζητείται ομοιότητα, καθώς και ο αντιπροσωπευτικός αριθμός του ερωτήματος. Ο τρόπος υλοποίησης αυτών των διαδικασιών αναλύεται παρακάτω λεπτομερώς.

4.2.2 Add

Προκειμένου να πραγματοποιηθεί η εισαγωγή δεδομένων στη σύνοψη εφαρμόζεται ο Αλγόριθμος 4.1. Συγκεκριμένα, για κάθε μία μετοχή, δημιουργείται μια λίστα, η οποία αποθηκεύει στιγμιότυπα της κλάσης COEF. Η τελευταία διατηρεί όλη την χρήσιμη πληροφορία που σχετίζεται με τους κανονικοποιημένους DFTs, την αρχή του εκάστοτε παραθύρου μιας συγκεκριμένης μετοχής, καθώς επίσης τις πραγματικές τιμές που αντιστοιχούν σε αυτό το χρονικό παράθυρο, το χαρακτηριστικό όνομα της μετοχής, το μοναδικό bucketID, καθώς και το πλήθος των γειτονικών κελιών στο πλέγμα για την εκάστοτε μετοχή.



Σχήμα 4.4: *Δομή των Sliding Windows*

Η λογική με την οποία πραγματοποιείται το γέμισμα των sliding windows είναι η εξής: για μια νέα τιμή μιας μετοχής γεμίζει αρχικά το basic window που εμπεριέχεται στο sliding window. Αν για παράδειγμα, το μήκος του sliding window είναι 8 (οκτώ) και αντίστοιχα το μήκος του basic window είναι 2 (δύο) αυτό σημαίνει ότι πρέπει να συμπληρωθούν 4 (τέσσερα) basic windows. Μετά το πέρας αυτών των χρονικών στιγμών, μετρημένων σε λεπτά, με το που εισάγεται η πρώτη τιμή του νέου basic window που δεν ανήκει στο τρέχον sliding window, το πρώτο basic window του γεμισμένου sliding window, απορρίπτεται, ώστε να συμπληρωθεί το

νέο και να ληφθεί υπόψη στον υπολογισμό των κανονικοποιημένων τιμών του DFT, Σχήμα 4.4. Παράλληλα, εκτός από τη συμπλήρωση των παραθύρων, τα οποία σχετίζονται με τον DFT, λειτουργεί ένα επιπλέον εξωτερικό χρονικό παράθυρο, το οποίο ρυθμίζει ανά πόσες χρονικές στιγμές θα λαμβάνονται υπόψη οι δημιουργημένοι DFTs, οι οποίοι θα εισάγονται σε μια λίστα. Η λίστα αυτή δημιουργείται για καθεμιά μετοχή ξεχωριστά. Αξίζει να σημειωθεί ότι το μήκος του εξωτερικού sliding window διατηρείται μικρότερο του μήκους του sliding window DFT και αποτελεί μια παράμετρο του συστήματος, με βάση την οποία εξάγονται ορισμένα συμπεράσματα, τα οποία αναλύονται σε επόμενη ενότητα.

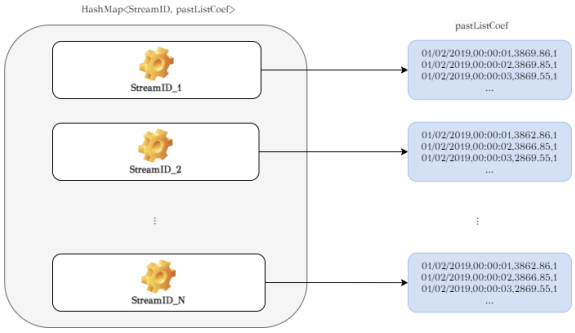
Για κάθε μια νέα χρονική στιγμή μιας μετοχής υπολογίζεται η χρονική της απόσταση από την προηγούμενη στιγμή της (Γραμμή 2 Αλγόριθμος 4.1). Ανάλογα με το μέγεθος αυτής της απόστασης, πραγματοποιούνται και οι αντίστοιχες χρονικές προσθήκες, όπως έχει αναλυθεί παραπάνω στην προ-επεξεργασία των δεδομένων. Αφού πραγματοποιηθεί το γέμισμα του sliding window, υπολογίζονται οι κανονικοποιημένοι DFTs, για το συγκεκριμένο παράθυρο και εν συνεχεία αποθηκεύονται ως μια νέα μεταβλητή στο αντικείμενο της δημιουργηθείσας κλάσης COEF. Συγχρόνως, παράγεται ένα Treemap με όλες τις φιλτραρισμένες χρονικές στιγμές, το οποίο περιέχει τις αντίστοιχες τιμές των μετοχών για κάθε χρονική στιγμή.

Αλγοριθμος 4.1: *Εισαγωγή δεδομένων στη σύνοψη*

Είσοδος: X (*ημερομηνία και ώρα μετοχής*)
Έξοδος: Y (*συνδεδεμένη λίστα με αντικείμενα της κλάσης COEF*)
  $currentDate \leftarrow$ τωρινή τοπική ώρα
  $diff \leftarrow currentDate - lastDate$
  for $i \leftarrow 1$ μέχρι $diff$ **do**
    Γέμισμα του DFT παραθύρου με την τελευταία τιμή που έχει εισαχθεί
    if έχει γεμίσει το παράθυρο εισαγωγής δεδομένων **then**
      Δημιουργία ενός αντικειμένου της κλάσης COEF
      Εισαγωγή του αντικειμένου της κλάσης COEF στη συνδεδεμένη λίστα
    end if
  end for
  $lastDate \leftarrow currentDate$

4.2.3 Estimate

Σε ένα Hashmap κρατείται ως το κλειδί το όνομα της νεοεισερχόμενη μετοχής και ως η τιμή της ένα στιγμιότυπο της κλάσης pastListCoef, η οποία εμπεριέχει όλη την πληροφορία για την εκάστοτε χρονοσειρά, όπως φαίνεται και στο Σχήμα 4.5. Τη στιγμή που καλείται η μέθοδος του estimate, αρχικοποιείται ένα νέο Hashmap, το οποίο εμπεριέχει τόσα κλειδιά όσα και ο εκάστοτε αριθμός των buckets που προκύπτει. Για καθεμιά μετοχή διατρέχεται η λίστα που διατηρεί την πληροφορία των coefficients και καθένα παράθυρο της μετοχής αποστέλλεται σε ένα bucket, σύμφωνα με ένα συγκεκριμένο bucketID. Η πολιτική, με την οποία κάθε χρονικό παράθυρο, γίνεται hashed σε ένα bucket, καθώς και η αποστολή του παραθύρου στα αντίστοιχα γειτονικά buckets, αναλύεται παρακάτω λεπτομερώς. Με αυτό τον τρόπο πραγματοποιείται η προσθήκη των εκάστοτε χρονικών παραθύρων στα buckets του Hashmap, το οποίο εν τέλει επιστρέφεται από τη μέθοδο estimate.

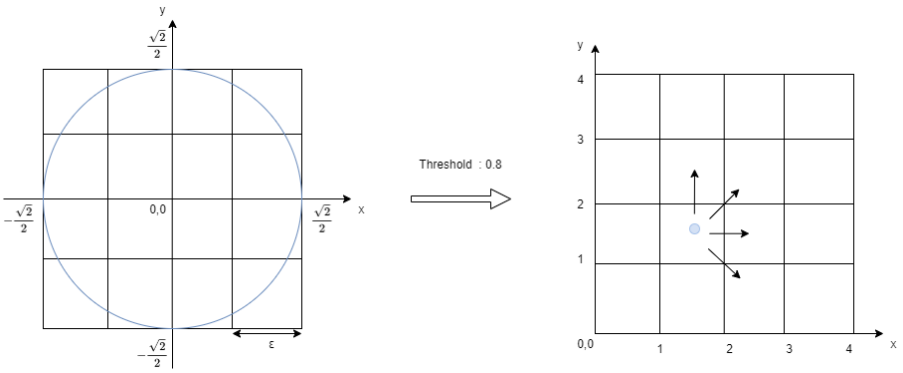


Σχήμα 4.5: *Δομή της λειτουργίας add*

Grid & Hashing

Σύμφωνα με την πηγή [9] και όπως έχει περιγραφεί στο θεωρητικό υπόβαθρο, η νόρμα των DFT συντελεστών ορίζεται από το $-\frac{\sqrt{2}}{2}$ έως το $\frac{\sqrt{2}}{2}$. Επιλέγοντας, έναν από τους οκτώ διαθέσιμους συντελεστές, για την εξαγωγή της χρήσιμης πληροφορίας, δημιουργείται ένα δισ-διάστατο πλέγμα, όπως αυτό φαίνεται στο Σχήμα 4.6. Στην πραγματικότητα, το ευθύγραμμο τμήμα μεταξύ των σημείων του $-\frac{\sqrt{2}}{2}$ έως το $\frac{\sqrt{2}}{2}$, τόσο στον άξονα $x'x$ όσο και στον άξονα $y'y$, διαιρείται σε υποτμήματα, το μήκος των οποίων καθορίζεται από την τιμή του ε όπως φαίνεται στο αριστερό τμήμα του Σχήματος 4.6.

Προκειμένου, να παραχθεί το bucketID, με βάση το οποίο μια χρονοσειρά θα γίνει hashed σε κάποιο κελί του πλέγματος, χρειάζεται να μετατοπιστεί το πλέγμα δεξιότερα και προς τα πάνω, ώστε να δοθούν θετικές τιμές στο κλειδί. Πιο συγκεκριμένα, πραγματοποιείται μια μετατόπιση κατά $\frac{\sqrt{2}}{2}$ και στους δύο άξονες. Επιπλέον, το εύρος των τιμών του bucketID, ποικίλλει ανάλογα με την τιμή του threshold.



Σχήμα 4.6: *Δομή του Grid*

Στο παράδειγμα της παραπάνω εικόνας έχει θεωρηθεί ότι η τιμή του threshold είναι ίση με 0.8. Αυτό έχει άμεση συνέπεια να δημιουργηθεί ένα πλέγμα με 16 (δεκαέξι) κελιά, σύμφωνα με τον τύπο, όπως αυτός έχει δοθεί, στη Ενότητα 2. Για κάθε ένα DFT συντελεστή, λαμβάνεται υπόψη τόσο το πραγματικό όσο και το φανταστικό μέρος του, στα οποία προστίθενται το offset $\frac{\sqrt{2}}{2}$. Εν συνεχεία, οι παραγόμενοι αριθμοί διαιρούνται με το ε και στο τέλος λαμβάνεται το άνω όριο για καθένα από τους δύο αυτούς αριθμούς. Μέσω αυτής της διαδικασίας, παράγονται οι συντεταγμένες για τον εκάστοτε μιγαδικό, οι οποίες θα αντιστοιχηθούν σε κάποιο από τα

κελιά του πλέγματος. Η αντιστοίχιση αυτή, πραγματοποιείται μέσω της συνάρτησης:

$$(y-1) * maxX + x$$

όπου τα x,y είναι οι συντεταγμένες που έχουν παραχθεί και η τιμή του maxX ορίζεται ως το μέγιστο πλήθος των στηλών του πλέγματος.

Επιπλέον, για κάθε ένα κελί του πλέγματος διατηρείται και μια λίστα με τα γειτονικά κελιά, με σκοπό την αποστολή της πληροφορίας σε αυτά. Πιο συγκεκριμένα, κάθε bucket αποτελείται από δύο λίστες, οι οποίες είναι οι εξής: η native και η neighbors. Στην πρώτη αποθηκεύονται τα χρονικά παράθυρα τα οποία γίνονται όντως hashed στο εκάστοτε κελί, ενώ στη δεύτερη τα χρονικά παράθυρα των γειτονικών κελιών. Τα χρονικά παράθυρα αποστέλλονται στα γειτονικά κελιά με βάση μια συγκεκριμένη πολιτική, η οποία είναι η εξής: τα χρονικά παράθυρα αποστέλλονται στα πάνω, στα πάνω δεξιά, στα δεξιά και στα κάτω δεξιά κελιά, όπως φαίνεται και στη δεξιά πλευρά του Σχήματος 4.6. Με βάση αυτή την πολιτική διατηρούνται τα μισά pairs, καθώς οι συμμετρικές σχέσεις δεν λαμβάνονται υπόψη (για παράδειγμα στη σχέση κελιών 1-2 και 2-1, λαμβάνεται υπόψη μόνο η 1-2), για τις δύο διαστάσεις που έχουν επιλεχθεί.

Αλγοριθμος 4.2: *Αλγόριθμος δημιουργίας και γεμίσματος των buckets*

Είσοδος: X (*Threshold*)
Έξοδος: Y (*HashMap το οποίο περιέχει τα buckets με τους DFTs*)
  Δημιουργία του Grid
  Αρχικοποίηση των buckets
  for Για κάθε entry του HashMap που περιέχει τις χρονοσειρές(pastListCoef) **do**
    for Για κάθε ένα στοιχείο COEF **do**
      Υπολόγισε το bucketID
      Στείλε το χρονικό παράθυρο στους γείτονες και στο κελί που ανήκεις
    end for
  end for

Όπως έχει περιγραφεί στην Ενότητα Grid & Hashing, δημιουργείται το πλέγμα και αμέσως μετά αρχικοποιούνται τα buckets, στα οποία θα κατακερματιστούν τα αντίστοιχα παράθυρα της εκάστοτε μετοχής (Γραμμές 2, 3 του Αλγορίθμου 4.2). Με βάση τον Αλγόριθμο 4.2, για κάθε μία μετοχή, δηλαδή για κάθε μία χρονοσειρά του πεδίου value του Hashmap που διατηρεί όλες τις χρονοσειρές των μετοχών, λαμβάνεται κάθε ένα στοιχείο της χρονοσειράς, για το οποίο υπολογίζεται το αντίστοιχο bucketID, με βάση το οποίο θα πραγματοποιηθεί ο κατακερματισμός στο bucket που ανήκει.

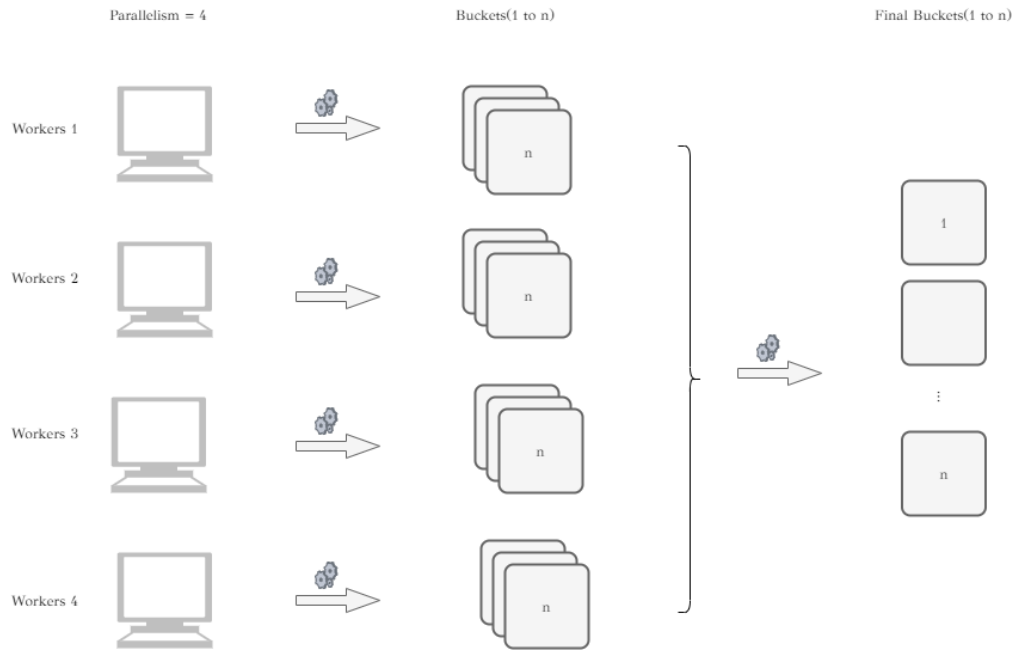
Workers & Buckets

Προκειμένου να πραγματοποιηθεί η διαδικασία της δημιουργίας και του γεμίσματος των buckets, χρειάζεται να αποσταλεί ένα αίτημα στο SDE, όπως ακριβώς διατυπώνεται παραπάνω για τη διαδικασία της εισαγωγής δεδομένων στη σύνοψη. Η διαφορά ανάγεται στο γεγονός ότι οι παράμετροι που εισάγονται είναι οι εξής: δίδεται η τιμή του threshold, η τιμή του k (ο αριθμός αυτός σχετίζεται με τις k πιο όμοιες μετοχές που προκύπτουν), η τιμή του T (ο αριθμός αυτός υποδηλώνει το μέγιστο χρονικό διάστημα που μπορεί να απέχουν δύο χρονικά

παράθυρα), η λίστα με τις μετοχές για τις οποίες είναι επιθυμητή η ομοιότητα, καθώς και ένας χαρακτηριστικός αριθμός, ο οποίος υποδηλώνει το query που επιθυμεί να εκτελέσει ο χρήστης κάθε φορά.

4.2.4 Reduce

Στο Σχήμα 4.7 πραγματοποιείται η υπόθεση ότι το σύστημα λειτουργεί με παραλληλισμό 4, γεγονός που σημαίνει ότι τέσσερα διαφορετικά φυσικά μηχανήματα αναλαμβάνουν να εκτελέσουν τον κώδικα της δημιουργηθείσας σύνοψης. Πιο συγκεκριμένα, σε κάθε ένα από τα μηχανήματα εισάγονται τα δεδομένα στη σύνοψη (λειτουργία της διαδικασίας add) και αμέσως μετά εκτελείται η διαδικασία της εύρεσης των συσχετίσεων μεταξύ των μετοχών (λειτουργία της διαδικασίας της estimate). Αφού έχουν σταλεί τα δεδομένα - τα αντίστοιχα χρονικά παράθυρα κάθε μετοχής - στα buckets, σειρά έχει η συνένωση των λιστών native και neighbors από κάθε ένα worker. Αναλυτικότερα, τα buckets με ID ίσο με 1 έως n ενώνουν τις λίστες τους (native & neighbors), με αποτέλεσμα να προκύπτουν τα τελικά n buckets, στα οποία εμπεριέχεται όλη η χρήσιμη πληροφορία, πάνω στην οποία θα πραγματοποιηθεί και το ερώτημα σε επόμενη φάση. Δηλαδή, αν $n = 4$ σημαίνει ότι κάθε worker παράγει 4 διαφορετικά buckets, γεγονός που σημαίνει ότι για κάθε bucket υπάρχουν 4 διαφορετικές λίστες native & neighbors, οι οποίες πρέπει να συνενωθούν, ώστε να προκύψουν οι τελικές 4 λίστες που θα περιέχουν όλοι την πληροφορία. Η διαδικασία αυτή συνιστά τη λειτουργία του Reduce.



Σχήμα 4.7: *Δομή της διαδικασίας του Reduce*

Query

Προκειμένου να βρεθεί η ομοιότητα των μετοχών χρειάζεται να δοθεί στην είσοδο η λίστα των επιθυμητών μετοχών, οι οποίες θα προβλεφθούν. Αυτή η λίστα δίνεται ως είσοδος στον Αλγόριθμο 4.3, με βάση τον οποίο αναζητείται το τελευταίο χρονικό παράθυρο που αντιστοιχεί

σε καθεμία μετοχή που βρίσκεται στη λίστα εισόδου. Αφού πραγματοποιηθεί η αναζήτηση επιστρέφεται ένα Hashmap το οποίο περιέχει ως κλειδί το όνομα κάθε μετοχής της λίστας και ως τιμή το τελευταίο χρονικό παράθυρο.

Αλγοριθμος 4.3: *Αναζήτηση χρονικών παραθύρων*

Είσοδος: *X* (*Λίστα με τις μετοχές προς αναζήτηση*)
Έξοδος: *Y* (*HashMap: key ->μετοχή, value -> τελευταίο παράθυρο*)
for Για κάθε μία μετοχή της λίστας εισόδου **do**
for Για κάθε ένα στοιχείο COEF της native λίστας **do**
if Το όνομα της μετοχής ταυτίζεται με αυτό της λίστας **then**
if Το τρέχον παράθυρο που ελέγχεται είναι το τελευταίο **then**
Πρόσθεσε στο χρονικό παράθυρο που έχει βρεθεί στο αντίστοιχο κλειδί
end if
end if
end for
end for

Εν συνεχεία, το τελευταίο χρονικό παράθυρο που έχουν αναζητηθεί και βρεθεί, συγκρίνεται με τα αντίστοιχα χρονικά παράθυρα που υπάρχουν είτε στη native είτε στη neighbors λίστα. Στον Αλγόριθμο 4.4, στη Γραμμή 1 καλείται ο Αλγόριθμος 4.3, από τον οποίο προκύπτει το Hashmap, που περιέχει για κάθε μια μετοχή της λίστας εισόδου, το τελευταίο χρονικό παράθυρο. Εν συνεχεία, για κάθε μία τιμή αυτού του Hashmap (Γραμμή 2, Αλγόριθμος 4.4) και ταυτόχρονα για κάθε τιμή της native λίστας ελέγχεται εάν τα ονόματα των μετοχών δεν είναι τα ίδια και ταυτόχρονα, η χρονική διαφορά των χρονικών παραθύρων τους δεν υπερβαίνει την τιμή της παραμέτρου T. Στην ουσία στο σημείο αυτό πραγματοποιείται ένα είδος φιλτραρίσματος μιας και η παράμετρος αυτή αντιπροσωπεύει το χρονικό διάστημα, το οποίο θα καθορίσει το πόσο παλιά θα αναζητηθούν τα χρονικά παράθυρα που θα χρησιμοποιηθούν για την εύρεση της ομοιότητας. Στην περίπτωση που ικανοποιείται η συνθήκη αυτή υπολογίζεται το correlation. Σε περίπτωση, που ικανοποιηθεί η συνθήκη και το correlation ξεπερνά την τιμή του threshold και ταυτόχρονα το τελευταίο δεν υπερβαίνει την τιμή του ενός, δημιουργείται ένα αντικείμενο της κλάσης PAIR, το οποίο αρχικοποιείται και εκχωρείται σε ένα Hashmap. Με αυτό τον τρόπο διατηρούνται όλα τα correlated pairs για κάθε μια μετοχή ξεχωριστά. Από αυτά, σύμφωνα με την τιμή του k, λαμβάνονται υπόψη τα πρώτα k πιο όμοια ζευγάρια. Αμέσως μετά, η ίδια διαδικασία ακολουθείται για τη neighbors λίστα (Γραμμές 12-19, Αλγόριθμος 4.4). Όσον αφορά στον τρόπο με τον οποίο πραγματοποιείται ο υπολογισμός του correlation, ανάγεται στην εύρεση της Ευκλείδειας Απόστασης μεταξύ των εκάστοτε DFT συντελεστών. Με βάση τον τύπο που έχει περιγραφεί στην Ενότητα 2 - Εξίσωση 2.5.

TopK

Όλα τα παραπάνω συνθέτουν τη διαδικασία της εύρεσης των k πιο όμοιων μετοχών, η οποία απεικονίζεται και στο Σχήμα 4.8. Με βάση το Σχήμα 4.8, καθένα bucket αποτελείται από τις λίστες native και neighbors, πάνω στις οποίες αναζητούνται τα όμοια χρονικά παράθυρα των μετοχών. Αφού έχει πραγματοποιηθεί η εύρεση των όμοιων χρονικών παραθύρων έχει σχηματιστεί ένα νέο Hashmap, το οποίο περιέχει ως κλειδί το όνομα της εκάστοτε μετοχής

Αλγοριθμος 4.4: *Αλγόριθμος Query 1-Αναζήτηση ομοιότητας χρονικών παραθύρων των μετοχών*

Είσοδος: *X* (*Threshold, K, listOfStocks, T*)
Έξοδος: *Y* (*HashMap<String, List of PAIRS>*)
HashMap<String, COEF> ← κάλεσε τη μέθοδο "Αναζήτηση Χρονικών Παραθύρων"
for Για κάθε ένα entry του HashMap **do**
for Για κάθε ένα στοιχείο COEF της λίστας native λίστας **do**
if Το Stream ID1 δεν ταυτίζεται με το Stream ID2 και η διαφορά των χρονικών παραθύρων δεν υπερβαίνει το T **then**
Υπολόγισε το correlation
if correlation > threshold και correlation <= 1 **then**
Δημιούργησε ένα αντικείμενο της κλάσης PAIR
Εκχώρησε το αντικείμενο σε ένα Hashmap της κλάσης PAIR
end if
end if
end for
for Για κάθε ένα στοιχείο COEF της λίστας neighbors λίστας **do**
if Το Stream ID1 δεν ταυτίζεται με το Stream ID2 και η διαφορά των χρονικών παραθύρων δεν υπερβαίνει το T **then**
Υπολόγισε το correlation
if correlation > threshold και correlation <= 1 **then**
Δημιούργησε ένα αντικείμενο της κλάσης PAIR
Εκχώρησε το αντικείμενο σε ένα Hashmap της κλάσης PAIR
end if
end if
end for
Υπολόγισε τα TopK
end for

εισόδου και ως τιμή μια λίστα με τις σχετιζόμενες μετοχές (αντικείμενα της κλάσης PAIR), όπως αποτυπώνεται και στο Σχήμα 4.8. Εν συνεχεία, προκειμένου να υπολογιστούν τα k πιο όμοια ζευγάρια εκτελείται ο Αλγόριθμος 4.5, κατά τον οποίο για κάθε ένα ζευγάρι τιμών, δηλαδή για κάθε αντικείμενο της κλάσης PAIR, υπολογίζεται το Pearson Correlation. Στην περίπτωση που η τιμή αυτή υπερβαίνει ή είναι ίση με τη τιμή του threshold, τότε έχει βρεθεί ένα πράγματι όμοιο ζευγάρι, το οποίο με τη σειρά του προστίθεται σε μια λίστα. Από αυτή τη λίστα τελικά λαμβάνονται υπόψη τα k πιο όμοια. Αυτή η διαδικασία αποτελεί τη βάση για την πρόβλεψη των χρηματιστηριακών μετοχών στην επόμενη Ενότητα, στην οποία αποτυπώνεται ο τρόπος με τον οποίο θα αξιοποιηθούν, ώστε να δοθεί απάντηση στο αρχικό ερώτημα.

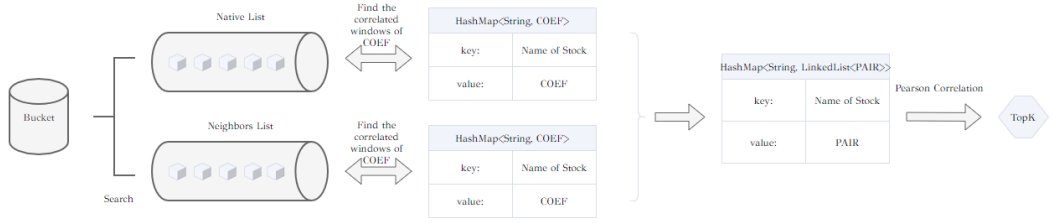
4.2.5 Πρόβλεψη μετοχών

Στο τελευταίο στάδιο της παρούσας διπλωματικής εργασίας εντάσσεται η πρόβλεψη τον χρηματιστηριακών μετοχών, η οποία έχει ως επικείμενο στόχο να συμβάλει στη μείωση της ανάληψης ρίσκου, καθώς επίσης και στη μεγιστοποίηση των κερδών στον επενδυτικό τομέα.

Έως αυτή τη στιγμή έχει δειχθεί ο τρόπος υπολογισμού των k πιο όμοιων μετοχών δεδομένου ότι έχει δοθεί στην είσοδο μια λίστα από μετοχές, οι οποίες είναι επιθυμητό να προβλεφθούν. Όμως, είναι εμφανές ότι οι μετοχές μεταξύ τους εμφανίζουν διαφορετικά εύρη

Αλγοριθμος 4.5: *TopK*

Είσοδος: *X* (*k, threshold*)
Έξοδος: *Y* (*Λίστα με τα TopK*)
  for Για κάθε ένα ζευγάρι εισόδου **do**
    Υπολόγισε το Pearson Correlation
    if Η τιμή του Pearson είναι μεγαλύτερη ή ίση από το threshold **then**
      Πρόσθεσε σε μια λίστα το αντικείμενο της κλάσης PAIR
    end if
  end for
  Από τη λίστα βρες τα k πιο όμοια και επέστρεψε τα



Σχήμα 4.8: *Top K*

τιμών με αποτέλεσμα να μην καθίσταται εφικτή η σύγκριση. Προκειμένου να αντιμετωπιστεί αυτός ο περιορισμός κρίνεται αναγκαία η κανονικοποίηση. Πιο συγκεκριμένα, χρησιμοποιείται ο παρακάτω τύπος:

$$y = \frac{(x - \min)}{(\max - \min)} \tag{4.1}$$

Με βάση των παραπάνω τύπο προκύπτουν τιμές που κυμαίνονται μεταξύ του [0,1]. Ο λόγος που επιλέγεται η κανονικοποίηση έγκειται στο γεγονός ότι τα δεδομένα εισόδου εμφανίζονται σε διαφορετική κλίμακα με αποτέλεσμα να μην καθίσταται εφικτή η οπτικοποίηση των αποτελεσμάτων, ώστε να επιτευχθεί η σύγκριση.

Προκειμένου να πραγματοποιηθεί η πρόβλεψη χρησιμοποιούνται όλα τα correlated pairs που έχουν προκύψει από την παραπάνω διαδικασία. Για καθεμιά μετοχή διατηρούνται τα k όμοια χρονικά παράθυρα πάνω στα οποία εφαρμόζεται το μοντέλο του Multiple Linear Regression. Πριν την εφαρμογή του μοντέλου ελέγχεται αν οι ανεξάρτητες μεταβλητές, δηλαδή τα correlated pairs συσχετίζονται, ώστε να αποφευχθεί το πρόβλημα της πολυσυγγραμμικότητας και να πραγματοποιηθεί ορθότερα η πρόβλεψη. Αφού εφαρμοστούν οι τύποι όπως έχουν αναλυθεί στη Ενότητα 2 (Θεωρητικό Υπόβαθρο), ακολουθεί η αξιολόγηση του μοντέλου. Στην ουσία χρησιμοποιούνται βασικές σχέσεις της Γραμμικής Άλγεβρας, ώστε να υπολογιστούν οι συντελεστές που σχετίζονται με τον προσδιορισμό της κλίσης του επιπέδου παλινδρόμησης.

Προκειμένου να πραγματοποιηθεί η αποδοτικότητα του μοντέλου υπολογίζεται ένας πίνακας με μήκος 90 (τα προβλεπόμενα 90 λεπτά από τη στιγμή που έχει ολοκληρωθεί το τελευταίο χρονικό παράθυρο), του οποίου οι τιμές αξιολογούνται με βάση την απόκλισή τους από τις

πραγματικές μελλοντικές τιμές των ανεξάρτητων μεταβλητών. Επίσης, αξίζει να σημειωθεί ότι χρησιμοποιείται ο υπολογισμός τόσο του αθροίσματος τετραγώνων των σφαλμάτων όσο (Sum Square Error - SSE) και του συνολικού αθροίσματος τετραγώνων (Sum Square Total - SST), ώστε να φανερωθεί η ορθότητα του μοντέλου. Το Sum Square Error - SSE είναι επιθυμητό να είναι μικρό και το Sum Square Total - SST μπορεί να θεωρηθεί ως ένα μέτρο της συνολικής μεταβλητότητας του συνόλου δεδομένων.

Κεφάλαιο 5

Εξαγωγή Πειραμάτων

Στο παρόν κεφάλαιο παρουσιάζονται τα πειραματικά αποτελέσματα, τα οποία αποδεικνύουν τόσο την εγκυρότητα όσο και την ορθότητα του αλγορίθμου.

5.1 Τοπολογία του cluster

Τα πειράματα διεξάγονται στον cluster του Softnet, ο οποίος αποτελείται από 11 όμοια μηχανήματα Dell PowerEdge R300, τα οποία διαθέτουν τα παρακάτω χαρακτηριστικά:

- Quad Core Xeon X3323 2.5GHz, 11 μηχανήματα με 3 task slots το καθένα
- 8GB RAM
- 500GB HDD

5.2 Δεδομένα

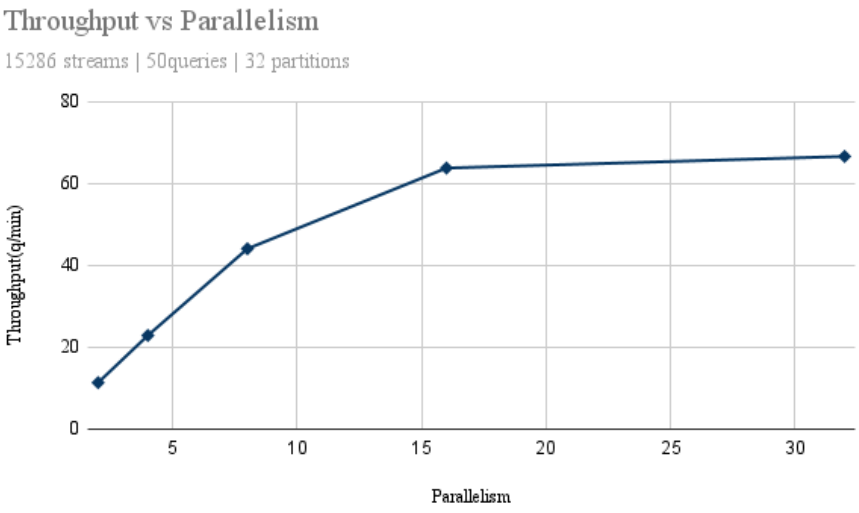
Τα χρηματιστηριακά δεδομένα που χρησιμοποιούνται ως είσοδος στο σύστημα του Kafka, αντλούνται από την πηγή: [12] και αφορούν μετρήσεις των τιμών των μετοχών για το έτος 2019. Πιο συγκεκριμένα, δημιουργείται ένα dataset το οποίο διαθέτει έως 60000 μετοχές, για τις οποίες λαμβάνονται τιμές ανά λεπτό. Οι μετοχές αυτές αφορούν όλες τις ημέρες του εκάστοτε μήνα του έτους 2019.

5.3 Πειραματικά αποτελέσματα

5.3.1 Πείραμα 1

Χρησιμοποιώντας 15286 streams, στέλνοντας 50 queries και δημιουργώντας 32 partitions στα δεδομένα, ώστε κάθε φυσικό μηχάνημα, ανάλογα με τον αριθμό των workers, να διαβάζει παράλληλα και ένα διαφορετικό τμήμα των δεδομένων, προκύπτει το Σχήμα 5.1. Η γραφική παράσταση του Σχήματος 5.1 δείχνει, πως αυξάνοντας τον αριθμό των workers, που χρησιμοποιούνται για την επεξεργασία των δεδομένων, αυξάνεται, και για την ακρίβεια σχεδόν διπλασιάζεται, το throughput σε κάθε περίπτωση. Αυτό συμβαίνει μέχρι ο αριθμός των workers γίνει ίσος με 16, καθώς μετά από αυτό το σημείο σταθεροποιείται. Στο συγκεκριμένο

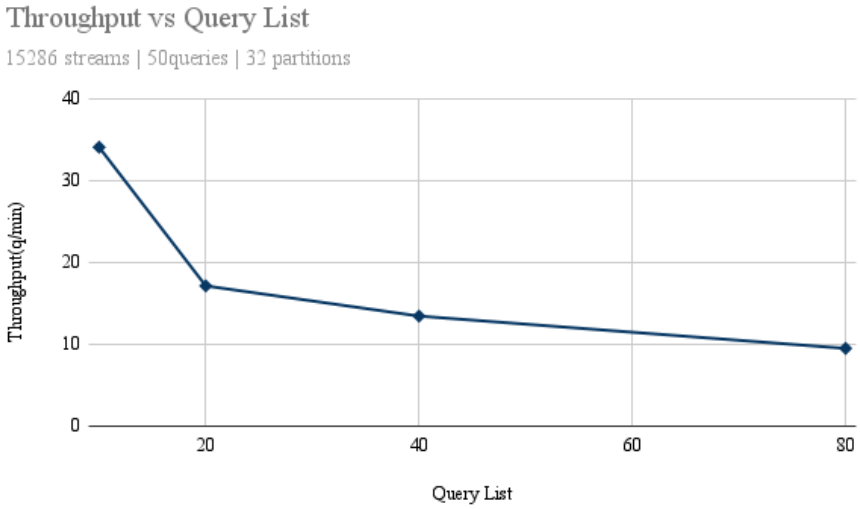
πείραμα επιλέγεται η αύξηση του αριθμού των φυσικών μηχανημάτων, προκειμένου να δειχθεί εάν επιτυγχάνεται η κλιμακωσιμότητα. Αυτό σημαίνει ότι όσο αυξάνεται ο αριθμός των πόρων-workers, ο αλγόριθμος που εφαρμόζεται, επεξεργάζεται αποδοτικότερα και ταχύτερα, μεγαλύτερο πλήθος δεδομένων. Αυτό στην περίπτωση αυτή συμβαίνει, όπως αποδεικνύεται και στο παρακάτω γράφημα. Ωστόσο, αξίζει να σημειωθεί ότι η αύξηση του μεγέθους του συστήματος έχει αντίτιμο στην απόδοση, καθώς δυσχεραίνεται η επικοινωνία και ο συγχρονισμός των επιμέρους μηχανημάτων, γεγονός που αποδεικνύεται με την σταθεροποίηση του throughput για περισσότερους από 16 workers.



Σχήμα 5.1: *Throughput vs Parallelism*

5.3.2 Πείραμα 2

Χρησιμοποιώντας 15286 streams, στέλνοντας 50 queries και δημιουργώντας 32 partitions στα δεδομένα, ώστε κάθε φυσικό μηχάνημα, ανάλογα με τον αριθμό των workers, να διαβάζει παράλληλα και ένα διαφορετικό τμήμα των δεδομένων, προκύπτει το Σχήμα 5.2. Στην προκειμένη περίπτωση δίδεται στην είσοδο του συστήματος μια λίστα με τις επιθυμητές μετοχές για τις οποίες απαιτείται η πρόβλεψη. Όσο περισσότερο αυξάνεται το μέγεθος αυτής της λίστας τόσο περισσότερο μειώνεται και ο ρυθμός με τον οποίο δίνεται και η απάντηση από το σύστημα. Αναλυτικότερα, μέχρι το μέγεθος να γίνει ίσο με 40 σχεδόν υποδιπλασιάζεται το throughput, ώσπου όταν το μέγεθος λάβει τη τιμή 80 ο ρυθμός απάντησης δεν μειώνεται δραματικά σε σχέση με τις 40 μετοχές στην είσοδο. Αυτό μπορεί να οφείλεται στο γεγονός ότι το πλήθος των δεδομένων (χρονικά παράθυρα) ανά μετοχή ποικίλλει, με αποτέλεσμα να μην χρειάζεται πολύς χρόνος για να δοθεί απάντηση σε κάποιες από τις ερωτηθέντες μετοχές και αντίστροφα.



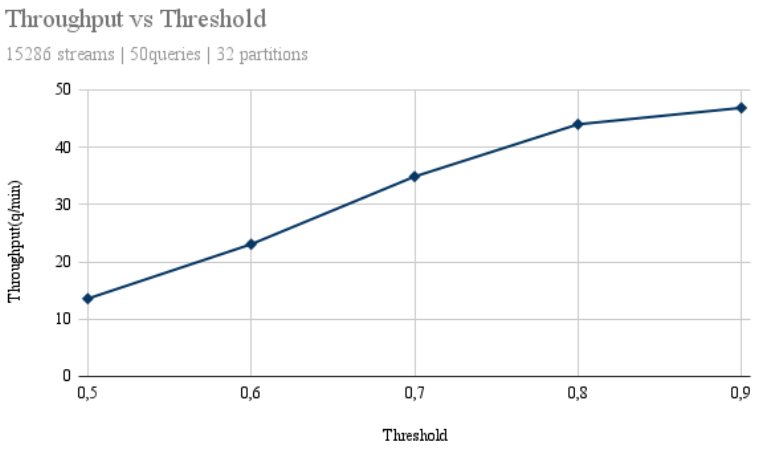
Σχήμα 5.2: *Throughput vs Query List*

Πίνακας 5.1: *Threshold vs Buckets*

Threshold	**Number of buckets**
0.5	4
0.6	9
0.7	9
0.8	16
0.9	25

5.3.3 Πείραμα 3

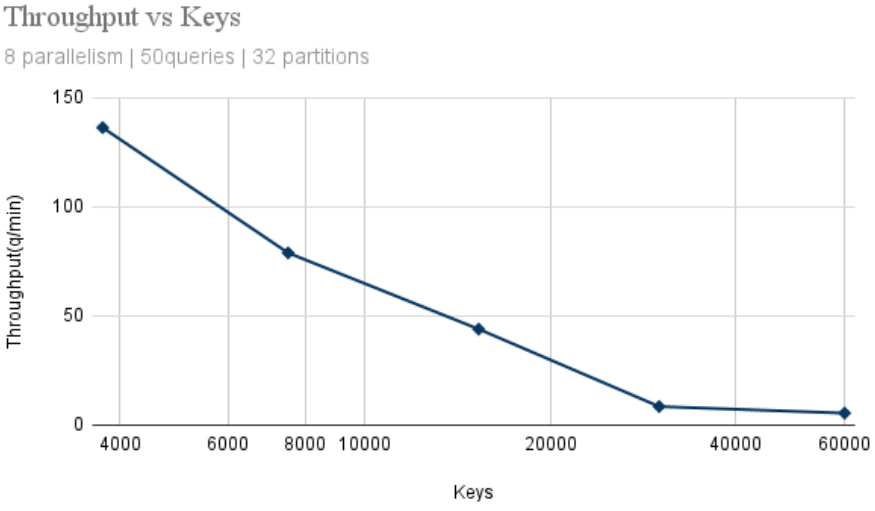
Ομοίως, χρησιμοποιώντας 15286 streams, στέλνοντας 50 queries και δημιουργώντας 32 partitions στα δεδομένα, προκύπτει το Σχήμα 5.3. Στην προκειμένη περίπτωση είναι επιθυμητό να δειχθεί κατά πόσο η τιμή του threshold είναι δυνατό να επηρεάσει το throughput. Συγκεκριμένα, όσο αυξάνεται το threshold, τόσο αυξάνεται, σχεδόν διπλασιάζεται και σε αυτή την περίπτωση, ο ρυθμός με τον οποίο απαντώνται τα εκάστοτε ερωτήματα. Αυτό σχετίζεται με τον αριθμό των buckets που δημιουργούνται, καθώς επίσης και με το πλήθος και τον τρόπο κατανομής των δεδομένων στα επικείμενα buckets. Πιο συγκεκριμένα, με βάση τους τύπους που αναλύονται στην Ενότητα του μαθηματικού υποβάθρου, προκύπτει ο πίνακας 5.1. Είναι λογικό ότι όσα περισσότερα buckets δημιουργούνται, τα δεδομένα θα κατανέμονται με αποδοτικότερο τρόπο, σχεδόν ομοιόμορφα και επομένως η αναζήτηση στα buckets θα πραγματοποιείται με λιγότερα δεδομένα όσο αυξάνεται το threshold. Όλα τα παραπάνω αποτυπώνονται στο γράφημα του Σχήματος 5.3, στο οποίο η σχέση του threshold με το πλήθος των buckets προσεγγίζει τη γραμμική.



Σχήμα 5.3: *Throughput vs Threshold*

5.3.4 Πείραμα 4

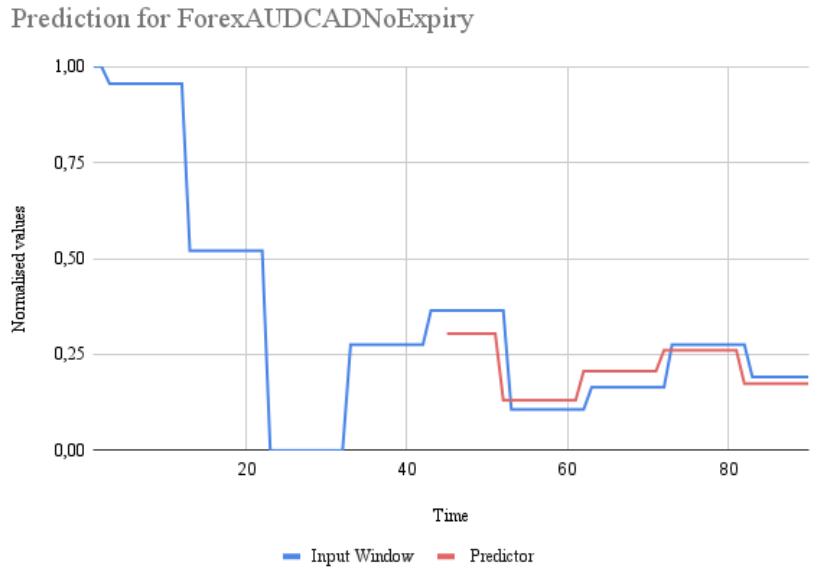
Ομοίως, χρησιμοποιώντας παραλληλισμό ίσο με 8, δημιουργώντας 16 buckets, στέλνοντας 50 queries και δημιουργώντας 32 partitions στα δεδομένα, προκύπτει το Σχήμα 5.4. Από αυτό το πείραμα είναι εμφανές ότι εισάγοντας και τις 60000 μετοχές, το σύστημα ανταποκρίνεται, αν και παράγει έξοδο με αργό ρυθμό. Βάζοντας λίγα κλειδιά, μόλις 4000, το throughput είναι αρκετά υψηλό και σταδιακά όσο περισσότερα κλειδιά εισάγονται προς επεξεργασία, το throughput μειώνεται με ομαλό τρόπο. Συγκεκριμένα, μέχρι τα 15286 κλειδιά ο ρυθμός με τον οποίο απαντώνται τα ερωτήματα υποδιπλασιάζεται όσο αυξάνεται το πλήθος των κλειδιών, ενώ στα 30000 μειώνεται απότομα και στα 60000 ελαχιστοποιείται, διατηρώντας ακόμη την ικανότητα απάντησης, αλλά με αρκετά αργό ρυθμό. Αυτό μπορεί να οφείλεται στο γεγονός ότι τα χρονικά παράθυρα των μετοχών, τα οποία ανέρχονται κοντά στις 900.000 ($\sim$ 15 παράθυρα ανά μετοχή X 60000 μετοχές), κατανέμονται σχεδόν ομοιόμορφα και όχι απολύτως με αποτέλεσμα να αυξάνεται ο αριθμός των συγκρίσεων σε κάθε bucket.



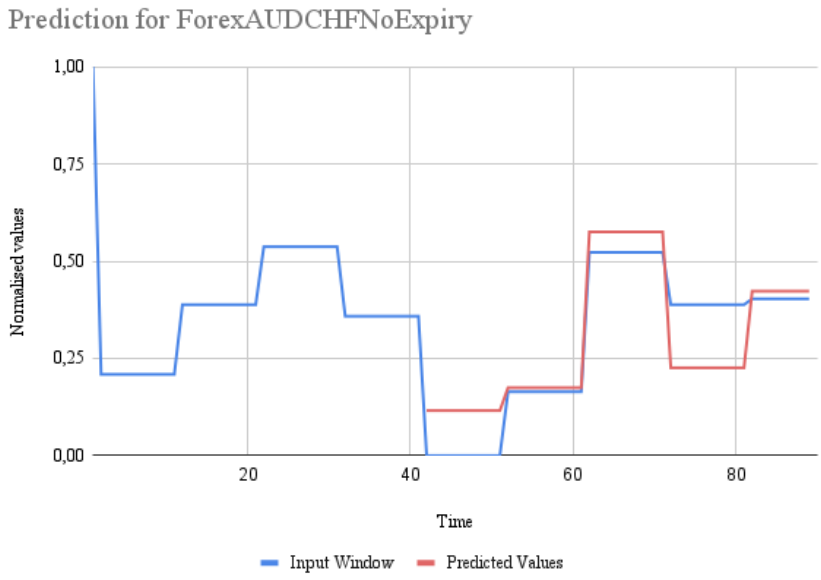
Σχήμα 5.4: *Throughput vs Keys*

5.3.5 Πρόβλεψη

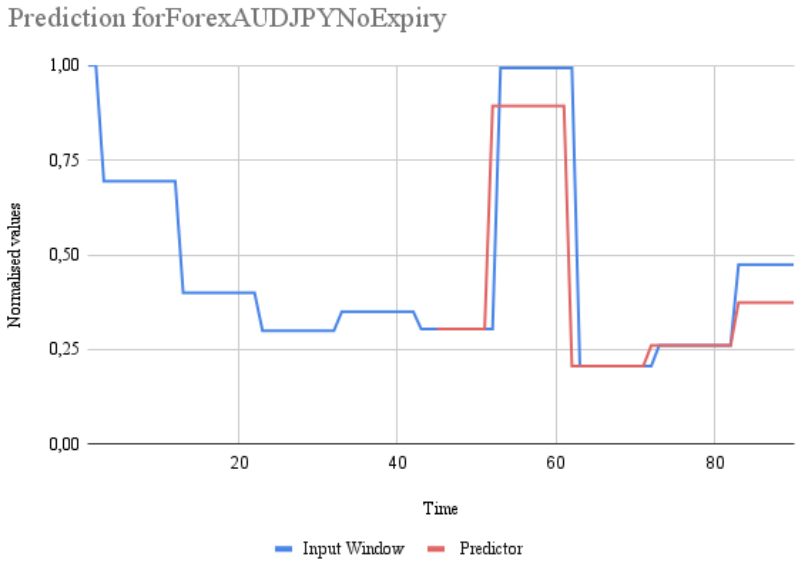
Έχοντας εφαρμόσει το μοντέλο του Multiple Linear Regression, έχει προκύψει ένα διάνυσμα, στο οποίο εμπεριέχονται όλες οι τιμές που συνιστούν τις προβλεπόμενες. Αναλυτικότερα όπως είναι εμφανές και στα Σχήματα 5.5, 5.6 και 5.7, έχει χρησιμοποιηθεί ως παράθυρο εισόδου το χρονικό διάστημα από την τιμή 0 - 45. Μετά από αυτή την χρονική στιγμή, όπως φαίνεται και με την κόκκινη γραμμή, ελέγχεται κατά πόσο οι προβλεπόμενες τιμές προσεγγίζουν τις πραγματικές μελλοντικές τιμές της εκάστοτε μετοχής.



Σχήμα 5.5: *Πρόβλεψη Τάσης Μετοχής*



Σχήμα 5.6: *Πρόβλεψη Τάσης Μετοχής*



Σχήμα 5.7: *Πρόβλεψη Τάσης Μετοχής*

Εύκολα μπορεί να δειχθεί ότι τα αποτελέσματα είναι ικανοποιητικά, καθώς οι προβλεπόμενες μελλοντικές τιμές των μετοχών (κόκκινη γραμμή) ακολουθεί αρκετά καλά την μπλε γραμμή, δηλαδή τις πραγματικές τιμές που λαμβάνει η εκάστοτε μετοχή που ελέγχεται. Παράλληλα, αξίζει να σημειωθεί ότι μετά από την πρόβλεψη 10 μετοχών και αφού έχει καταγραφεί το άθροισμα των σφαλμάτων, υπολογίζεται το μέσο τετραγωνικό σφάλμα το οποίο στην συγκεκριμένη περίπτωση είναι ίσο με 0.57109, γεγονός που υποδηλώνει ότι το μοντέλο πρόβλεψης προσεγγίζει ικανοποιητικά τις προβλεπόμενες μετοχές. Η ορθότητα της πρόβλεψης οφείλεται στο γεγονός ότι για την τροφοδότηση του μοντέλου έχουν χρησιμοποιηθεί κάθε φορά οι k πιο όμοιες μετοχές, οι οποίες έχουν προκύψει ορθά από το προηγούμενο στάδιο της εργασίας.

Κεφάλαιο 6

Επίλογος

Στο κεφάλαιο αυτό θα αναλυθούν τα αποτελέσματα της παρούσας διπλωματικής εργασίας, καθώς επίσης και οι αντίστοιχες μελλοντικές προεκτάσεις που είναι εφικτό να εφαρμοστούν.

6.1 Συμπεράσματα

Στην παρούσα διπλωματική εργασία επιχειρείται η πρόβλεψη της πορείας των χρηματιστηριακών μετοχών μέσω της κατανεμημένης και παράλληλης επεξεργασίας μεγάλου όγκου δεδομένων μετοχών. Η διαδικασία αυτή πραγματοποιείται με την εισαγωγή δεδομένων στο Kafka, τα οποία μετατρέπονται σε streams από το SDE, το οποίο με τη σειρά του παρέχει τη δυνατότητα διαχείρισης και επεξεργασίας μεγάλου όγκου δεδομένων, κατανεμημένα και παράλληλα. Προκειμένου να επιτευχθεί το ζητούμενο δημιουργείται μια σύνοψη στο σύστημα του SDE. Στη σύνοψη αυτή εισάγονται χρονικά παράθυρα μετοχών, στα οποία πρώτα έχει εφαρμοστεί ο αλγόριθμος Discrete Fourier Transform, ώστε να αποθηκευτεί μόνο η χρήσιμη πληροφορία, η οποία είναι απαραίτητη για τη μετέπειτα επεξεργασία και την εξαγωγή αποτελεσμάτων. Σε επόμενο στάδιο, τα εκάστοτε χρονικά παράθυρα κάθε μετοχής αποστέλλονται στα επικείμενα buckets, στα οποία ανήκουν, μέσω της διαδικασίας του estimate. Ακολούθως, ενώνονται τα αντίστοιχα buckets από κάθε worker ξεχωριστά, ώστε να προκύψουν τα τελικά k, στα οποία θα εφαρμοστεί και το ερώτημα, το οποίο τίθενται από τον χρήστη. Η διαδικασία αυτή αντιπροσωπεύει το reduce. Με αυτό τον τρόπο θα προκύψουν οι k πιο όμοιες μετοχές μέσω της εφαρμογής του Pearson Correlation. Αξίζει να σημειωθεί ότι εξετάζεται η αποδοτικότητα του αλγορίθμου μεταβάλλοντας διάφορες παραμέτρους, ώστε να αποδειχθεί αν διαθέτει τη δυνατότητα του scalability: σε σχέση με τους πόρους του συστήματος, τα streams εισόδου και τη λίστα των επιθυμητών μετοχών που πρόκειται να προβλεφθούν. Προκειμένου να αποδειχθούν τα παραπάνω, τα πειράματα λαμβάνονται τόσο τοπικά όσο και απομακρυσμένα στο cluster του Πολυτεχνείου. Τα αποτελέσματα είναι ικανοποιητικά, καθώς το σύστημα διαχειρίζεται ένα μεγάλο όγκο κλειδιών-μετοχών σε εύλογα χρονικά διαστήματα.

Παράλληλα, εφαρμόζεται το μοντέλο πρόβλεψης Multiple Linear Regression, αφού ελεγχθούν όλες οι προϋποθέσεις, στο οποίο λαμβάνονται υπόψη οι k πιο όμοιες μετοχές. Το μοντέλο πρόβλεψης τροφοδοτείται με τις παρελθοντικές τιμές των όμοιων μετοχών. Με αυτό τον τρόπο προκύπτει, μέσω του υπολογισμού πινάκων, μια εξίσωση παλινδρόμησης, η οποία περιλαμβάνει τα χρήσιμα χαρακτηριστικά, τα οποία επηρεάζουν την πορεία της μετοχής που

είναι επιθυμητό να προβλεφθεί. Τα αποτελέσματα είναι ικανοποιητικά, λόγω του συντελεστή προσδιορισμού R squared, με βάση τον οποίο αποδεικνύεται ότι οι μεταβολές στις ανεξάρτητες μεταβλητές θα επηρεάσουν την πορεία της εξαρτημένης μεταβλητής.

6.2 Μελλοντικές Επεκτάσεις

Το σύστημα που αναπτύσσεται στα πλαίσια αυτής της διπλωματικής εργασίας θα μπορούσε να βελτιωθεί και να επεκταθεί περαιτέρω. Συγκεκριμένα, στο πλαίσιο της πρόβλεψης είναι δυνατόν να δοθούν βάρη σε κάθε ένα χαρακτηριστικό (k όμοια μετοχή ή ανεξάρτητη μεταβλητή), ώστε να πραγματοποιηθεί μείωση των διαστάσεων στον χώρο και να καθίσταται εφικτή η οπτικοποίηση των αποτελεσμάτων.

Παράλληλα, όσον αφορά στη βελτίωση της απόδοσης του αλγορίθμου DFT, θα ήταν εφικτό να εφαρμοστεί μια διαφορετική μετρική απόστασης (όπως για παράδειγμα η Cosine ή Hamming) με βάση την οποία θα πραγματοποιηθεί η εύρεση της ομοιότητας. Μάλιστα, μπορεί να επεκταθεί και η λειτουργικότητά αυτή, δίνοντας τη δυνατότητα να επιλέξει ο εκάστοτε χρήστης. Τέλος, άλλη μια βελτίωση που θα επιφέρει θετικά αποτελέσματα στην απόδοση του αλγορίθμου, ώστε να επεξεργάζεται γρήγορα περισσότερα κλειδιά, θα μπορούσε να είναι η βελτιστοποίηση που σχετίζεται με τον ομοιόμορφο κατακερματισμό των χρονικών παραθύρων στα εκάστοτε buckets.

Βιβλιογραφία

[1] *Apache Hadoop*. https://hadoop.apache.org/docs/stable/.

[2] *Map Reduce*. https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html.

[3] *Apache Spark*. http://spark.apache.org.

[4] *Apache Storm*. https://storm.apache.org.

[5] *Apache Flink*. https://flink.apache.org/flink-applications.html.

[6] Hector Garcia-Molina, Jeffrey D. και Ullman Jennifer Widom. *Database Systems. The complete book*. Prentice Hall, 2η έκδοση, 2009.

[7] Phillip B. Gibbons και Yossi Matias. *Synopsis Data Structures for Massive Data Sets*. 1991.

[8] Nikolaos Pavlakis. *Scaling out streaming time series analytics on Storm*. Μεταπτυχιακή διπλωματική εργασία, Technical University of Crete, 2017.

[9] Yunyue Zhu και Dennis Shasha. *StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time. Proceedings of the 28th VLDB Conference*, Hong Kong, China, 2002.

[10] Arun N. Swami Rakesh Agrawal, Christos Faloutsos. *Efficient Similarity Search In Sequence Databases*. 1993.

[11] Antonis Kontaxakis. *DESIGN AND IMPLEMENTATION OF A DISTRIBUTED SYNOPSIS DATA ENGINE ON APACHE FLINK*. Μεταπτυχιακή διπλωματική εργασία, Technical University of Crete, 2020.

[12] *Financial Data*. https://zenodo.org/record/3886895#.YMHv1i0RrOT.