Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα

Εργασία 3

Δημητρακόπουλος Κωνσταντίνος AM: 1115201500034

Κακαβάς Αντώνιος AM: 1115201500052

Ανάπτυξη

Η εργασία αναπτύχθηκε στο περιβάλλον Google Collaboratory με τύπο αρχείου python notebook, όπου παραδίδεται. Επιπρόσθετα παραδίδεται ο πηγαίος κώδικας της εργασίας σε αρχεία .py σύμφωνα με τις προϋποθέσεις της εκφώνησης. Ή συνεργασία κατά την ανάπτυξη πραγματοποιήθηκε στο private repository.

Εκτέλεση

Τα παραδοτέα αρχεία εκτελούνται ως εξής:

- python3 forecast.py -d <dataset path> -n <number of time series selected> -t <"offline_all" or "online_all" or "online_self">
- python3 detect.py -d <dataset path> -n <number of time series selected> -t <"offline_all" or "online_all"> -mae < error value as double >
- python3 reduce.py –d <dataset> -q <queryset> -od <output_dataset_file> -oq <output_query_file> -t <"offline_all" or "online_all" >

Σχετικά με την παράμετρο '-t':

- "offline_all": Φόρτωση του offline εκπαιδευμένου ανά σύνολο χρονοσειρών μοντέλου.
- "online_all": Εκπαίδευση ανά σύνολο χρονοσειρών κατά την εκτέλεση.
- "online_self": Εκπαίδευση ανά χρονοσειρά κατά την εκτέλεση.

Η διαρρύθμιση των καταλόγων της εργασίας δεν πρέπει να μεταβληθεί για επιτυχή εκτέλεση της.

Κατά την εκτέλεση των αρχείων .py και όχι του αρχείου .ipynb εμφανίζονται warnings σχετικά με την μελλοντική μη υποστήριξη ορισμένων συναρτήσεων από τη πλατφόρμα tensorflow και μπορούν να αγνοηθούν, αφού δεν αφορούν την τωρινή εκτέλεση της εργασίας.

Κατάλογος αρχείων

Η εργασία που παραδίδεται αποτελείται από τα εξής αρχεία:

- split_dataset:

  Δέχεται ως όρισμα το μονοπάτι του dataset της εργασίας και το διαμελίζει στα σετ δεδομένων dataset.csv και queryset.csv τοποθετώντας αυτά στο ίδιο μονοπάτι.

- common_utils:

  Περιέχει κοινές βοηθητικές συναρτήσεις για όλες τις υποενότητες της εργασίας.

- forecast.py:

  Το αντίστοιχο αρχείο που περιγράφει η εκφώνηση. Πραγματοποιείται έλεγχος των ορισμάτων, φόρτωση του σετ δεδομένων και για διαφορετικές τιμές του ορίσματος “-t” πραγματοποιείται είτε φόρτωση του offline εκπαιδευμένου ανά σύνολο χρονοσειρών μοντέλου, είτε εκπαίδευση ανά σύνολο χρονοσειρών κατά την εκτέλεση με αποθήκευση του παραγόμενου μοντέλου, είτε εκπαίδευση ανά χρονοσειρά κατά την εκτέλεση. Η ανάλυση του μοντέλου πραγματοποιείται παρακάτω. Τελικά πραγματοποιείται πρόγνωση τιμών χρονοσειρών.

- detect.py:

  Το αντίστοιχο αρχείο που περιγράφει η εκφώνηση. Πραγματοποιείται έλεγχος των ορισμάτων, φόρτωση του σετ δεδομένων και για διαφορετικές τιμές του ορίσματος “-t” πραγματοποιείται είτε φόρτωση του offline εκπαιδευμένου ανά σύνολο χρονοσειρών μοντέλου, είτε εκπαίδευση ανά σύνολο χρονοσειρών κατά την εκτέλεση με αποθήκευση του παραγόμενου μοντέλου. Η ανάλυση του μοντέλου πραγματοποιείται παρακάτω. Τελικά πραγματοποιείται ανίχνευση ανωμαλιών χρονοσειρών.

- reduce.py:

  Το αντίστοιχο αρχείο που περιγράφει η εκφώνηση. Πραγματοποιείται έλεγχος των ορισμάτων, φόρτωση των σετ δεδομένων και για διαφορετικές τιμές του ορίσματος “-t” πραγματοποιείται είτε φόρτωση του offline εκπαιδευμένου ανά σύνολο χρονοσειρών μοντέλου, είτε εκπαίδευση ανά σύνολο χρονοσειρών κατά την εκτέλεση με αποθήκευση του παραγόμενου μοντέλου και εμφάνιση των παραγώμενων χρονοσειρών (αρχική, κωδικοποιημένη, αυτοκωδικοποιημένη) κατά την εκπαίδευση σε γραφήματα. Η ανάλυση του μοντέλου πραγματοποιείται παρακάτω. Τελικά παράγονται οι συμπιεσμένες-κωδικοποιημένες χρονοσειρές και εξάγονται κατάλληλα στα αρχεία εξόδου.

- project-algorithms.ipynb:

  Το notebook στο οποίο αναπτύχθηκε η εργασία. Περιλαμβάνει όλα τα παραπάνω.

Πρόγνωση Τιμών Χρονοσειρών

Όλα τα πειράματα πραγματοποιήθηκαν για ένα μεγάλο αριθμό από epochs. Τελικά επιλέχθηκε η τιμή της παραμέτρου epochs που ελαχιστοποιεί το σφάλμα πριν εμφανιστεί το φαινόμενο overfitting. Η αναγνώριση του κατάλληλου epoch προήλθε από τη γραφική παράσταση σφάλματος για τα σετ train, test κατά την εκπαίδευση (συνάρτηση plot_training_loss).

Τα πειράματα που πραγματοποιήθηκαν συνοψίζονται στο παρακάτω πίνακα.

Πείραμα	Batch Size	Layers	Units/Layer	Time Steps	Train Loss	Val Loss
1	32	4	32	50	8.3730e-04	0.0807
2	32	4	64	60	5.3747e-04	0.0847
3	32	4	128	50	5.7512e-04	0.0919
4	64	6	64	60	5.6178e-04	0.0865
5	64	6	128	50	5.5015e-04	0.0909
6	64	6	32	60	8.6226e-04	0.0900
7	64	8	64	50	5.7873e-04	0.0828
8	128	8	128	60	4.4686e-04	0.0857
9	128	6	64	50	5.7284e-04	0.0855

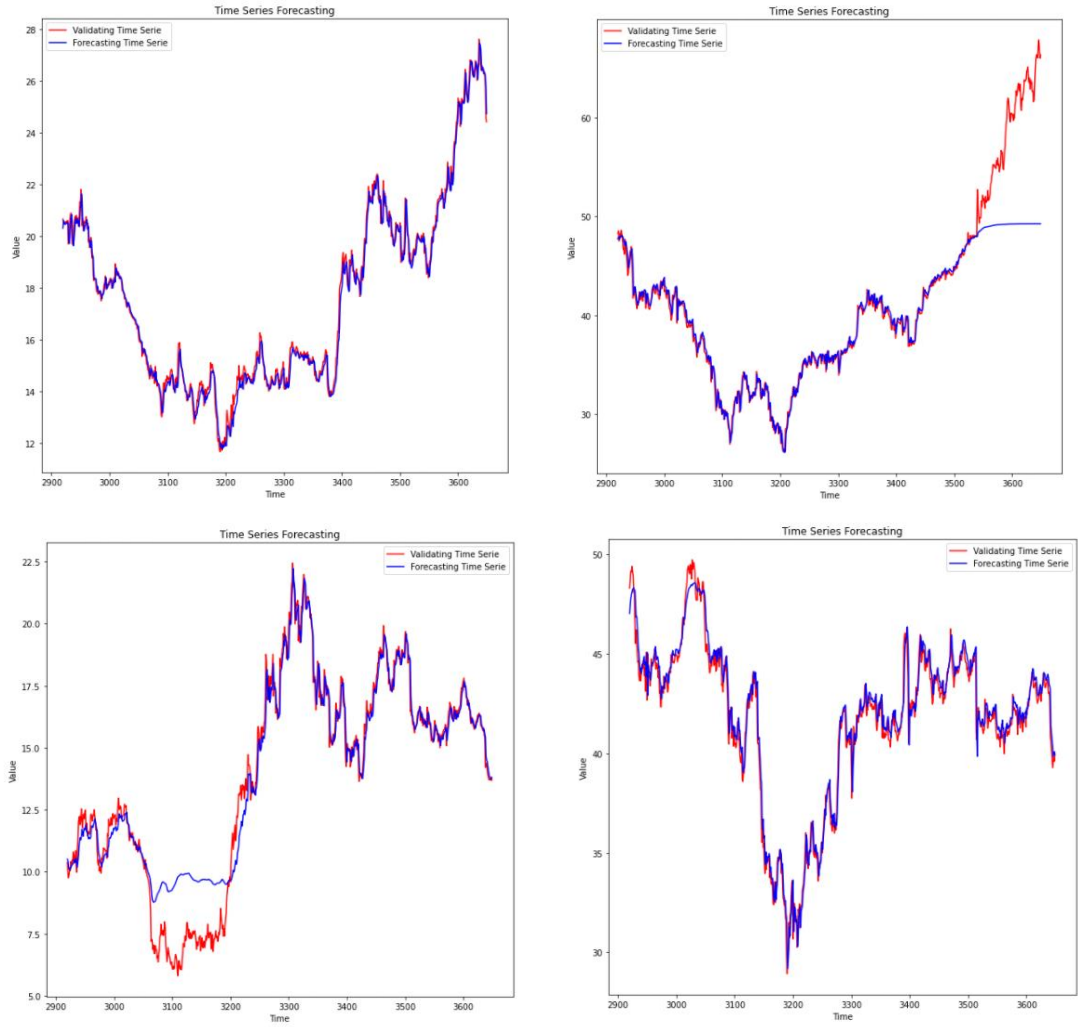
Κάθε layer αποτελείται από ένα επίπεδο LSTM και ένα επίπεδο dropout με πιθανότητα απόρριψης 0.2. Έτσι, αποφεύγεται το φαινόμενο υπερπροσαρμογής (overfitting).

Παρατηρήσεις:

- Ο χρόνος εκτέλεσης μειώνεται σημαντικά με την αύξηση του batch size, ωστόσο με χρήση πολύ μεγάλου batch size (>1024) το loss δεν ελαχιστοποιείται.
- Αυξάνοντας τον αριθμό των κρυφών επιπέδων του μοντέλου δεν παρατηρείται μείωση του σφάλματος. Τελικά το πρόβλημα δεν απαιτεί ένα πιο περίπλοκο μοντέλο από αυτά που δοκιμάστηκαν.
- Σημαντική επίδραση φαίνεται να έχει ο αριθμός των νευρώνων (units) ανά επίπεδο. Αυξάνοντας την τιμή της υπερπαραμέτρου το σφάλμα μειώνεται.
- Η υπερπαράμετρος time-steps (look-back) δεν μεταβάλλει σημαντικά τη τιμή του σφάλματος.

Το τελικό μοντέλο περιέχει τις τιμές υπερπαραμέτρων του πειράματος 8, όπου είναι οι βέλτιστες.

Ενδεικτικές προγνώσεις του τελικού μοντέλου



Το μοντέλο προσεγγίζει άρτια τις χρονοσειρές.

Ανίχνευση Ανωμαλιών Χρονοσειρών

Όπως στη πρόγνωση τιμών χρονοσειρών επιλέχθηκε κατάλληλα η τιμή της υπερπαραμέτρου epoch.

Τα πειράματα που πραγματοποιήθηκαν συνοψίζονται στο παρακάτω πίνακα.

Πείραμα	Batch Size	Layers	Units	Time Steps	Train Loss	Val Loss
1	32	2	64	50	0.2111	0.6069
2	32	2	128	60	0.2120	0.6148
3	32	2	256	50	0.2110	0.6074
4	64	2	64	60	0.2143	0.6132
5	64	4	128	50	0.2112	0.6121
6	128	6	64	60	0.1987	0.6241
7	128	6	128	50	0.1998	0.6218
8	128	6	32	60	0.1975	0.6243

Κάθε layer αποτελείται από ένα επίπεδο LSTM και ένα επίπεδο dropout με πιθανότητα απόρριψης 0.2. Έτσι, αποφεύγεται το φαινόμενο υπερπροσαρμογής (overfitting).

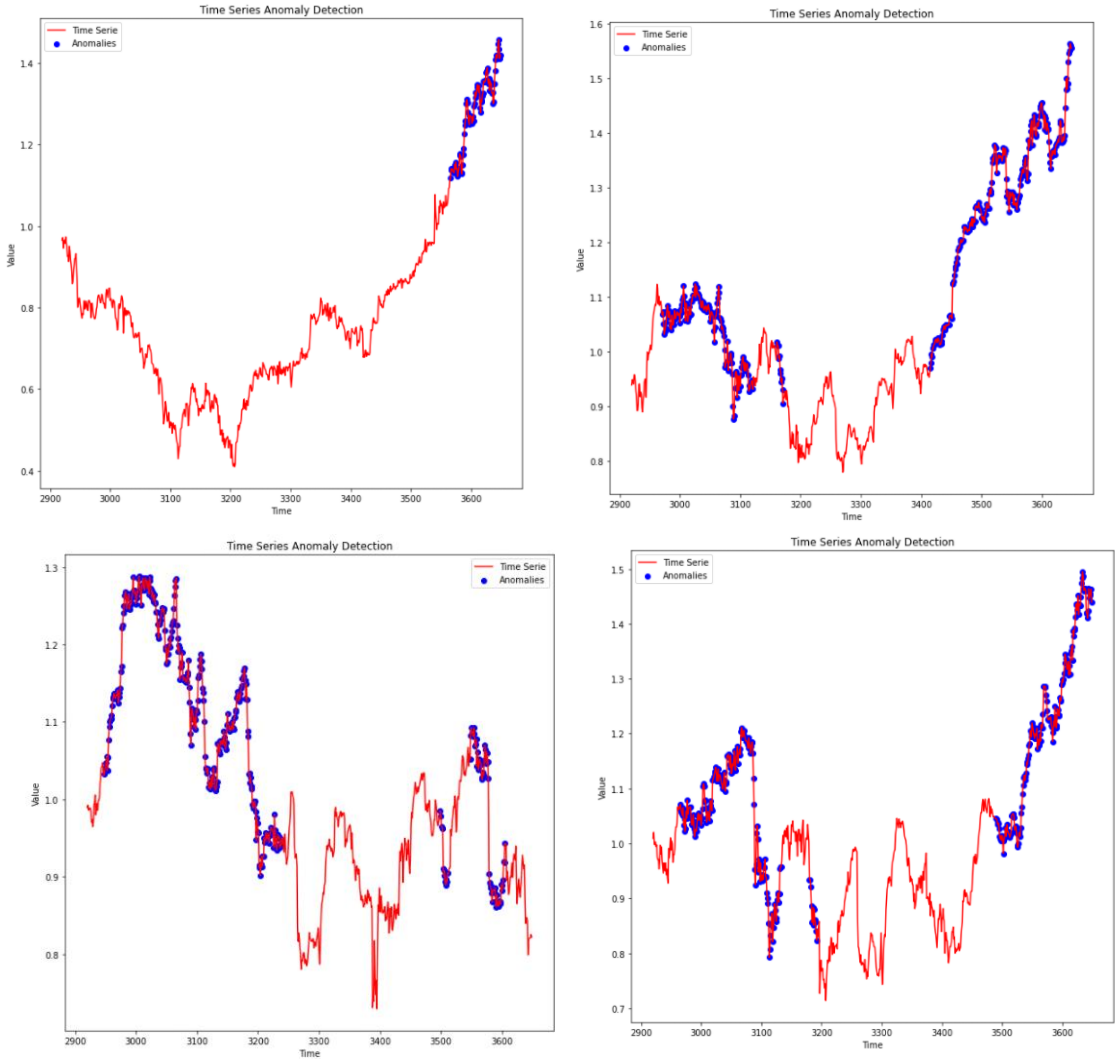
Παρατηρήσεις:

- Ο χρόνος εκτέλεσης μειώνεται σημαντικά με την αύξηση του batch size, ωστόσο με χρήση πολύ μεγάλου batch size (>1024) το loss δεν ελαχιστοποιείται.
- Αυξάνοντας τον αριθμό των κρυφών επιπέδων του μοντέλου παρατηρείται σημαντική μείωση του σφάλματος. Το πρόβλημα απαιτεί ένα αρκετά περίπλοκο μοντέλο.
- Μικρή επίδραση φαίνεται να έχει ο αριθμός των νευρώνων (units) ανά επίπεδο. Αυξάνοντας την τιμή της υπερπαραμέτρου το σφάλμα δεν μεταβάλλεται ισχυρά.
- Η υπερπαράμετρος time-steps (look-back) δεν μεταβάλλει σημαντικά τη τιμή του σφάλματος.

Το τελικό μοντέλο περιέχει τις τιμές υπερπαραμέτρων του πειράματος 8, όπου είναι οι βέλτιστες.

Ενδεικτικές ανιχνεύσεις του τελικού μοντέλου

MAE=0.65



Οι ανωμαλίες επισημαίνονται σε σημεία με απότομες μεταβάσεις ή απρόβλεπτη πορεία τιμών σε σύγκριση με τις υπόλοιπες χρονοσειρές, όπως αναμενόταν.

Συνελικτική Αυτοκωδικοποίηση Χρονοσειρών

Όπως στη πρόγνωση τιμών και στην ανίχνευση ανωμαλιών χρονοσειρών επιλέχθηκε κατάλληλα η τιμή της υπερπαραμέτρου epoch.

Τα πειράματα που πραγματοποιήθηκαν συνοψίζονται στο παρακάτω πίνακα.

Πείραμα	Batch Size	Layers	Filter Size	Window Length	Latent Dimension	Train Loss	Val Loss
1	32	2	16	10	3	0,6791	0,7576
2	32	2	16	50	13	0,5563	0,2912
3	32	2	32	50	13	0,5303	-0,1815
4	32	3	16	50	7	0,5317	-0,5217
5	32	3	16	10	2	0,5436	-0,9232
6	64	3	32	50	7	0,5304	-0,8348
7	64	3	64	10	2	0,5295	-0,2726
8	32	4	16	50	4	0,5307	-0,1027
9	32	4	32	50	4	0,6665	-0,8822

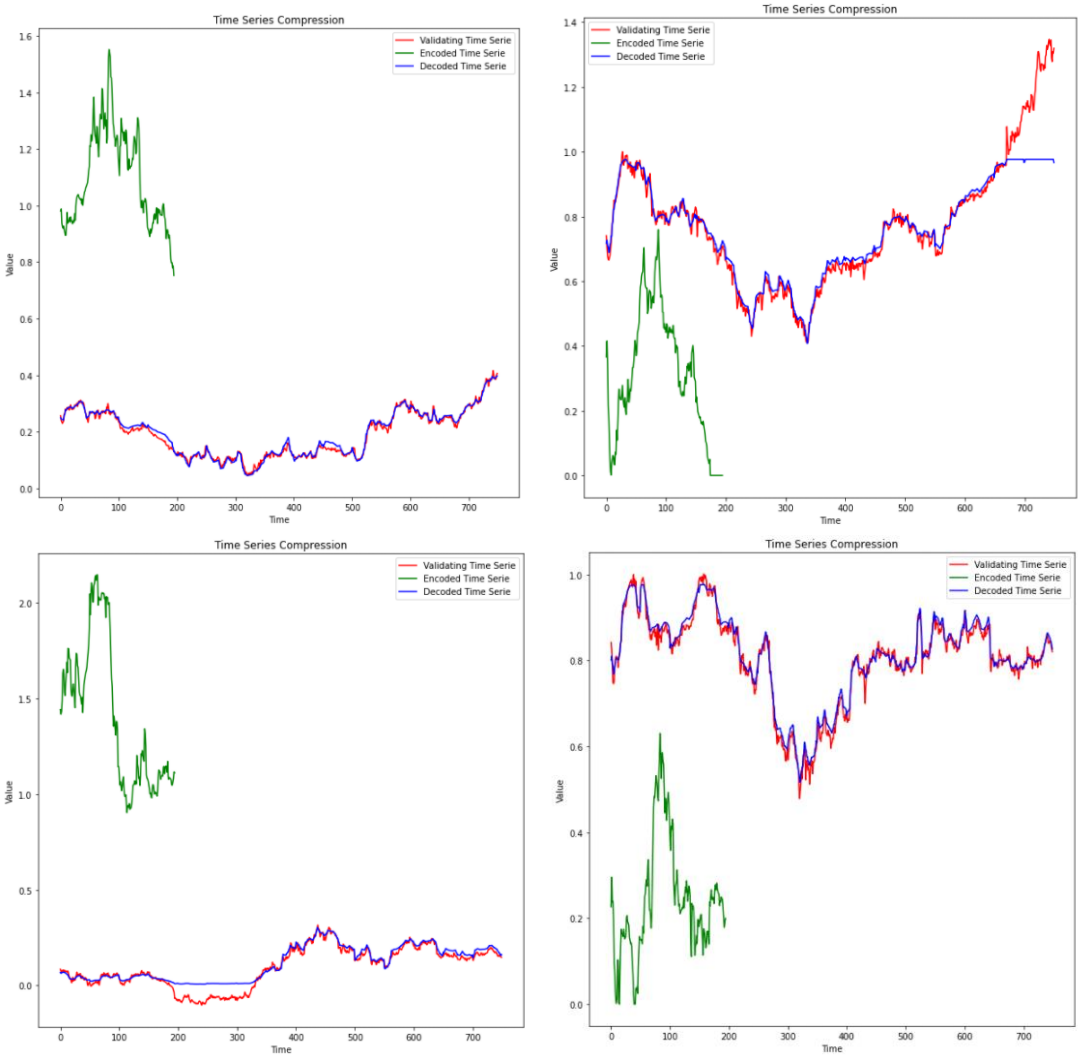
Κάθε layer αποτελείται από ένα επίπεδο conv1d και ένα επίπεδο αύξησης/μείωσης διάστασης. Κατά την αναφορά layers=3, υποδεικνύεται η ύπαρξη 3 επιπέδων downsampling (conv1d και maxpooling1d) και τριών εππέδων upsampling (conv1d και upsampling).

Παρατηρήσεις:

- Ο χρόνος εκτέλεσης μειώνεται σημαντικά με την αύξηση του batch size, ωστόσο με χρήση πολύ μεγάλου batch size (>1024) το loss δεν ελαχιστοποιείται.
- Σημαντική αύξηση ή μείωση του αριθμού των επιπέδων αποκλείει την ελαχιστοποίηση του σφάλματος.
- Μικρή επίδραση φαίνεται να έχει ο αριθμός των νευρώνων (units) ανά επίπεδο. Αυξάνοντας την τιμή της υπερπαραμέτρου το σφάλμα δεν μεταβάλλεται ισχυρά.
- Η υπερπαράμετρος time-steps (look-back) δεν μεταβάλλει σημαντικά τη τιμή του σφάλματος.

Το τελικό μοντέλο περιέχει τις τιμές υπερπαραμέτρων του πειράματος 4, όπου είναι οι βέλτιστες.

Ενδεικτικές συμπιέσεις του τελικού μοντέλου



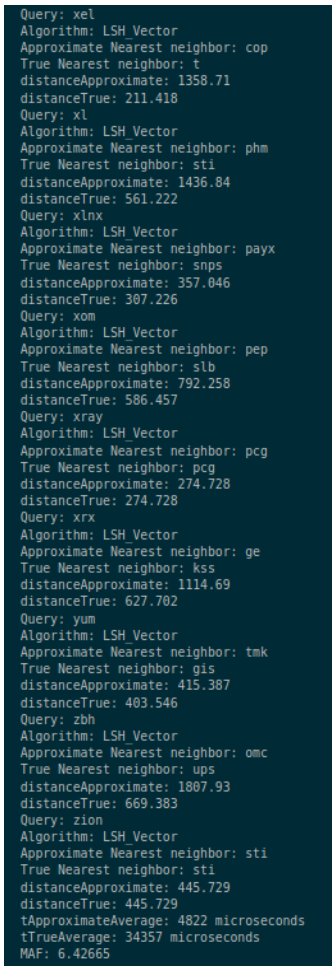
Η προσέγγιση της αρχικής χρονοσειράς είναι ικανοποιητική. Η encoded χρονοσειρά εμφανίζει κοινά μοτίβα με τις υπόλοιπες δύο, όπως αναμενόταν.

Σύγκριση σε Αναζήτηση και Συσταδοποίηση

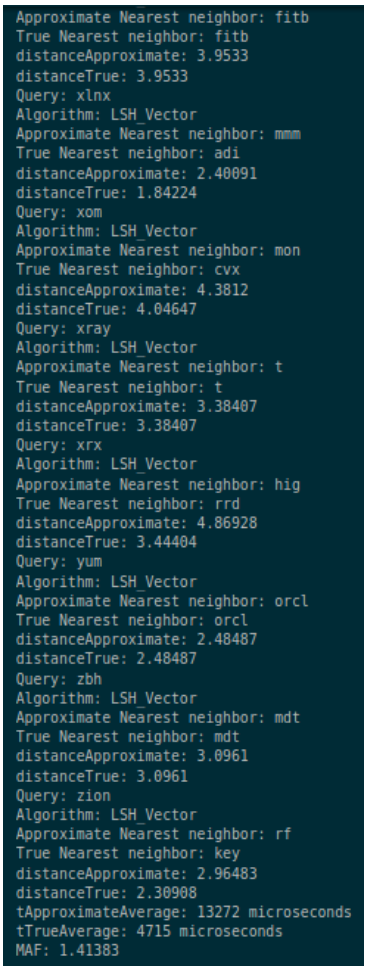
Οι παραγόμενες συμπιεσμένες χρονοσειρές από το μοντέλο συνελικτικής αυτοκωδικοποίησης συγκρίνονται με τις αρχικές χρονοσειρές του σετ δεδομένων στα συστήματα αναζήτησης πλησιέστερων γειτόνων (search) και συσταδοποίησης (cluster) διανυσμάτων και χρονοσειρών της εργασίας 2. Τα αποτελέσματα ανά εκτέλεση δύναται να διαφέρουν επειδή οι αλγόριθμοι search και cluster είναι πιθανοκρατικοί.

Αναζήτηση Διανυσμάτων με LSH (k=8 , L=5)

Τα αποτελέσματα:



Vector LSH initial



Vector LSH Compressed

Οι πραγματικές αποστάσεις των γειτόνων στη συμπιεσμένη αναπαράσταση είναι μικρότερες, από της αντίστοιχης αρχικής. Αυτό είναι αναμενόμενο, αφού πραγματοποιείται αναπαραγωγή της χρονοσειράς σε μικρότερη διάσταση και οι αναλογίες πραγματικής και σχετικής απόστασης παραμένουν όμοιες στις δύο αναπαραστάσεις. Συγκρίνοντας τις δύο αναπαραστάσεις ξεχωρίζεται ο κοινός γείτονας “t” υποδεικνύοντας μας πως η συμπιεσμένη αναπαράσταση αποτελεί μια καλή προσέγγιση της αρχικής.

Αναζήτηση Διανυσμάτων με Hypercube (probes=10 , k=10)

Τα αποτελέσματα:

```
Query: xel
Algorithm: Hypercube
Approximate Nearest neighbor: wor
True Nearest neighbor: t
distanceApproximate: 306.496
distanceTrue: 211.418
Query: xl
Algorithm: Hypercube
Query: xlnx
Algorithm: Hypercube
Approximate Nearest neighbor: msft
True Nearest neighbor: snps
distanceApproximate: 335.431
distanceTrue: 307.226
Query: xom
Algorithm: Hypercube
Approximate Nearest neighbor: pg
True Nearest neighbor: slb
distanceApproximate: 626.793
distanceTrue: 586.457
Query: xray
Algorithm: Hypercube
Approximate Nearest neighbor: wfc
True Nearest neighbor: pcg
distanceApproximate: 481.973
distanceTrue: 274.728
Query: xrx
Algorithm: Hypercube
Approximate Nearest neighbor: kss
True Nearest neighbor: kss
distanceApproximate: 627.702
distanceTrue: 627.702
Query: yum
Algorithm: Hypercube
Approximate Nearest neighbor: txn
True Nearest neighbor: gis
distanceApproximate: 558.55
distanceTrue: 403.546
Query: zbh
Algorithm: Hypercube
Approximate Nearest neighbor: ph
True Nearest neighbor: ups
distanceApproximate: 1350.49
distanceTrue: 669.383
Query: zion
Algorithm: Hypercube
Approximate Nearest neighbor: ncr
True Nearest neighbor: sti
distanceApproximate: 1245.41
distanceTrue: 445.729
tApproximateAverage: 3342 microseconds
tTrueAverage: 37103 microseconds
MAF: 2.79409
```

Vector Hypercube

```
Query: xel
Algorithm: Hypercube
Approximate Nearest neighbor: adp
True Nearest neighbor: ed
distanceApproximate: 1.9617
distanceTrue: 1.62213
Query: xl
Algorithm: Hypercube
Approximate Nearest neighbor: mxim
True Nearest neighbor: fitb
distanceApproximate: 9.05043
distanceTrue: 3.9533
Query: xlnx
Algorithm: Hypercube
Approximate Nearest neighbor: mco
True Nearest neighbor: adi
distanceApproximate: 4.11979
distanceTrue: 1.84224
Query: xom
Algorithm: Hypercube
Approximate Nearest neighbor: flr
True Nearest neighbor: cvx
distanceApproximate: 7.82497
distanceTrue: 4.04647
Query: xray
Algorithm: Hypercube
Approximate Nearest neighbor: duk
True Nearest neighbor: t
distanceApproximate: 3.4706
distanceTrue: 3.38407
Query: xrx
Algorithm: Hypercube
Approximate Nearest neighbor: ati
True Nearest neighbor: rrd
distanceApproximate: 9.86862
distanceTrue: 3.44404
Query: yum
Algorithm: Hypercube
Approximate Nearest neighbor: mcd
True Nearest neighbor: orcl
distanceApproximate: 2.51684
distanceTrue: 2.48487
Query: zbh
Algorithm: Hypercube
Approximate Nearest neighbor: ca
True Nearest neighbor: mdt
distanceApproximate: 3.38861
distanceTrue: 3.0961
Query: zion
Algorithm: Hypercube
Approximate Nearest neighbor: key
True Nearest neighbor: key
distanceApproximate: 2.30908
distanceTrue: 2.30908
tApproximateAverage: 237 microseconds
tTrueAverage: 4746 microseconds
MAF: 2.86542
```

Vector Hypercube Compressed

Ξανά συγκρίνοντας τις δύο αναπαραστάσεις ξεχωρίζεται ο κοινός γείτονας “t” υποδεικνύοντας μας πως η συμπιεσμένη αναπαράσταση αποτελεί μια καλή προσέγγιση της αρχικής.

Αναζήτηση Χρονοσειρών με Discrete Frechet (k=8 , L=5)

Τα αποτελέσματα:

```
Query: xel
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: fitb
True Nearest neighbor: cms
distanceApproximate: 38.892
distanceTrue: 7.2957
Query: xl
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: mkc
True Nearest neighbor: sti
distanceApproximate: 72.1053
distanceTrue: 34.366
Query: xlnx
Algorithm: LSH_Frechet_Discrete
Query: xom
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: nue
True Nearest neighbor: pg
distanceApproximate: 52.6524
distanceTrue: 24.085
Query: xray
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: etr
True Nearest neighbor: pcg
distanceApproximate: 51.8454
distanceTrue: 15.8576
Query: xrx
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: jhg
True Nearest neighbor: kss
distanceApproximate: 29.9598
distanceTrue: 22.7972
Query: yum
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: gps
True Nearest neighbor: tmk
distanceApproximate: 53.9371
distanceTrue: 16.6603
Query: zbh
Algorithm: LSH_Frechet_Discrete
Query: zion
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: unh
True Nearest neighbor: sti
distanceApproximate: 167.14
distanceTrue: 16.953
tApproximateAverage: 1311267 microseconds
tTrueAverage: 453160309 microseconds
MAF: 9.85902
```

Time-Serie Discrete Frechet

```
Query: xel
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: cms
True Nearest neighbor: cms
distanceApproximate: 0.199831
distanceTrue: 0.199831
Query: xl
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: rf
True Nearest neighbor: rf
distanceApproximate: 0.600322
distanceTrue: 0.600322
Query: xlnx
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: a
True Nearest neighbor: a
distanceApproximate: 0.295518
distanceTrue: 0.295518
Query: xom
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: cvx
True Nearest neighbor: cvx
distanceApproximate: 0.427988
distanceTrue: 0.427988
Query: xray
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: apd
True Nearest neighbor: apd
distanceApproximate: 0.406797
distanceTrue: 0.406797
Query: xrx
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: rrd
True Nearest neighbor: rrd
distanceApproximate: 0.495628
distanceTrue: 0.495628
Query: yum
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: wat
True Nearest neighbor: wat
distanceApproximate: 0.327785
distanceTrue: 0.327785
Query: zbh
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: ca
True Nearest neighbor: ca
distanceApproximate: 0.369852
distanceTrue: 0.369852
Query: zion
Algorithm: LSH_Frechet_Discrete
Approximate Nearest neighbor: key
True Nearest neighbor: key
distanceApproximate: 0.354276
distanceTrue: 0.354276
tApproximateAverage: 37597657 microseconds
tTrueAverage: 8634456 microseconds
MAF: 1
```

Time-Serie Discrete Frechet Compressed

Ξανά συγκρίνοντας τις δύο αναπαραστάσεις ξεχωρίζεται ο κοινός γείτονας “cms” υποδεικνύοντας μας πως η συμπιεσμένη αναπαράσταση αποτελεί μια καλή προσέγγιση της αρχικής.

Αναζήτηση Χρονοσειρών με Continuous Frechet (k=8 , L=5)

Τα δειγματοληπτικά αποτελέσματα:

```
Query: xel
Algorithm: LSH_Frechet_Continuous
Approximate Nearest neighbor: aep
True Nearest neighbor: adp
distanceApproximate: 0.373937
distanceTrue: 0.365148
Query: xl
Algorithm: LSH_Frechet_Continuous
Approximate Nearest neighbor: afl
True Nearest neighbor: aee
distanceApproximate: 1.00081
distanceTrue: 0.765189
Query: xlnx
Algorithm: LSH_Frechet_Continuous
Query: xom
Algorithm: LSH_Frechet_Continuous
Approximate Nearest neighbor: aes
True Nearest neighbor: adm
distanceApproximate: 0.915197
distanceTrue: 0.675205
Query: xray
Algorithm: LSH_Frechet_Continuous
Approximate Nearest neighbor: aes
True Nearest neighbor: a
distanceApproximate: 0.913134
distanceTrue: 0.593819
Query: xrx
Algorithm: LSH_Frechet_Continuous
Query: yum
Algorithm: LSH_Frechet_Continuous
Query: zbh
Algorithm: LSH_Frechet_Continuous
Query: zion
Algorithm: LSH_Frechet_Continuous
Approximate Nearest neighbor: abc
True Nearest neighbor: aa
distanceApproximate: 1.40266
distanceTrue: 0.651523
tApproximateAverage: 11457641 microseconds
tTrueAverage: 203219369 microseconds
MAF: 2.1529
```

Time-Serie Continuous Frechet

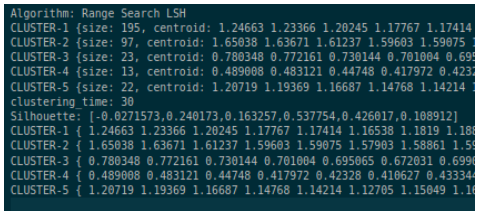
```
Query: xel
Algorithm: LSH_Frechet_Continuous
Query: xl
Algorithm: LSH_Frechet_Continuous
Approximate Nearest neighbor: afl
True Nearest neighbor: aee
distanceApproximate: 1.00081
distanceTrue: 0.765189
Query: xlnx
Algorithm: LSH_Frechet_Continuous
Query: xom
Algorithm: LSH_Frechet_Continuous
Approximate Nearest neighbor: aes
True Nearest neighbor: adm
distanceApproximate: 0.915197
distanceTrue: 0.675205
Query: xray
Algorithm: LSH_Frechet_Continuous
Approximate Nearest neighbor: aes
True Nearest neighbor: a
distanceApproximate: 0.913134
distanceTrue: 0.593819
Query: xrx
Algorithm: LSH_Frechet_Continuous
Query: yum
Algorithm: LSH_Frechet_Continuous
Approximate Nearest neighbor: adsk
True Nearest neighbor: aapl
distanceApproximate: 0.498318
distanceTrue: 0.426215
Query: zbh
Algorithm: LSH_Frechet_Continuous
Query: zion
Algorithm: LSH_Frechet_Continuous
Approximate Nearest neighbor: abc
True Nearest neighbor: aa
distanceApproximate: 1.40266
distanceTrue: 0.651523
tApproximateAverage: 7989246 microseconds
tTrueAverage: 204221841 microseconds
MAF: 2.1529
```

Time-Serie Continuous Frechet

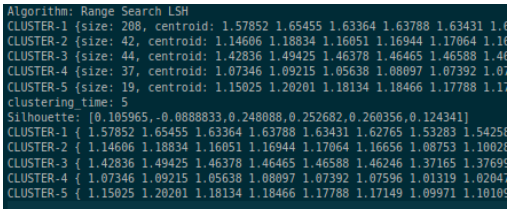
Πολλά ερωτήματα δεν επέστρεψαν κοντινότερο γείτονα εξ αιτίας του μικρού αριθμού δείγματος. Παρόλα αυτά αναγνωρίζονται πολλές κοινές χρονοσειρές ως κοντινότεροι γείτονες. Συγκεκριμένα οι afl, aee, aes, adm, a, aa, abc. Το μοντέλο φαίνεται αποδοτικό.

Συσταδοποίηση Διανυσμάτων με LSH

Τα αποτελέσματα:



Vector LSH Clustering

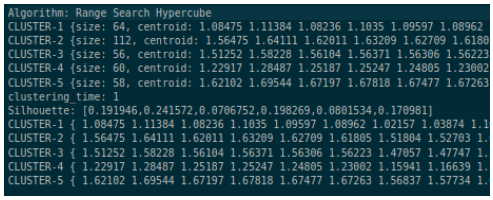


Vector LSH Clustering Compressed

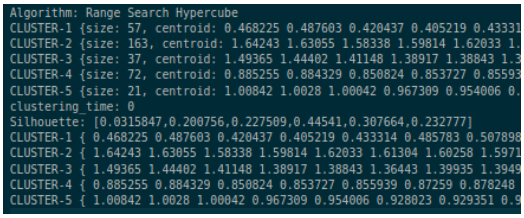
Τα κεντροειδή και τα περιεχόμενα των συστάδων δεν αναγράφονται γιατί αποτελούν μεγάλο όγκο δεδομένων. Τα αποτελέσματα είναι ικανοποιητικά. Η μετρική silhouette επιβεβαιώνει τη σωστή επιλογή των τεσσάρων συστάδων. Υπάρχουν αρκετά όμοια διανύσματα εντός των ίδιων συστάδων στις δύο αναπαραστάσεις.

Συσταδοποίηση Διανυσμάτων με Hypercube

Τα αποτελέσματα:



Vector Hypercube Clustering

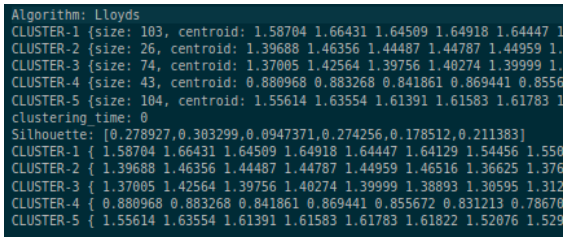


Vector Hypercube Clustering Compressed

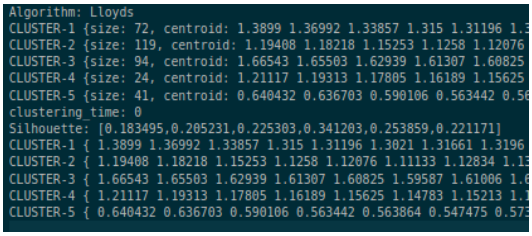
Όπως στη συσταδοποίηση διανυσμάτων με LSH, τα αποτελέσματα είναι ικανοποιητικά. Η μετρική silhouette επιβεβαιώνει τη σωστή επιλογή των τεσσάρων συστάδων. Υπάρχουν αρκετά όμοια διανύσματα εντός των ίδιων συστάδων στις δύο αναπαραστάσεις.

Συσταδοποίηση Διανυσμάτων με Lloyds

Τα αποτελέσματα:



Vector Lloyds Clustering

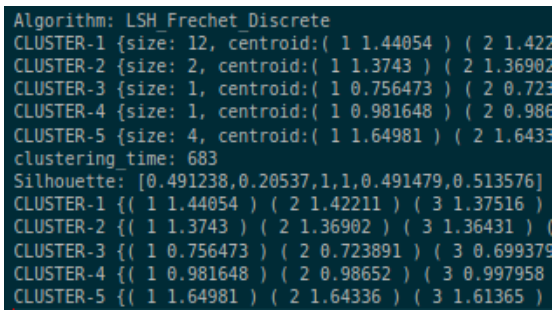


Vector Lloyds Clustering Compressed

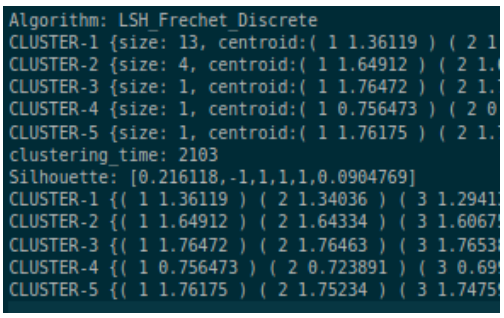
Τα μεγέθη εντός των συστάδων διαφέρουν ελαφρώς. Παρόλα αυτά υπάρχουν αρκετά κοινά διανύσματα εντός αυτών.

Συσταδοποίηση Χρονοσειρών με LSH Discrete Frechet

Τα δειγματοληπτικά αποτελέσματα:



Time-Serie LSH Discrete Frechet Clustering

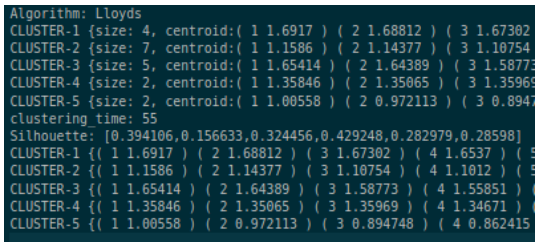


Time-Serie LSH Discrete Frechet Clustering Compressed

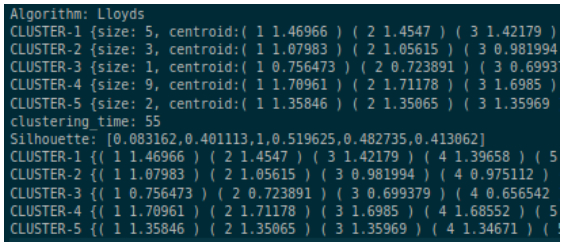
Τα μεγέθη των συστάδων συγκλίνουν στους ίδιους αριθμούς. Υπάρχουν αρκετές κοινές χρονοσειρές εντός των συστάδων των δύο αναπαραστάσεων.

Συσταδοποίηση Χρονοσειρών με Lloyds Discrete Frechet

Τα δειγματοληπτικά αποτελέσματα:



Time-Serie Lloyds Discrete Frechet Clustering



Time-Serie Lloyds Discrete Frechet Clustering Compressed

Τα μεγέθη των συστάδων διαφέρουν σχετικά με τις προηγούμενες μετρικές. Ωστόσο κάποιες χρονοσειρές είναι κοινές μεταξύ των συστάδων.

Συμπεράσματα

Οι πειραματικές εκτελέσεις των δύο αναπαραστάσεων επιφέρουν άρτια αποτελέσματα κατά τη συσταδοποίηση. Στη πλειοψηφία των περιπτώσεων υπάρχουν αρκετά κοινά μέλη μεταξύ των συστάδων των δύο αναπαραστάσεων. Η συγκεκριμένη συμπεριφορά δεν αναγνωρίζεται εύκολα κατά την αναζήτηση του πλησιέστερου γείτονα. Η αναγνώριση θα ήταν ευκολότερη στην εμφάνιση των Ν πλησιέστερων γειτόνων. Η πιθανότητα όμοιου κοντινότερου γείτονα δεν κυριαρχεί, συνεπώς κατά τους Ν κοντινότερους γείτονες θα υπάρχουν αρκετές κοινές αναφορές.