α) Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα - Εργασία 3

Η εργασία αποτελείται από τέσσερα μέρη. Στο πρώτο υλοποιήθηκε το ζητούμενο LSTM μοντέλο για την πρόγνωση τιμών μετοχών. Το πρόγραμμα αφού διαβάσει τις παραμέτρους, δημιουργεί τα σύνολα εκπαίδευσης και ελέγχου και προχωρά στην δημιουργία των X και y πινάκων με βάση τις τιμές υστέρησης. Στην συνέχεια πραγματοποιείται η εκπαίδευση του μοντέλου και η πρόγνωση των χρονοσειρών που έχουν επιλεχθεί. Υποστηρίζεται εκπαίδευση ανά χρονοσειρά και ανά συνόλου χρονοσειρών. Επίσης μπορεί ο χρήστης να προχωρήσει απευθείας στην πρόγνωση τιμών με χρήση παραμέτρου. Αφού εκτυπωθούν τα σφάλματα, παράγονται οι γραφικές παραστάσεις για την εκτίμηση του μοντέλου.

Το δεύτερο μέρος αφορά το LSTM μοντέλο για την ανίχνευση ανωμαλιών. Το πρόγραμμα αφού διαβάσει τις παραμέτρους, δημιουργεί τα σύνολα εκπαίδευσης και ελέγχου και προχωρά στην δημιουργία των X και y πινάκων με βάση τις τιμές υστέρησης. Στην συνέχεια πραγματοποιείται η εκπαίδευση του μοντέλου και η πρόγνωση των χρονοσειρών που έχουν επιλεχθεί. Επίσης μπορεί ο χρήστης να προχωρήσει απευθείας στην πρόγνωση τιμών με χρήση παραμέτρου. Με βάση την πρόγνωση παρατηρείται η σχέση του σφάλματος πρόγνωσης (test loss) με το κατώφλι μέσου απόλυτου σφάλματος που έχει επιλεχθεί. Αφού εκτυπωθούν τα σφάλματα, παράγονται οι γραφικές παραστάσεις για την εκτίμηση του μοντέλου.

Το τρίτο μέρος αφορά το συνελικτικό μοντέλο για την μείωση της πολυπλοκότητας χρονοσειρών. Το πρόγραμμα αφού διαβάσει τις παραμέτρους, δημιουργεί τα σύνολα εκπαίδευσης και ελέγχου και προχωρά στην δημιουργία των X πινάκων με βάση το ‘παράθυρο’ χρονικών στιγμών. Στην συνέχεια πραγματοποιείται η εκπαίδευση του μοντέλου και η πρόγνωση των χρονοσειρών που έχουν επιλεχθεί. Επίσης μπορεί ο χρήστης να προχωρήσει απευθείας στην πρόγνωση τιμών με χρήση παραμέτρου. Στο συγκεκριμένο μοντέλο η πρόγνωση πάνω στο test set χρησιμοποιείται αποκλειστικά για την αξιολόγηση του μοντέλου και δεν φέρει κάποια άλλα χρησιμότητα. Αφού εκτυπωθούν τα σφάλματα, παράγονται οι μειωμένες χρονοσείρες και αποθηκεύονται στα αρχεία που έχει επιλέξει ο χρήστης.

Με βάση τα αρχεία που παράγονται από το τρίτο μοντέλο, εκτελούνται οι αλγόριθμοι της δεύτερης εργασίας και υλοποιείται έτσι και το τέταρτο μέρος.

Σχεδιαστικές επιλογές:

Παραθέτουμε τις παραδοχές και επεκτάσεις υλοποίησης που πήραμε για κάποια τμήματα των προγραμμάτων και αιτιολόγηση αυτών.

1) Το μοντέλο του Α ερωτήματος αποθηκεύεται για εύκολη επαναχρήση και για να μην χάνεται χρόνος στην εκπαίδευση. Αυτό ισχύει βέβαια μόνο στην εκπαίδευση ανά σύνολο μιας και στην περίπτωση του ανά χρονοσειρά, γίνονται απευθείας οι προγνώσεις οπότε δεν είχε ιδιαίτερο νόημα η αποθήκευση. Επίσης θα έπρεπε σε αυτή την περίπτωση να αποθηκεύσουμε υπερβολικά πολλά μοντέλα.

2) Το μοντέλο του Β ερωτήματος αποθηκεύεται για εύκολη επαναχρήση και για να μην χάνεται χρόνος στην εκπαίδευση. Επίσης στο ερώτημα αυτό ο διαχωρισμός των συνόλων εκπαίδευσης και ελέγχου γίνεται με βάση του dataset, γιατί παρατηρήσαμε ότι έδινε κάπως πιο ισορροπημένα σφάλματα στο test set. Στο Α και Γ ερώτημα ο διαχωρισμός γίνεται ανά χρονοσειρά. Όπως έδειχναν και τα tutorials του φροντιστηρίου.

3) Στο Γ ερώτημα η εκπαίδευση γίνεται επαναληπτικά αλλά ο ορισμός του μοντέλου είναι εκτός επανάληψης. Αποφασίσαμε ότι είναι μια πιο σώστη επιλογή στο μοντέλο αυτό μιας και ως είσοδο δέχεται το input window, που αφορά μια χρονοσειρά. Στο e-class είχε ειπωθεί ότι αυτή η προσέγγιση είναι σωστή εφόσον το μοντέλο εκπαιδεύεται ‘αυξητικά’, και άρα είναι ανά σύνολο το training. Στο Α και Β, η εκπαίδευση γίνεται με ένα fit μετά από ένωση των χρονοσειρών. Επίσης, γίνεται αποθήκευση του μοντέλου, όπως και στα άλλα ερωτήματα.

4) Στο Γ ερώτημα παρότι αναφέρει η εκφώνηση να δώσουμε το window και την διάσταση του latent vector ως παραμέτρους, αυτές οι τιμές κατέληξαν να είναι hardcoded. Ο λόγος που έγινε αυτό είναι γιατί αυτές οι τιμές επηρεάζουν την δομή του νευρωνικού με αποτέλεσμα να μην έχει ελευθερία ο χρήστης σε αυτό το κομμάτι. Σε κάθε περίπτωση, πειραματιστήκαμε με διάφορες τιμές πριν καταλήξουμε σε αυτές του παραδοτέου. Το ίδιο ισχύει και για τις τιμές υστέρησης των Α και Β ερωτημάτων.

5) Σε όλα τα ερωτήματα εκτυπώνονται τα validation losses μετά το training (το validation είναι το 0.1 του training set) και τα test losses μετά τις προγνώσεις για την αξιολόγηση των μοντέλων.

β) Κατάλογος αρχείων κώδικα και περιγραφή τους

forecast.py // Python script για την πρόγνωση τιμών του Α ερωτήματος.

detect.py // Python script για την ανίχνευση ανωμαλιών του Β ερωτήματος.

reduce.py // Python script για την μείωση πολυπλοκότητας χρονοσειρών του Γ ερωτήματος.

Έχουμε παραδώσει επίσης τα αποθηκευμένα μοντέλα (ίδια ονόματα του αντίστοιχου .py σε κατάληξη .h5) καθώς και ενδεικτικά αποτελέσματα για το Δ ερώτημα στους φακέλους nearest_neighbor_results και clustering_results.

γ) Οδηγίες χρήσης του προγράμματος

Α) Το πρόγραμμα εκτελείται με το script: forecast.py

Πέρα από τις παραμέτρους για τον αριθμό χρονοσειρών (default = 1) και του ονόματος του αρχείου, παρέχουμε τις εξής δικές μας παραμέτρους: -total (boolean, για την εκπαίδευση ανά σύνολο, default είναι False), -predict (boolean, για την απευθείας πρόγνωση, default είναι False), -graphs_n (int, για να σχεδιαστούν λιγότερες από n γραφικές παραστάσεις, default είναι 1) και -graphs_all (boolean, για την σχεδίαση όλων των γραφικών παραστάσεων, default είναι False). Ακολουθούν ενδεικτικές εκτελέσεις του script:

python forecast.py -d nasdaq2007_17.csv -n 5 -graphs_all -predict
python forecast.py -d nasdaq2007_17.csv -n 5 -graphs_all -total -predict

Β) Το πρόγραμμα εκτελείται με το script: detect.py

Πέρα από τις παραμέτρους για τον αριθμό χρονοσειρών (default = 1), του μέσου απόλυτου σφάλματος (default = 0.09) και του ονόματος του αρχείου, παρέχουμε τις εξής δικές μας παραμέτρους: -predict (boolean, για την απευθείας πρόγνωση, default είναι False), -graphs_n (int, για να σχεδιαστούν λιγότερες από n γραφικές παραστάσεις, default είναι 1), -graphs_all (boolean, για την σχεδίαση όλων των γραφικών παραστάσεων, default είναι False) και -loss_plot (boolean, για την σχεδίαση των γραφικών παραστάσεων των test losses, default είναι False). Ακολουθούν ενδεικτικές εκτελέσεις του script:

python detect.py -d nasdaq2007_17.csv -n 5 -loss_plot -mae 0.09 -graphs_all -predict

Γ) Το πρόγραμμα εκτελείται με το script: reduce.py

Πέρα από τις παραμέτρους για τα ονόματα των αρχείων, παρέχουμε τις εξής δικές μας παραμέτρους: -predict (boolean, για την απευθείας πρόγνωση, default είναι False). Ακολουθούν ενδεικτικές εκτελέσεις του script:

python reduce.py -d nasdaq2007_17_input.csv -q nasdaq2007_17_query.csv -od reduced_nasd_input.csv -oq reduced_nasd_query.csv -predict

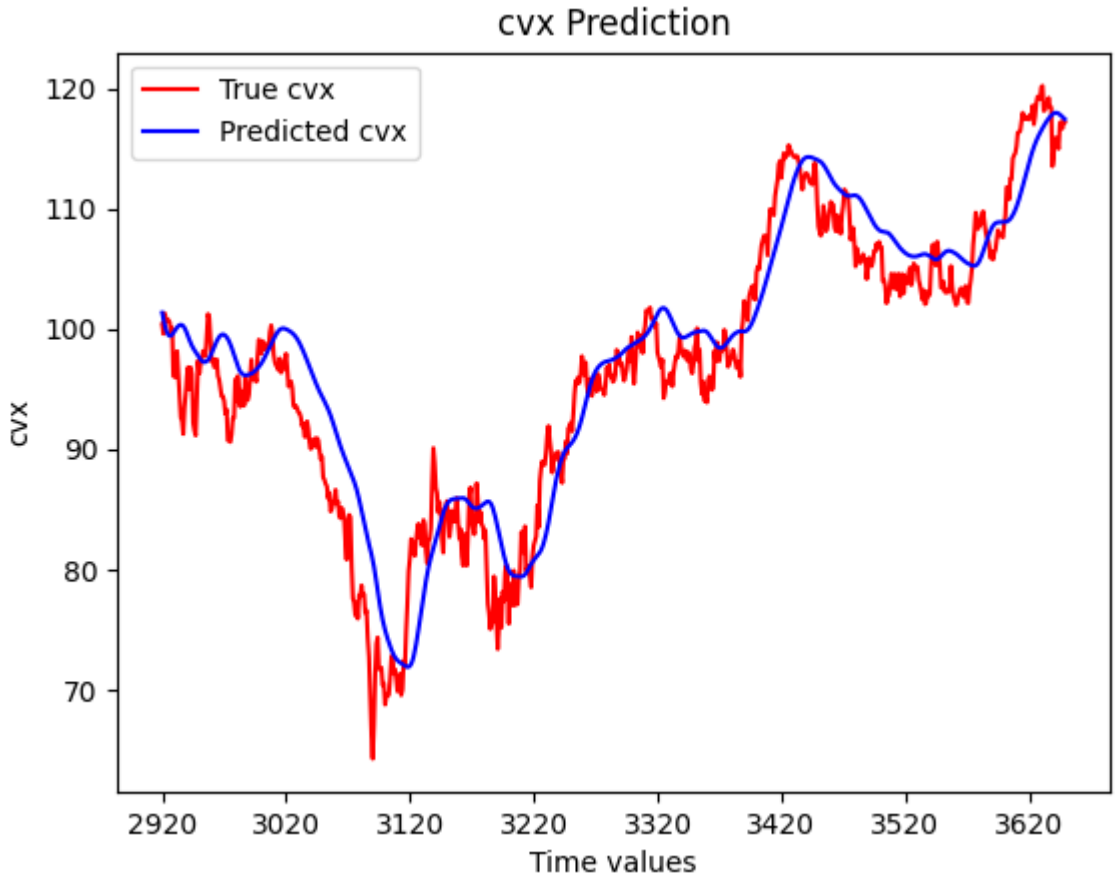
Δ) Το παραδοτέο της δεύτερης εργασίας δεν έχει τροποποιηθεί, οπότε ισχύουν οι οδηγίες χρήσης του προηγούμενου README.

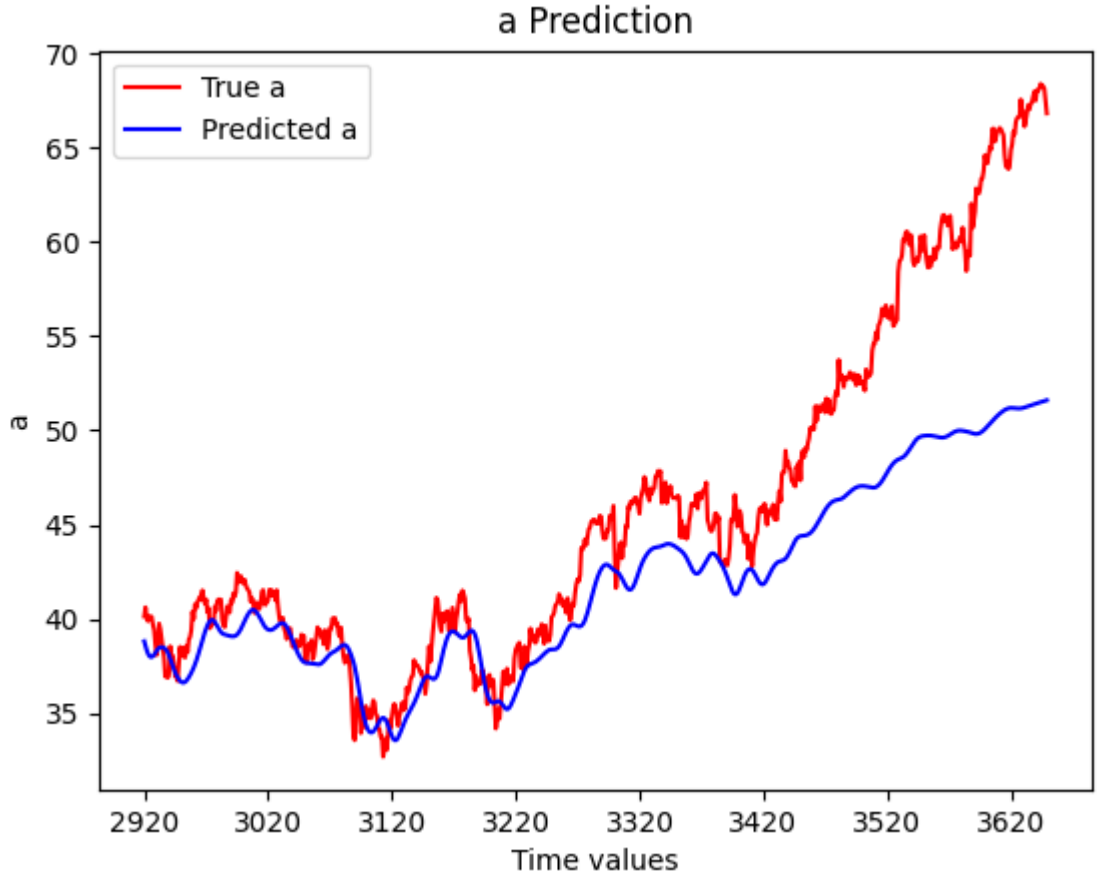
δ) Σχολιασμοί αποτελεσμάτων

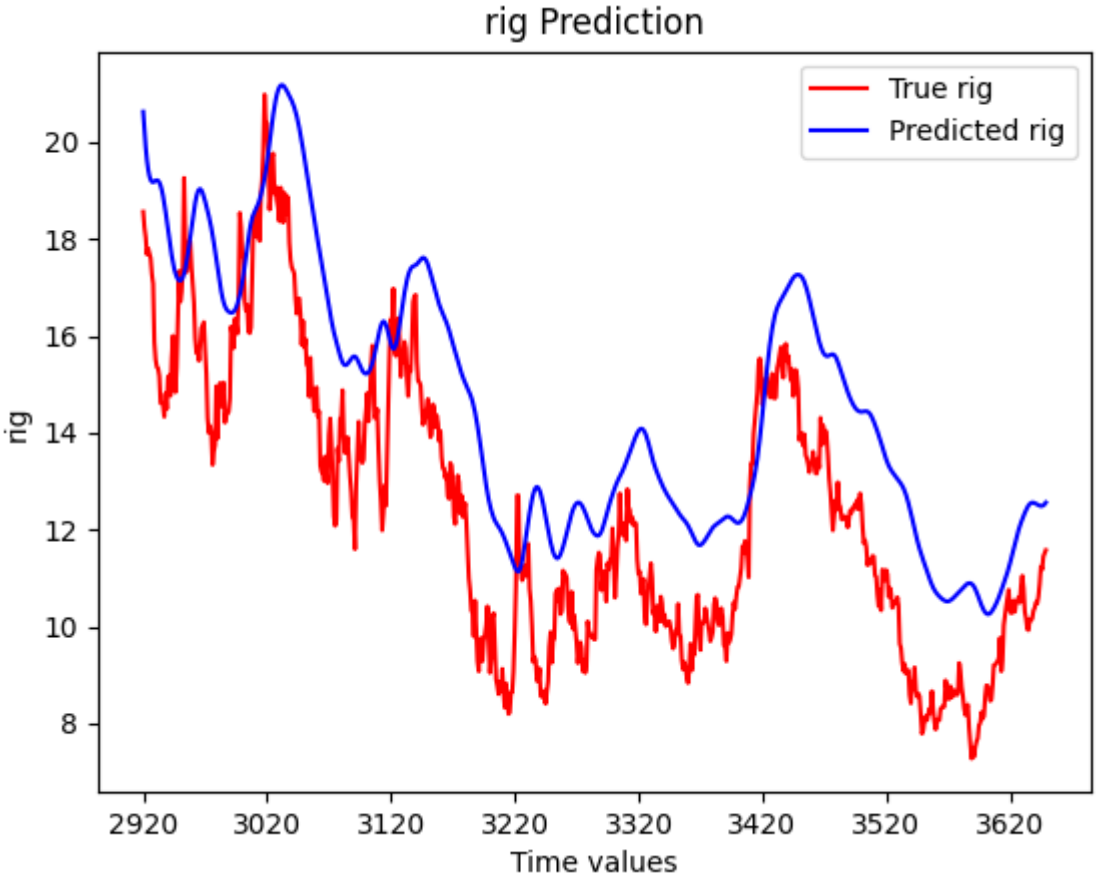
Α)

- Εκπαίδευση ανά χρονοσειρά

Ακολουθούν τρεις ενδεικτικές γραφικές παραστάσεις από μια εκτέλεση του προγράμματος.

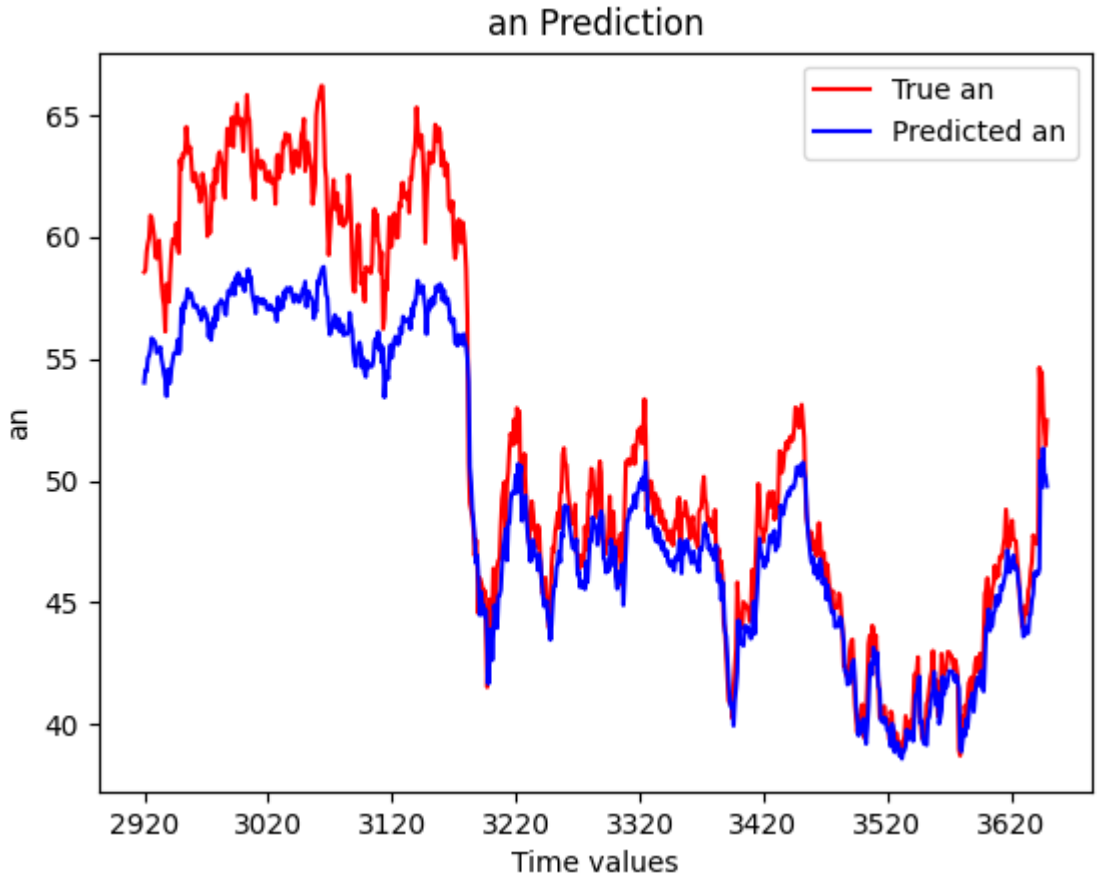

cvx Prediction
True cvx
Predicted cvx
120
110
100
90
80
70
cvx
2920
3020
3120
3220
3320
3420
3520
3620
Time values

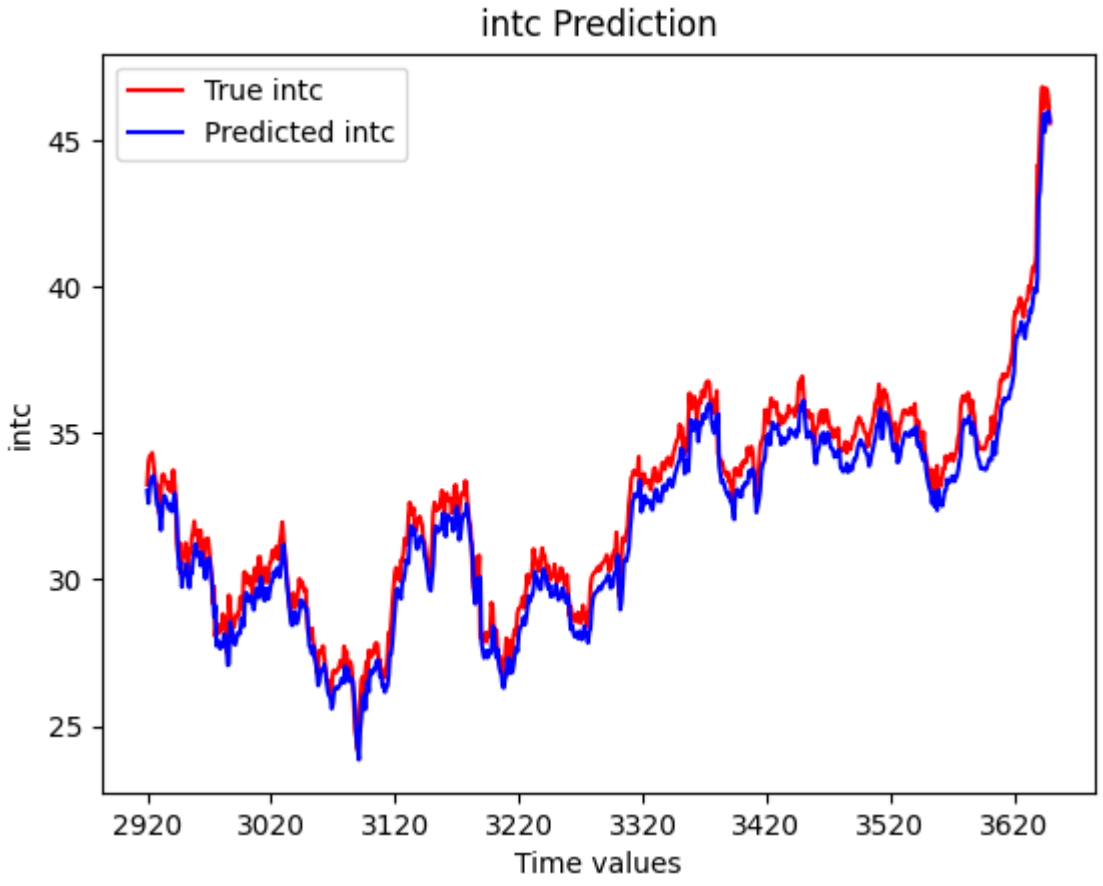

a Prediction
True a
Predicted a
70
65
60
55
50
45
40
35
a
2920
3020
3120
3220
3320
3420
3520
3620
Time values

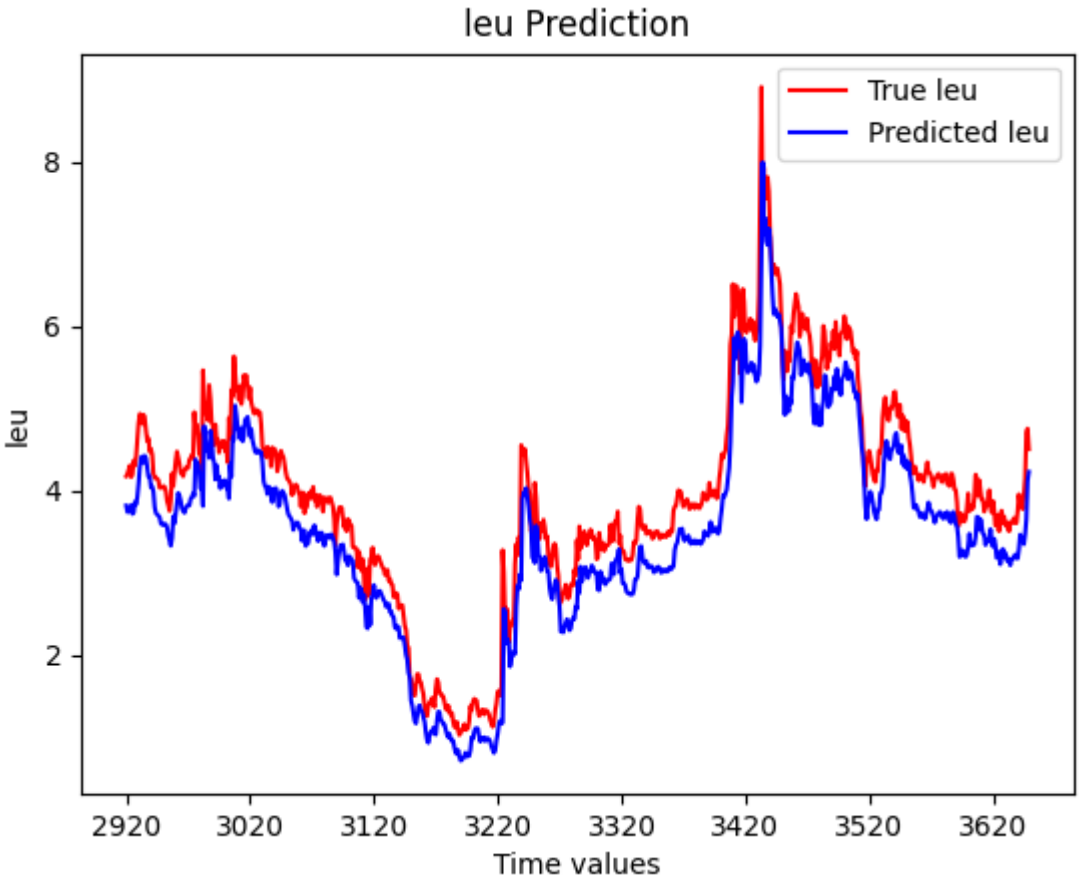


- Εκπαίδευση ανά σύνολο

Ακολουθούν τρεις ενδεικτικές γραφικές παραστάσεις από μια εκτέλεση του προγράμματος.


an Prediction
True an
Predicted an
an
65
60
55
50
45
40
2920
3020
3120
3220
3320
3420
3520
3620
Time values


intc Prediction
True intc
Predicted intc
intc
45
40
35
30
25
2920
3020
3120
3220
3320
3420
3520
3620
Time values



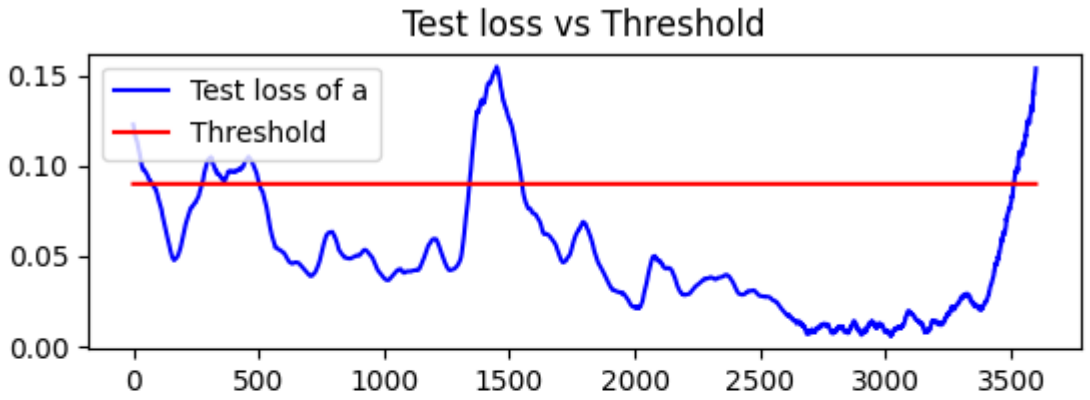
Από τις γραφικές παραστάσεις μπορούμε να συμπεράνουμε τα εξής: Αρχικά βλέπουμε ότι και στις δύο περιπτώσεις οι predicted χρονοσειρές μοιάζουν αρκετά με τις αντίστοιχες πραγματικές. Υπενθυμίζουμε σε αυτό το σημείο ότι προβλέπουμε το τελικό 20% για κάθε χρονοσειρά, μιας και το υπόλοιπο 80% χρησιμοποιείται για training. Αν και οι προβλέψεις και στις δύο περιπτώσεις φαίνονται να είναι αρκετά ακριβείς, στην περίπτωση της εκπαίδευσης ανά σύνολο οι predicted φαίνονται να είναι πιο κοντά στις πραγματικές σε σχέση με τις εκπαιδευμένες ανά χρονοσειρά. Σε αυτό πιθανότατα οφείλεται από το γεγονός ότι το μοντέλο στο ανά σύνολο λαμβάνει υπόψη όλες τις χρονοσειρές και άρα μπορεί να κάνει πιο ακριβείς προβλέψεις. Στο μοντέλο ανά χρονοσειρά γίνεται με μόνο μία, αλλά ακόμη και εκεί βλέπουμε ότι το γενικό outline είναι αρκετά κοντά στην πραγματική.

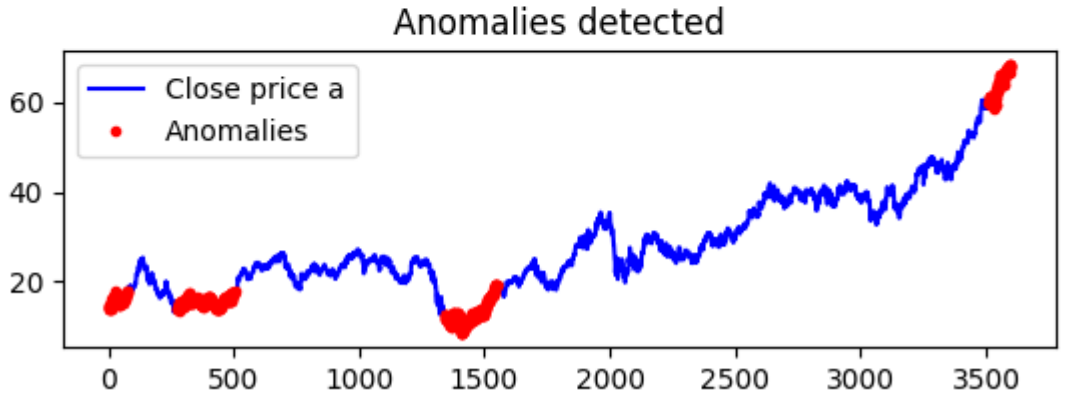
Ως προς τις παραμέτρους των μοντέλων είδαμε όλα όσα ζητούσε η εκφώνηση. Συγκεκριμένα, σχετικά με τον αριθμό στρωμάτων και το μέγεθος αυτών είδαμε πως όσο μεγαλώνουν, άλλο τόσο καθυστερεί η εκτέλεση το οποίο είναι και αναμενόμενο μιας και επιπλέον νευρώνες δημιουργούν ένα πιο πολύπλοκο μοντέλο. Φυσικά το ίδιο ισχύει και για τα epochs, για αυτό και δεν επιλέξαμε μεγάλο αριθμό. Ούτως ή αλλως, και μικρός αριθμός epochs έδιναν πίσω μια καλή απόδοση όπως φαίνεται από τις γραφικές παραστάσεις. Το batch size γενικά επιλέξαμε να το αυξήσουμε για να επιταχύνουμε την εκτέλεση του μοντέλου. Φυσικά, δεν πρέπει το batch size να αυξηθεί πολύ γιατί ναι μεν σε κάθε epoch γίνεται επεξεργασία από λιγότερα batch, αλλά αυτό επηρεάζει και τα accuracy scores. Επίσης προσθέσαμε και dropout layers μιας και βελτιστοποιούσαν τις προβλέψεις.

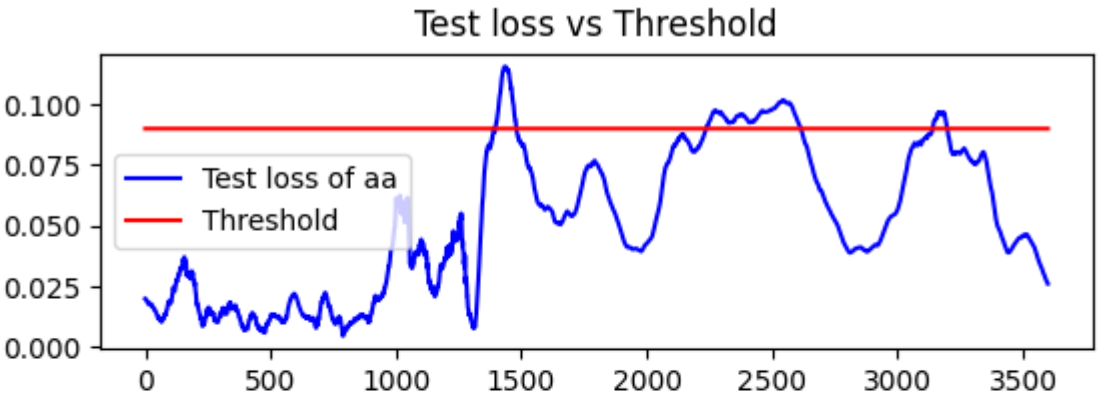
Το τελικό μοντέλο στο ανά χρονοσειρά έχει: 4 στρώματα, 64 unit το καθένα, 7 epochs, 128 batch size και 0.7 dropout.
Το τελικό μοντέλο στο ανά σύνολο έχει: 2 στρώματα, 64 unit το καθένα, 5 epochs, 512 batch size και 0.5 dropout.

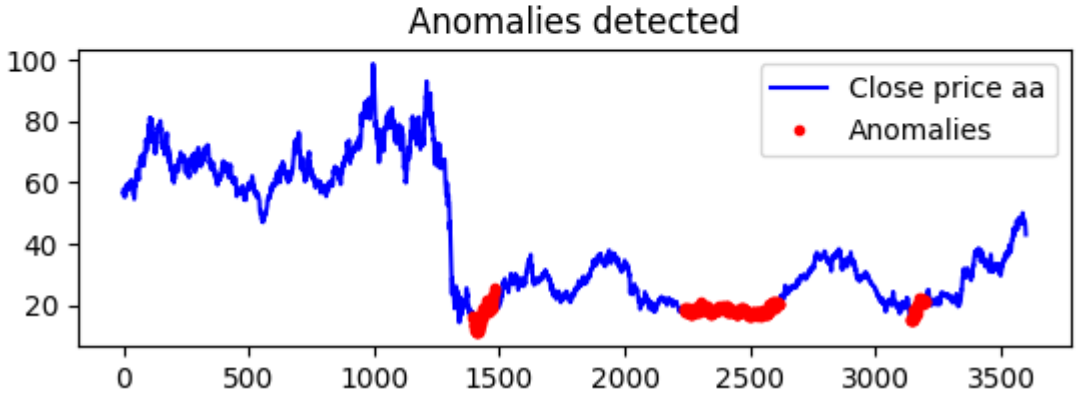
Τέλος, επιλέξαμε οι χρονικές τιμές υστέρησης να είναι 50. Αυτός ο αριθμός θεωρήσαμε ότι ήταν κατάλληλος για τις χρονοσειρές που επεξεργαζόμαστε.

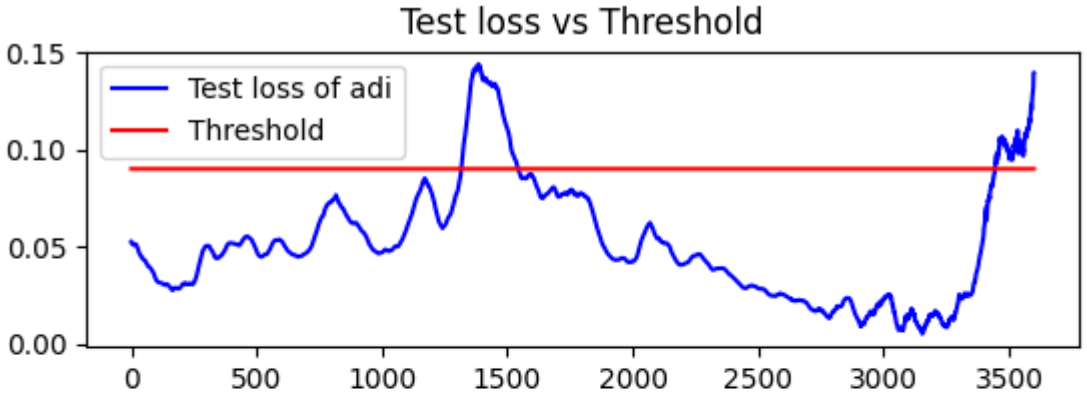
Β) Ακολουθούν τρεις ενδεικτικές γραφικές παραστάσεις από μια εκτέλεση του προγράμματος.

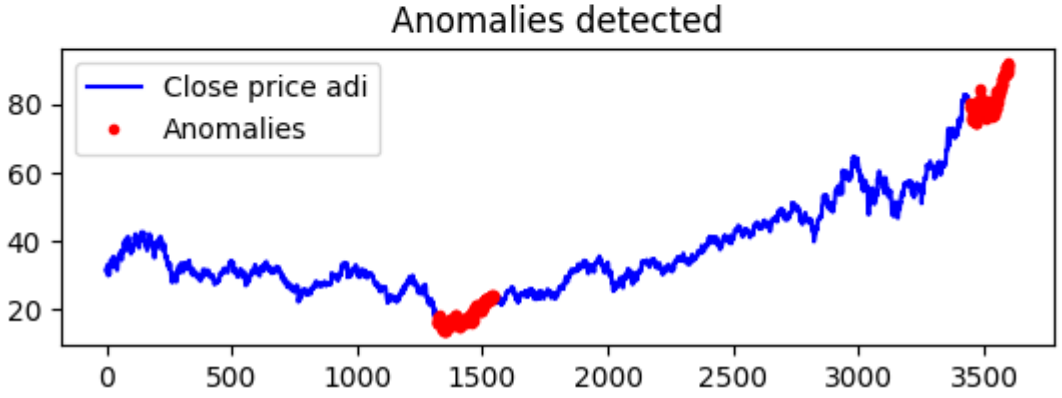





Test loss vs Threshold
Test loss of aa
Threshold
0.100
0.075
0.050
0.025
0.000
0
500
1000
1500
2000
2500
3000
3500


Anomalies detected
Close price aa
Anomalies
100
80
60
40
20
0
500
1000
1500
2000
2500
3000
3500


Test loss vs Threshold
Test loss of adi
Threshold
0.15
0.10
0.05
0.00
0
500
1000
1500
2000
2500
3000
3500


Anomalies detected
Close price adi
Anomalies
80
60
40
20
0
500
1000
1500
2000
2500
3000
3500

Από τις γραφικές παραστάσεις μπορούμε να συμπεράνουμε τα εξής: Αρχικά το test loss, μας βοηθάει να κρίνουμε πως συμπεριφέρεται το μοντέλο καθώς και η χρονοσειρά καθ' αυτή. Γενικά παρατηρήσαμε ότι στις πιο πολλές χρονοσειρές το test loss ξεκινάει αρκετά ψηλά, κάτι που είναι λογικό μιας και το μοντέλο είναι στην αρχή ακόμα και θέλει επιπλέον χρόνο για να ισορροπήσει το loss. Αφού γίνει αυτό και αρχίσει να πέφτει, στην πορεία βλέπουμε να ξανα να ανεβαίνει σε κάποια διαστήματα και μετά να ξαναπέφτει. Αυτά τα διαστήματα συμπίπτουν με τις αντίστοιχες αυξομειώσεις της χρονοσειράς. Άμα η χρονοσειρά εμφανίσει μια απότομη αλλαγή, θα δούμε και στο test loss μια αντίστοιχη συμπεριφορά. Μιας και έχουν αλλάξει τα δεδομένα που ήξερε εώς πριν το μοντέλο παίρνει λίγο χρόνο για να αρχίσει να ξανά έχει ένα σταθερό loss.

Αποφασίσαμε λοιπόν με βάση το test loss να επιλέξουμε το αντίστοιχο κατώφλι για το MAE/threshold, ένα που να είναι κοινό για τις πιο πολλές χρονοσειρές. Η θεωρία μας ήταν πως αν το loss αυξάνεται πάρα πολύ, αυτό σημαίνει ότι αρχίζουμε και έχουμε ανωμαλίες μιας και στην αντίστοιχη χρονοσειρά συνήθως υπάρχουν απότομες αυξομειώσεις στα αντίστοιχα σημεία. Όπως βλέπουμε από τις παραστάσεις, το μοντέλο φαίνεται να έχει ανιχνεύσει τις αντίστοιχες αυξομειώσεις, δηλαδή τις ανωμαλίες που ψάχνουμε. Για παράδειγμα, στην τελευταία χρονοσειρά οι τιμές που είναι προς το τέλος φαίνονται να είναι αρκετά ψιλά σε σχέση με την υπόλοιπη χρονοσειρά και άρα έχουν επισημανθεί ως ανωμαλίες. Φυσικά, το μοντέλο δεν είναι και 100% ακριβές μιας και το test loss μπορεί να αυξηθεί και για άλλους λόγους, όπως στην αρχή της χρονοσειράς για τους λόγους που εξηγήσαμε προηγουμένως. Το μοντέλο έχει ανιχνεύσει τα αντίστοιχα σημεία ως ανωμαλίες, κάτι που μπορούμε να θεωρήσουμε ως μια μικρή αστοχία.

Για τις παραμέτρους του νευρωνικού, ισχύουν λίγο πολύ όσα είπαμε και για το Α ερώτημα. Γενικά, μιας και το πρόβλημα ήταν παρόμοιας φύσεως, είπαμε να ακολουθήσουμε μια κοινή γραμμή, ειδικά μιας και η εκπαίδευση ανά σύνολο καθυστερεί αρκετά. Το τελικό μοντέλο έχει: 2 στρώματα, 128 unit το καθένα, 5 epochs, 512 batch size και 0.2 dropout.

Οι χρονικές τιμές υστέρησης να είναι 50 για τον ίδιο λόγο με το Α ερώτημα.

Γ)

Αυτό που παρατηρήσαμε κατά την ανάπτυξη του νευρωνικού ήταν ότι η εκπαίδευση γινόταν μακράν πιο γρήγορα σε σχέση με τα προηγούμενα ερωτήματα, κάτι το οποίο διευκόλυνε τον πειραματισμό με τις παραμέτρους. Από την άλλη βέβαια το μέγεθος του παραθύρου και το latent dim επηρεάζουν την δομή από όλο το νευρωνικό, όποτε πρέπει να αλλάζει καταλλήλως και ο κώδικας.

Σε κάθε περίπτωση, η γενική μας ιδέα ήταν να μειώσουμε αρκετά την πολυπλοκότητα για να δούμε τις διαφορές που θα προκύψουν σχετικά με την κανονική χρονοσειρά. Με τις default τιμές, window=10 και latent dim=3, έχουμε μια χρονοσειρά μεγέθους 1095 (για το nasdaq2007_17.csv του e-class), περίπου δηλαδή το 1/3 της κανονικής. Είδαμε ότι γενικά είχαμε καλά αποτελέσματα για το Δ, αλλά για να πειραματιστούμε λίγο παραπάνω καταλήξαμε σε window=20 και latent dim=5, άρα τελικό μέγεθος 910.

Το τελικό μοντέλο λοιπόν έχει: 2 συνελικτικά στρώματα, 32 μέγεθος το καθένα, 5 epochs και 128 batch size.

Δ)

Χρησιμοποιήσαμε τελικά window=20 και latent dim=5, άρα τελικό μέγεθος 910, για τις μειωμένες χρονοσειρές. Ως input και query set βασιστήκαμε στο nasdaq2007_17.csv από το οποίο επιλέξαμε τις τελευταίες 10 χρονοσειρές για query set και τις υπόλοιπες για input. Εκτελέσαμε λοιπόν τους αλγόριθμους αρχικά με τις κανονικές χρονοσειρές για να έχουμε σημεία αναφοράς και στην συνέχεια κάναμε και τις εκτελέσεις για τις μειωμένες χρονοσειρές. Σε αυτό το σημείο να σημειωθεί πως χρησιμοποήσαμε το query set στο clustering με Frechet γιατι το input αργούσε πάρα πολύ (πάνω από ώρα), οπότε επιλέξαμε κάτι μικρότερο για να έχουμε έστω μια εικόνα για το clustering με Frechet. Τα σχετικά αποτελέσματα από τις εκτελέσεις μπορείτε να τα δείτε στους φακέλους nearest_neighbor_results και clustering_results του παραδοτέου.

Καταλήγουμε στις εξής παρατηρήσεις: Σαφέστατα, ο χρόνος εκτέλεσης είναι γενικά μικρότερος κάτι το οποίο ήταν αναμενόμενο μιας και τα χρονικά σημεία είναι πολύ λιγότερα. Στους αλγόριθμους για nearest neighbor βλέπουμε επίσης ότι έχουμε αρκετά καλά MAF scores. Βέβαια, παρατηρούμε από την άλλη ότι αρκετοί πραγματικοί γείτονες άλλαξαν στις ‘μειωμένες’ εκτελέσεις, κάτι το οποίο είναι λογικό μιας και έχουν τροποποιηθεί αρκετά οι χρονοσειρές μας. Βέβαια κάτι το οποίο είναι ενδιαφέρον είναι ότι στους γείτονες που έμειναν ίδιοι και στις δύο αναπαραστάσεις, βλέπουμε ότι γενικά βρέθηκαν καλύτεροι approximate γείτονες στις μειωμένες χρονοσειρές. Αυτό πιθανότατα οφείλεται στο γεγονός ότι τα χρονικά σημεία είναι λιγότερα, κάτι το οποίο ‘διευκολύνει’ τους αλγόριθμους αναζήτησης. Αξίζει να σημειωθεί λοιπόν ότι παρά τις αλλαγές που κάνουμε στις χρονοσειρές, αυτό δεν επηρεάζει απαραίτητα την ακρίβεια των αποτελεσμάτων αναζήτησης μιας και όπως είδαμε σε κάποιες περιπτώσεις είναι ακόμη και καλύτερα από τα αντίστοιχα των κανονικών.

Στο clustering βλέπουμε ότι και εδώ ο χρόνος εκτέλεσης είναι αρκετά μικρότερος. Επίσης παρατηρούμε και εδώ παρά την μειωμένη πολυπλοκότητα έχουμε περιπτώσεις με μια αρκετά καλή ακρίβεια σε σχέση με τις πραγματικές τιμές. Αυτό φαίνεται και σε κάποια silhouette scores αλλά και με την χρήση της complete παραμέτρου βλέπουμε clusters τα οποία έχουν κοινά στοιχεία και στις κανονικές αλλά και στις μειωμένες αναπαραστάσεις. Μπορούμε να δούμε ένα χαρακτηριστικό παράδειγμα στα αποτελέσματα του clustering με LSH και Mean Vector update step: Ο Cluster-3 των πραγματικών και ο Cluster-4 των μειωμένων διαφέρουν κατά ένα μόλις στοιχείο.

Ως γενικό συμπέρασμα βλέπουμε ότι η επιλογή μας μείωσε αρκετά τους χρόνους εκτέλεσης αλλά επηρέασε σημαντικά τα αποτελέσματα από άποψη ακρίβειας. Όμως, ακόμη και με το ~1/3 της κανονικής χρονοσειράς ορισμένοι αλγόριθμοι βρήκαν αρκετα κοινά σημεία με τις κανονικές. Φυσικά όπως είναι λογικό όσο θα αυξάνει η διάσταση θα έχουμε καλύτερα αποτελέσματα αλλά σε μεγαλύτερο χρόνο.

ε) Ονοματεπώνυμα και ΑΜ:
Παντελεήμων Μαλέκας 1115201600268
Θεοφάνης Μπιρμπίλης 1115201600110

Η εργασία αναπτύχθηκε με χρήση Git και είναι διαθέσιμη στο repository: https://github.com/pantmal/Algorithmic-Problems-Project-3. Επίσης χρησιμοποίηθηκε και το extension Live Share του VS Code για την ταυτόχρονη συγγραφή κώδικα. Τέλος, αξιοποιήσαμε και το Google Colab για την δυνατότητα εκτέλεσης των προγραμμάτων μας με GPU, προκειμένου να επιταχύνουμε την εκπαίδευση των μοντέλων μας.