ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

"Πρόβλεψη τιμής χρηματιστηρίου με τεχνικές μηχανικής μάθησης μέσω ανάλυσης συναισθημάτων του Twitter."

ΠΑΝΑΓΙΩΤΟΠΟΥΛΟΣ ΠΑΝΑΓΙΩΤΗΣ

ΕΠΙΒΛΕΠΩΝ: Ιωάννης Τζήμας

Πάτρα, 2022

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Πάτρα, 21/11/2022

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Ιωάννης Τζήμας

2. Ιωάννης Τσακνάκης

3. Παρασκευάς Κίτσος

Υπεύθυνη δήλωση Φοιτητή

Βεβαιώνω ότι είμαι συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τη συγκεκριμένη εργασία. Η έγκριση της διπλωματικής εργασίας από το Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Πελοποννήσου δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος. Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Παναγιωτόπουλου Παναγιώτη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο Πανεπιστήμιο Πελοποννήσου, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οποιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.

Περιεχόμενα

ΠΙΝΑΚΑΣ ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ

LSTM	Long short-term memory
DNN	Deep Neural Network
RWP	Random Walk Pattern
SA	Sentiment Analysis
ML	Machine Learning
SVR	Support Vector Regression
LR	Linear Regression
XGBoost	eXtreme Gradient Boost
Vader	Valence Aware Dictionary and sEntiment Reasoner
NLP	Natural language processing
DJIA	Dow Jones Industrial Average
AF	Activation Functions
Tanh	Hyperbolic Tangent Function

ΠΕΡΙΛΗΨΗ

Η πρόβλεψη της τιμής μια μετοχής είναι αρκετά περίπλοκη, ωστόσο αρκετοί ερευνητές θεωρούν πως εκτός του ρίσκου της μετοχής, των ειδήσεων και της προσφοράς/ζήτησης, σημαντικό ρόλο παίζει και το συναίσθημα του κοινού. Ένας τρόπος εξαγωγής αυτού του συναισθήματος είναι από τα μέσα κοινωνικής δικτύωσης όπως Twitter, Facebook, Reddit κλπ όπου χρήστες εκφέρουν την γνώμη τους με μικρά μηνύματα.

Στην παρούσα διπλωματική εργασία, θα δημιουργήσουμε και θα εκπαιδεύσουμε μοντέλα Μηχανικής Μάθησης (Linear Regression, XGBoost, LSTM, SVR) για την πρόβλεψη των τιμών της μετοχής της Apple, Amazon, Google, Microsoft και Tesla μέσω Sentiment Analysis. Το σύνολο των δεδομένων εξήχθη από το Twitter και τα ιστορικά δεδομένα των μετοχών μέσω του Yahoo Finance. Η ανάλυση συναισθήματος πραγματοποιείται με το λεξικό Valence Aware Dictionary and sEntiment Reasoner (VADER).

Predicting the price of a stock is quite convoluted, but several researchers believe that apart from the risk of the stock, news and supply/demand, public sentiment also plays an important role. One way of extracting this sentiment is through social media sites such as Twitter, Facebook, Reddit etc. where users express their opinion with short messages.

In this thesis, we will create and train Machine Learning models (Linear Regression, XGBoost, LSTM, SVR) to predict the stock prices of Apple, Amazon, Google, Microsoft and Tesla through Sentiment Analysis. The dataset was extracted from Twitter and historical stock data via Yahoo Finance library. For sentiment analysis we used the Valence Aware Dictionary and sEntiment Reasoner (VADER).

Κεφάλαιο 1: Εισαγωγή

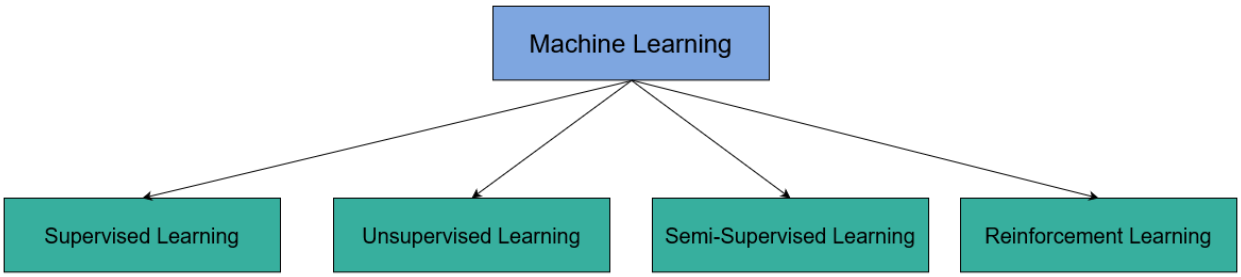
Η πρόβλεψη τιμών του Χρηματιστηρίου ερευνάται εκτενώς για πολλά χρόνια. Σύμφωνα με την Efficient Market Hypothesis (EMH) οι τιμές αυτές επηρεάζονται από νέες πληροφορίες και ακολουθούν γενικά ένα Μοτίβο Τυχαίου Περιπάτου. Έρευνες έδειξαν πως το συναίσθημα (Sentiment) και η ψυχολογία μπορούν να είναι επιπλέον παράγοντες στην τιμή των μετοχών [1] [2].

Στα πλαίσια αυτής της διπλωματικής, θα ερευνήσουμε την υπόθεση πως τα συναισθήματα και γενικά η ψυχολογία επηρεάζει την διαδικασία λήψης αποφάσεων, επομένως υπάρχει άμεση συσχέτιση μεταξύ της «κοινής γνώμης (public sentiment)» και του «market sentiment». Χρησιμοποιούμε την πλατφόρμα του Twitter για τα δεδομένα, στην οποία οποιοσδήποτε μπορεί να εκφράσει τις σκέψεις και την γνώμη του με μικρά μηνύματα. Έπειτα ακολουθεί η ανάλυση συναισθημάτων (sentiment analysis) όπου προσπαθούμε να εξάγουμε συναίσθημα είτε Θετικό ή Αρνητικό για την συγκεκριμένη μετοχή-εταιρεία. Τέλος, χρησιμοποιούμε αυτό το συναίσθημα και την τιμή της μετοχής την προηγούμενη μέρα για να προβλέψουμε την κίνηση της μετοχής (ανοδικά - καθοδικά) μέσω μοντέλων Μηχανικής Μάθησης (ML).

1.1. Μηχανική Μάθηση (Machine Learning)

Η μηχανική μάθηση ορίζεται ως η μελέτη των προγραμμάτων υπολογιστών που αξιοποιούν αλγορίθμους και στατιστικά μοντέλα για να μαθαίνουν μέσω συμπερασμάτων και μοτίβων χωρίς να έχουν σχεδιαστεί ειδικά για την αντιμετώπιση του εν λόγω προβλήματος.

1.2. Κατηγορίες Μηχανικής Μάθησης



Εικόνα 1. Κατηγορίες Μηχανικής Μάθησης

1.2.1. Επιβλεπόμενη Μάθηση (Supervised Learning)

Στην επιβλεπόμενη μάθηση ένας αλγόριθμος εκπαιδεύεται σε δεδομένα εισόδου με ετικέτα (labeled data) για μια συγκεκριμένη έξοδο. Το μοντέλο εκπαιδεύεται ανιχνεύοντας μοτίβα στα δεδομένα που του επιτρέπουν να εξάγει συμπεράσματα από δεδομένα που δεν έχει δει ξαναδεί.

1.2.2. Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning)

Στην Μη Επιβλεπόμενη Μάθηση δεν απαιτείται επίβλεψη του μοντέλου από τον χρήστη. Ασχολείται κυρίως σε δεδομένα χωρίς ετικέτες (unlabeled data) και αφήνει το μοντέλο να εκπαιδευτεί μόνο του για να ανακαλύψει πληροφορίες και μοτίβα.

1.2.3. Ημι-Επιβλεπόμενη Μάθηση (Semi-Supervised Learning)

Στην Ημι-Επιβλεπόμενη Μάθηση, κατά τη διάρκεια της εκπαίδευσης, αναμειγνύονται μεγάλες ποσότητες δεδομένων χωρίς ετικέτες με μια μικρή ποσότητα δεδομένων με ετικέτες. Είναι συνδυασμός της επιβλεπόμενης με την μη επιβλεπόμενη μάθηση και χρησιμοποιείται όταν τα δεδομένα είναι δαπανηρό ή δύσκολο να επισημανθούν.

1.2.4. Ενισχυτική Μάθηση (Reinforcement Learning)

Η ενισχυτική μάθηση επιτρέπει σε έναν πράκτορα να μαθαίνει σε ένα διαδραστικό περιβάλλον χρησιμοποιώντας τις δικές του εμπειρίες και ενέργειες ως ανατροφοδότηση, μέσω δοκιμών (trial and error). Χρησιμοποιείται κυρίως στην ρομποτική και η εφαρμογή τους είναι περιορισμένη λόγω της ανάγκης μεγάλου όγκου δεδομένων.

Κεφάλαιο 2: Σκοπός

2.1. Περιγραφή του προβλήματος

Οι προβλέψεις γίνονται δυσκολότερες και λιγότερο ακριβείς όταν ο χρονικός ορίζοντας γίνεται μεγαλύτερος. Για παράδειγμα, η πρόγνωση του καιρού είναι πιο ακριβής μερικές ώρες στο μέλλον, λιγότερο ακριβής για την επόμενη ημέρα και συνήθως δεν βλέπουμε προβλέψεις για τον επόμενο μήνα. Όπως και στο πρόβλημα της πρόβλεψης του καιρού, είναι δύσκολο να προβλεφθούν οι κινήσεις των χρηματιστηρίων και ακόμη πιο δύσκολο να προβλεφθούν οι τιμές των μετοχών σε μεγάλους χρονικούς ορίζοντες (πχ 1 μήνα μετά).

Για την επίλυση αυτού του προβλήματος ερευνητές και επενδυτές προσπαθούν να κατανοήσουν πότε και γιατί οι τιμές των μετοχών αλλάζουν. Τα τελευταία χρόνια έχει παρατηρηθεί πως τα μέσα κοινωνικής δικτύωσης (social media) παίζουν ολοένα και μεγαλύτερο ρόλο στην τιμή μιας μετοχής, και θεωρούνται εξωγενής μεταβλητές. Για παράδειγμα ο Διευθύνων Σύμβουλος της Tesla, Elon Musk, είχε γράψει στο Twitter πως σκεφτόταν να ιδιωτικοποιήσει την εταιρεία του όταν η τιμή της φτάσει στα 420$ ανά μετοχή, με αποτέλεσμα την ίδια μέρα η μετοχή της Tesla αυξήθηκε κατά 10%. Έτσι, δημιουργήθηκαν καινούριοι τομείς έρευνας όπως ανάλυση κοινωνικών δικτύων (social network analysis) και ανάλυση συναισθήματος (sentiment analysis) με την οποία θα ασχοληθούμε.

2.2. Σκοπός

Ο σκοπός της διπλωματικής αυτής, είναι η απόκτηση και επεξεργασία των κατάλληλων δεδομένων και η δημιουργία μοντέλων Μηχανικής Μάθησης και Νευρωνικών Δικτύων για την πρόβλεψη της τιμής μιας μετοχής την επόμενη μέρα μέσω ανάλυσης συναισθήματος αλλά και την απεικόνιση αυτών των δεδομένων.

2.3. Εξέλιξη Τεχνολογίας

Οι τιμές των μετοχών αυξομειώνονται κυρίως λόγω της προσφοράς και της ζήτησης των αγορών, σύμφωνα με τη θεωρία της μικροοικονομίας. Μια ποικιλία παραγόντων εμπλέκεται στην πρόβλεψη της αξίας μιας μετοχής, όπως οι γενικές οικονομικές συνθήκες, η πολιτική σταθερότητα, οι αξιολογήσεις των πελατών μιας εταιρείας, οι προσδοκίες των εμπόρων και τα μέσα κοινωνικής δικτύωσης.

Αν και η ανάλυση συναισθήματος κειμένων είναι ένας ώριμος τομέας, η ανάλυση συναισθήματος σε κείμενα web ξεκίνησε πριν από περίπου μια δεκαετία. Ειδικά τα τελευταία χρόνια παρατηρείται σημαντική αύξηση της ανάλυσης συναισθήματος στα μέσα κοινωνικής δικτύωσης, κυρίως το Twitter και το Facebook.

Αρκετοί ερευνητές έχουν χρησιμοποιήσει ανάλυση συναισθήματος [2], άλλοι τεχνικές Μηχανικής Μάθησης [3] και άλλοι έναν συνδυασμό αυτών [4], για την πρόβλεψη της τιμής μιας μετοχής.

Το 2010, ο Asur κ.α. [5] επικεντρώθηκαν στην πρόβλεψη της επιτυχίας των box office ταινιών μέσω της ανάλυσης συναισθήματος των Tweets. Η βασική υπόθεση ήταν πως όσο περισσότερο συζητείται μια ταινία, τόσο πιο πιθανό είναι να επιτύχει στο box office. Έπειτα, χρησιμοποίησαν δύο προσεγγίσεις για την πρόβλεψη:

- Χωρίς ανάλυση συναισθήματος, όπου μία αναφορά (mention) θεωρείται θετική με βασική υπόθεση ότι τα θετικά σχόλια για ταινίες είναι περισσότερα από τα αρνητικά. Έπειτα μέσω ενός μοντέλου παλινδρόμησης πραγματοποιούν την πρόβλεψη.
- Με ανάλυση συναισθήματος, μέσω του γλωσσικού μοντέλου N-gram όπου χρησιμοποίησαν το Amazon Mechanical Turk για τα δεδομένα και σύγκριναν την πρόβλεψη με το χρηματιστήριο του Χόλυγουντ (HSX).

Τα αποτελέσματα έδειξαν ότι τα δεδομένα του twitter μπορούν να χρησιμοποιηθούν για την πρόβλεψη της επιτυχίας του box office μιας ταινίας.

Ο Mao κ.α [6], μελέτησαν την συσχέτιση μεταξύ των δεδομένων του Twitter και της απόδοσης των μετοχών, όπου απέκτησαν στατιστικά σημαντικό προβάδισμα ενσωματώνοντας volume spikes του Twitter σε ένα Μπαγεσιανό Ταξινομητή (Bayesian Classifier).

Μια από τις πιο σημαντικές έρευνες είναι αυτή του Bollen κ.α. [2], όπου χρησιμοποιούν ανάλυση συναισθήματος στο twitter για να προβλέψουν το χρηματιστήριο. Η ανάλυση συναισθήματος βασίζεται στο OpinionFinder και το POMS (Profile of Mood States), το οποίο αποδίδει μια θετική ή αρνητική πολικότητα σε ένα tweet, ενώ το POMS αποδίδει μία από τις ακόλουθες έξι ετικέτες: ήρεμος, σε εγρήγορση, σίγουρος, ζωτικός, ευγενικός και χαρούμενος. Μια χρονοσειρά συναισθήματος κατασκευάζεται χρησιμοποιώντας το συλλογικό συναίσθημα των tweet ανά ημέρα. Η ανάλυσή τους δείχνει ισχυρή συσχέτιση μεταξύ της "ήρεμης" διάθεσης και των δεδομένων του Dow Jones Industrial Average (DJIA).

Τέλος, ο Ruiz κ.α. [7] χρησιμοποιούν tweets για συγκεκριμένες μετοχές και αναπαριστούν τα tweets μέσω γραφημάτων που αποτυπώνουν διάφορες πτυχές της συζήτησης για τις εν λόγω μετοχές. Στη συνέχεια, ορίζονται δύο ομάδες χαρακτηριστικών με βάση αυτούς τους γράφους:

χαρακτηριστικά με βάση τη δραστηριότητα και χαρακτηριστικά με βάση το γράφημα. Μελετώντας τις σχέσεις μεταξύ αυτών των χαρακτηριστικών και του όγκου συναλλαγών και της τιμής των μετοχών, αναπτύσσουν μια στρατηγική trading που αποδίδει καλύτερα σε σχέση με άλλες βασικές στρατηγικές.

2.4. Δεδομένα

Τα δεδομένα που χρησιμοποιήθηκαν συλλέχθηκαν από το Kaggle [8], καθώς το δωρεάν API του Twitter παρέχει πολύ περιορισμένες δυνατότητες. Το dataset διαθέτει περίπου 3 εκατομμύρια labeled tweets για Amazon, Apple, Google, Microsoft και Tesla την χρονική περίοδο 2015-2020. Για την ιστορικότητα των τιμών των μετοχών χρησιμοποιήθηκε η βιβλιοθήκη yahoo finance.

2.4.1. Καθαρισμός Δεδομένων

Η ανάλυση συναισθήματος απαιτεί τα δεδομένα που δίνουμε να είναι όσο το δυνατόν πιο «καθαρά» δηλαδή να μην περιέχουν άχρηστες πληροφορίες ή πληροφορίες που θα επηρεάσουν το αποτέλεσμα. Συνεπώς, από το dataset αφαιρέθηκαν όλα τα:

- @Mentions
- # (Hashtags)
- Retweets (RT)
- Links
- Κενά tweets ή tweets που είχαν μόνο αριθμούς

Τέλος το σύνολο των δεδομένων χωρίστηκε σε training set (90% αναλογία) και testing set (10%), με το training set να χωρίζεται επιπλέον σε 75% train και 25% validation.

Κεφάλαιο 3: Αλγόριθμοι

3.1. Ανάλυση Συναισθήματος (Sentiment Analysis)

Η ανάλυση συναισθήματος είναι γνωστή ως η διαδικασία εξαγωγής συμπερασμάτων και ερμηνείας πληροφοριών σε ένα κείμενο, όπως οι υποκειμενικές πληροφορίες. Με αυτόν τον τρόπο, τα εξαγόμενα δεδομένα μπορούν να χρησιμοποιηθούν σε μοντέλα μηχανικής μάθησης ή σε στατιστικά μοντέλα. Κατά την εφαρμογή της ανάλυσης συναισθήματος στα μέσα κοινωνικής δικτύωσης, το ίδιο το κείμενο ήταν το κύριο σημείο εστίασης. Οι Sul κ.ά. [9] μελέτησαν τον αριθμό των followers χρηστών του Twitter που έκαναν tweet για ορισμένες μετοχές. Διαπιστώθηκε ότι οι χρήστες με λιγότερους από 171 followers που έκαναν tweet για μια εταιρεία, παρουσίασαν μεγαλύτερη επίδραση στις αποδόσεις της μετοχής την επόμενη ημέρα, από ό,τι οι λογαριασμοί με περισσότερους followers.

Η ανάλυση συναισθήματος πραγματοποιείται μέσω μιας συστηματικής προσέγγισης από αλγορίθμους, για την εξαγωγή π.χ. της πολικότητας, των θεμάτων και των απόψεων από το κείμενο. Η μοντελοποίηση της γλώσσας βάσει κανόνων ή η εξαγωγή κρυμμένων μοτίβων μέσω τεχνητής νοημοσύνης είναι δύο μέθοδοι που μπορούν να εφαρμοστούν. Η ανάλυση συναισθήματος θεωρείται περίπλοκη, δεδομένου ότι η σύνταξη και η δομή της γλώσσας δεν συνοψίζονται και δεν αναπαρίστανται εύκολα με υπολογιστικά μοντέλα. Ένα από τα προβλήματα είναι η ασάφεια των λέξεων και η ανίχνευση του σαρκασμού. Οι λέξεις έχουν διαφορετικές σημασίες ανάλογα με τα συμφραζόμενα και τη χρήση λογοτεχνικών τεχνικών όπως η ειρωνεία. Επιπλέον, το κείμενο στα μέσα κοινωνικής δικτύωσης είναι σύντομο και περιλαμβάνει emoticons, συντομεύσεις και κεφαλαία γράμματα για να τονιστεί το νόημα και τα συναισθήματα. Τέλος, η [3] έδειξε ότι τα κείμενα στα μέσα κοινωνικής δικτύωσης μπορούν να εκφράσουν συναισθήματα με μια διαφορετική δομή από το συνηθισμένο κείμενο, γεγονός που περιπλέκει την ανάλυση

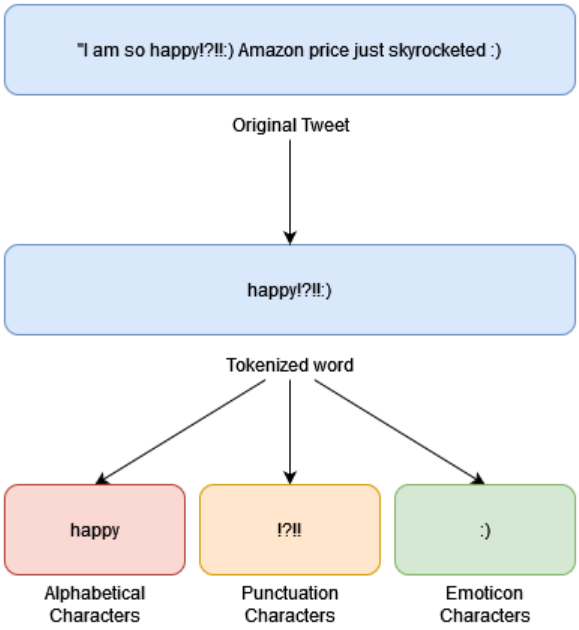
συναισθήματος.

3.2. VADER

Η VADER (Valence Aware Dictionary and sEntiment Reasoner) είναι μια βιβλιοθήκη ανάλυσης συναισθήματος βασισμένη σε λεξικό και κανόνες, γραμμένη σε Python και αποτελείται από ένα λεξικό με ένα μεγάλο σύνολο λέξεων και emoticons τα οποία επισημαίνονται σύμφωνα με την σημασία τους με βάρη. Το άθροισμα των βαρών όλων των λέξεων σε ένα κείμενο, είναι η πολικότητα που προκύπτει. Η προκύπτουσα πολικότητα εμπίπτει σε τρεις διαφορετικές κατηγορίες: Θετική, αρνητική ή ουδέτερη. Οι κατηγορίες διαχωρίζονται με δεκαδικά διαστήματα, γεγονός που καθιστά δυνατή την αξιολόγηση του πόσο ισχυρή είναι η πολικότητα ανά κατηγορία. Το VADER μπορεί να επεξεργαστεί όχι μόνο κείμενα αλλά και emojis και μπορεί επίσης να ανιχνεύσει τον σαρκασμό με μεγάλη ακρίβεια [10].

3.2.1. Tokenization

Πρόκειται για μια τεχνική διάσπασης ενός κειμένου σε μικρότερα κομμάτια, τα λεγόμενα tokens, τα οποία χωρίζονται σε λέξεις, χαρακτήρες, ή υπολέξεις. Έτσι, η διαδικασία της διάσπασης (tokenization) ταξινομείται σε γενικές γραμμές σε 3 τύπους - tokenization λέξεων, χαρακτήρων και υπολέξεων (n-gram characters).



Εικόνα 2. Tokenization λέξης και ταξινόμηση χαρακτήρων (VADER)

3.2.2. Stop Words

Κάθε γλώσσα διαθέτει λέξεις που δεν δίνουν ιδιαίτερη σημασία όταν χρησιμοποιούνται σε μια πρόταση. Αυτός οι λέξεις στην ανάλυση συναισθήματος ονομάζονται Stop Words και μπορούν με ασφάλεια να αγνοηθούν χωρίς να θυσιαστεί το νόημα της πρότασης. Για παράδειγμα, στο πλαίσιο μιας μηχανής αναζήτησης, αν το ερώτημα αναζήτησης είναι "πώς να δημιουργήσω μια εφαρμογή android", και η μηχανή αναζήτησης προσπαθήσει να βρει ιστοσελίδες που περιέχουν τους όρους "πώς", "να", "δημιουργήσω", "εφαρμογή", "android", η μηχανή αναζήτησης θα βρει πολύ περισσότερες σελίδες που περιέχουν τους όρους "πώς", "να" από ό,τι σελίδες που περιέχουν πληροφορίες σχετικά με την ανάπτυξη εφαρμογής android, επειδή οι όροι "πώς" και "να" χρησιμοποιούνται πολύ συχνά στην ελληνική γλώσσα. Αν αγνοήσουμε αυτούς τους δύο όρους, η μηχανή αναζήτησης μπορεί να επικεντρωθεί στην ανάκτηση σελίδων που περιέχουν τις λέξεις-κλειδιά: "δημιουργήσω" "εφαρμογή" "android" – με αποτέλεσμα να φέρνει σελίδες που

πραγματικά μας ενδιαφέρουν.

3.3. Γραμμική Παλινδρόμηση (Linear Regression)

Η γραμμική παλινδρόμηση χρησιμοποιείται για τη μελέτη της γραμμικής σχέσης μεταξύ μιας εξαρτημένης μεταβλητής Y (π.χ. αρτηριακή πίεση) και μιας ή περισσότερων ανεξάρτητων μεταβλητών X (ηλικία, βάρος, φύλο). Η εξαρτημένη μεταβλητή Y πρέπει να είναι συνεχής, ενώ οι ανεξάρτητες μεταβλητές μπορεί να είναι συνεχείς (ηλικία), δυαδικές (φύλο), ή κατηγορικές (κοινωνικό status). Η ανάλυση μεταξύ δύο συνεχών μεταβλητών συνήθως γίνεται με βάση ενός διαγράμματος διασποράς. Αυτός ο τύπος διαγράμματος θα δείξει αν η σχέση είναι γραμμική ή μη γραμμική. Η εκτέλεση γραμμικής παλινδρόμησης έχει νόημα μόνο εάν η σχέση είναι γραμμική. Στα πλαίσια αυτής της εργασίας, χρησιμοποιήθηκε Γραμμική Παλινδρόμηση Ελαχίστων Τετραγώνων που είναι και η πιο απλή υλοποίηση.

Αν X η ανεξάρτητη και Y η εξαρτημένη μεταβλητή έχουμε:

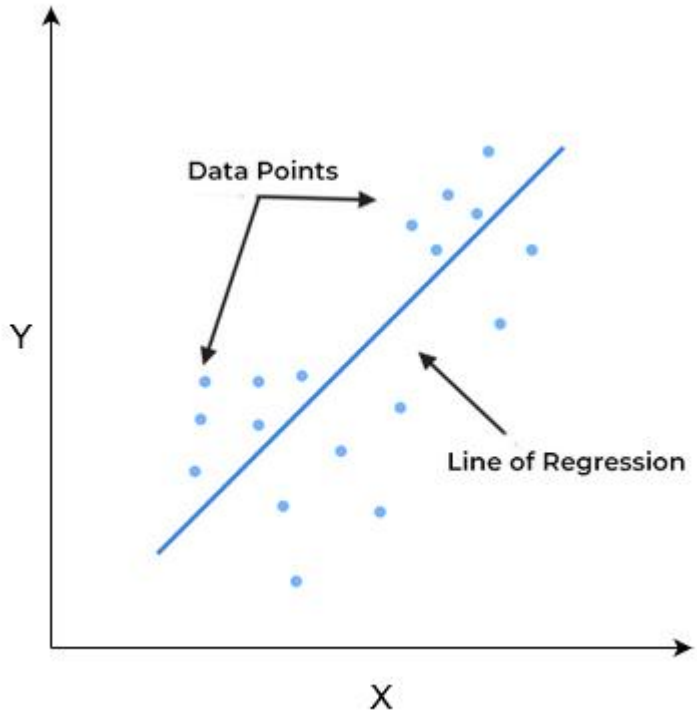
$$Y = mX + c$$

Όπου m είναι η κλίση της ευθείας και c είναι η τομή y. Έπειτα η μέθοδος Ελαχίστων Τετραγώνων:

$$m = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$c = \bar{y} - m\bar{x}$$

Με $\bar{x}$ είναι ο μέσος όρος όλων των τιμών στην είσοδο X και $\bar{y}$ είναι ο μέσος όρος όλων των τιμών στην επιθυμητή έξοδο Y.



Εικόνα 3. Γραμμική Παλινδρόμηση Πηγή: Toolbox

3.4. Γραμμική Παλινδρόμηση Διανυσμάτων Υποστήριξης (LinearSVR)

Πρόκειται για έναν αλγόριθμο επιβλεπόμενης μάθησης ο οποίος εφαρμόζεται στην πρόβλεψη διακριτών τιμών. Η παλινδρόμηση διανύσματος υποστήριξης χρησιμοποιεί την ίδια αρχή με τα SVM. Η εύρεση της γραμμής με τη βέλτιστη προσαρμογή (fit) είναι η θεμελιώδης αρχή της SVR. Εκεί, η γραμμή καλύτερης προσαρμογής είναι το υπερεπίπεδο που περίεχει τα περισσότερα σημεία. Σε αντίθεση με άλλα μοντέλα παλινδρόμησης που προσπαθούν να ελαχιστοποιήσουν το σφάλμα ανάμεσα σε πραγματική και προβλεπόμενη τιμή, η SVR προσπαθεί να προσαρμόσει την καλύτερη γραμμή εντός μιας τιμής κατωφλίου. Η τιμή κατωφλίου είναι η απόσταση μεταξύ του υπερεπιπέδου και της οριακής γραμμής. Επιπλέον ο SVR χρησιμοποιεί διάφορους υπερπαράμετρους όπως:

1. Υπερεπίπεδο (Hyperplane)

   Τα υπερεπίπεδα είναι όρια απόφασης που χρησιμοποιούνται για την πρόβλεψη της συνεχούς εξόδου. Επιπλέον, τα διανύσματα υποστήριξης είναι τα σημεία δεδομένων που βρίσκονται πλησιέστερα στον άξονα του υπερεπιπέδου και χρησιμοποιούνται για τη χάραξη της απαιτούμενης γραμμής που δείχνει την πρόβλεψη εξόδου του αλγορίθμου.

2. Πυρήνας (Kernel)

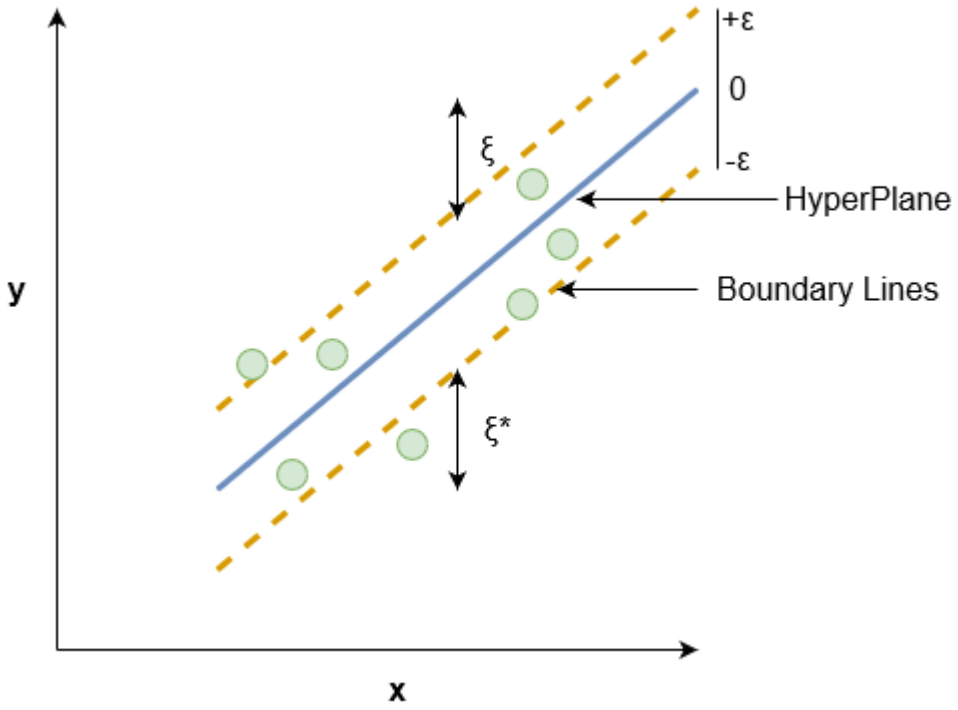
   Ένας πυρήνας είναι μια συλλογή μαθηματικών πράξεων που δέχεται δεδομένα ως είσοδο και τα τροποποιεί στην επιθυμητή μορφή. Συνήθως, χρησιμοποιούνται για τον εντοπισμό ενός υπερεπιπέδου σε χώρο μεγαλύτερων διαστάσεων με βασικότερους τον γραμμικό, τον μη γραμμικό, τον πολυωνυμικό, την ακτινική συνάρτηση βάσης (RBF) και τον σιγμοειδές. Από προεπιλογή, ως πυρήνας χρησιμοποιείται ο RBF. Στην υλοποίηση αυτής της πτυχιακής χρησιμοποιήθηκε ο γραμμικός LinearSVR.

3. Οριακές γραμμές (Boundary Lines)

   Αυτές είναι οι δύο γραμμές που σχεδιάζονται γύρω από το υπερεπίπεδο σε απόσταση ε. Χρησιμοποιείται για τη δημιουργία ενός περιθωρίου μεταξύ των σημείων δεδομένων.

$$y = \sum_{i=1}^{N} (\alpha_i - \alpha^*{}_i) * (x_i, x) + b$$

   Όπου (X_i, X) είναι μια δειγματοληψία των παραμέτρων σχεδιασμού εισόδου, $\alpha_i \geq 0$ και $\alpha^*{}_i \geq 0$ εισάγονται ως πολλαπλασιαστές Lagrange. [11]



Εικόνα 4. Παλινδρόμηση Διανυσμάτων Υποστήριξης

3.5. eXtreme Gradient Boosting

Είναι μια βιβλιοθήκη μηχανικής μάθησης που παρέχει μια αποδοτική και αποτελεσματική εφαρμογή του αλγορίθμου gradient boosting ο οποίος είναι γνωστός ως ένας από τους καλύτερους, σε επίδοση, αλγορίθμους που χρησιμοποιούνται για την επιβλεπόμενη μάθηση και χρησιμοποιείται για προβλήματα παλινδρόμησης και ταξινόμησης [12]. Αποτελείται από δέντρα αποφάσεων (decision trees) και βελτιώνεται μέσω μιας κυρτής συνάρτησης απώλειας (convex loss function). Για την πρόβλεψη της εξόδου υπολογίζεται

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)\,,\ f_k \in F$$

Όπου:

$$F = \{f(x) = w_{q(x)}\}(q : \mathbb{R}^m \to T, w \in \mathbb{R}^T)$$

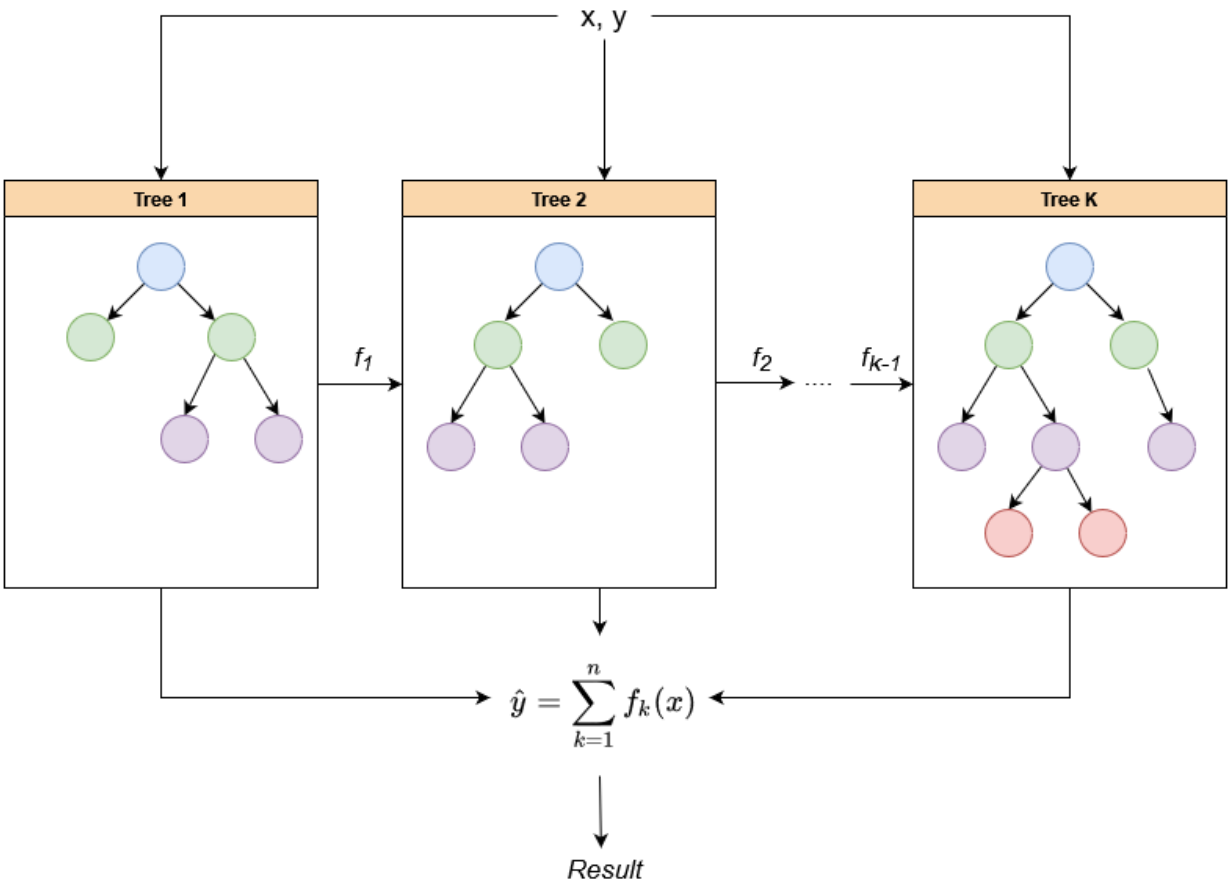
Είναι ο χώρος των δέντρων παλινδρόμησης (CART), q είναι η μορφή του δέντρου για κάθε περίπτωση του σχετικού δείκτη φύλλου. T είναι ο αριθμός των φύλλων του δέντρου, κάθε f_k αντιστοιχεί σε μια ανεξάρτητη δομή δέντρου q και βάρη φύλλων w. Για να μάθουμε το σύνολο των συναρτήσεων που χρησιμοποιούνται στο μοντέλο, ελαχιστοποιούμε τον ακόλουθο στόχο:

$$\mathrm{L} = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$

Όπου:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2$$

με l μια διαφορίσιμη κυρτή συνάρτηση απώλειας που τιμωρεί την πολυπλοκότητα του μοντέλου και αξιολογεί τη διαφορά μεταξύ της πρόβλεψης $\hat{y}_i$ και του στόχου y_i.



Εικόνα 5. XGBoost Αρχιτεκτονική

3.6. Μακράς Βραχυπρόθεσμης Μνήμης (LTSM)

Πρόκειται για ένα ανατροφοδοτούμενο νευρωνικό δίκτυο το οποίο προτάθηκε αρχικά από τους Hochreiter και Schmidhuber (1997) [13] , υλοποιήθηκε για την μοντελοποίηση και την προσέγγιση χρονικών ακολουθιών και έχει χρησιμοποιηθεί επιτυχώς για την πρόβλεψη χρονοσειρών [14] [15].

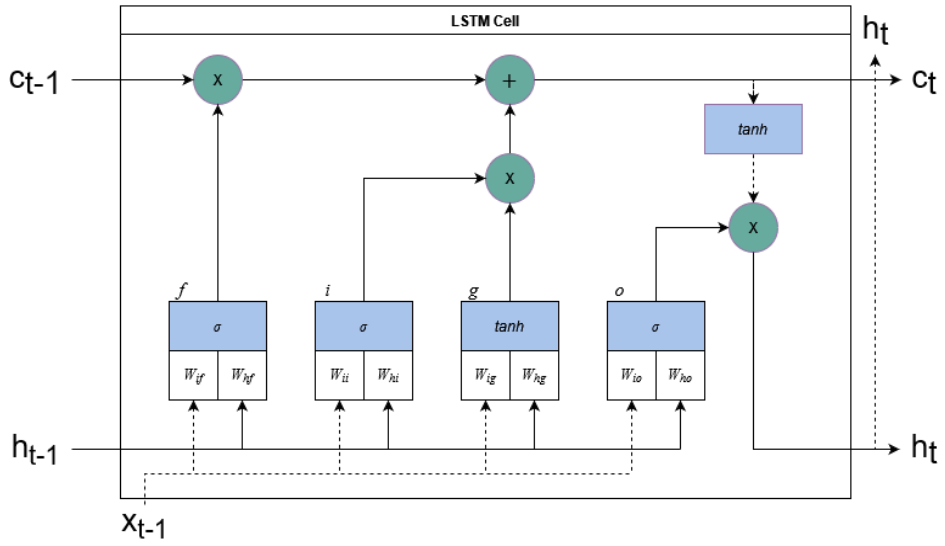
Επιπλέον, στα ανατροφοδοτούμενα νευρωνικά δίκτυα παρατηρείται το πρόβλημα της εξαφανιζόμενης κλίσης (vanishing gradient problem), το οποίο καλούνται να αντιμετωπίσουν τα δίκτυα βραχυπρόθεσμης μνήμης, με αυτά να διαφέρουν από τα απλά νευρωνικά δίκτυα λόγω των συνδέσεων ανατροφοδότησης που διαθέτουν. Μέσω αυτής της ιδιότητας μπορούν και χειρίζονται χρονοσειρές χωρίς να αντιμετωπίζουν κάθε σημείο της ακολουθίας ξεχωριστά, αντιθέτως

διατηρώντας πληροφορίες που είναι χρήσιμες σχετικά με τα δεδομένα της ακολουθίας που προηγήθηκαν για να εξυπηρετήσουν στην ανάλυση των νέων σημείων δεδομένων (data points).

Ένα δίκτυο LSTM υπολογίζει μια απεικόνιση (mapping) από μια ακολουθία εισόδου

$x = (x_1, \dots, x_t)$ σε μια ακολουθία εξόδου $y = (y, \dots, y_t)$ υπολογίζοντας τις ενεργοποιήσεις των μονάδων του δικτύου χρησιμοποιώντας τις ακόλουθες εξισώσεις επαναληπτικά από t = 1 έως T :

$$i_t = \sigma(W_{ii}\, x_t + b_{ii} + W_{hi}\, h_{t-1} + b_{hi})$$
$$f_t = \sigma(W_{if} x_t + b_{if} + W_{hf} h_{t-1} + b_{hf})$$
$$g_t = tanh(W_{ig} x_t + b_{ig} + W_{hg} h_{t-1} + b_{hg})$$
$$o_t = \sigma(W_{io} x_t + b_{io} + W_{ho} h_{t-1} + b_{ho})$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$
$$h_t = o_t \odot tanh(c_t)$$



Εικόνα 6. LSTM Αρχιτεκτονική

Όπου συνοπτικά:

- W οι πίνακες βαρών,

- h_t η κρυφή κατάσταση την στιγμή t,
- b τα διανύσματα bias (b_i είναι το bias διάνυσμα της πύλης εισόδου),
- σ η σιγμοειδής συνάρτηση,
- i η πύλη εισόδου,
- f η πύλη επιλεκτικής συγκράτησης,
- O η πύλη εξόδου,
- g η πύλη στοιχείου,
- $\circ$ το γινόμενο Hadamard (πολλαπλασιασμός πινάκων)

3.6.1. Συναρτήσεις Ενεργοποίησης

Στα νευρωνικά δίκτυα, οι συναρτήσεις ενεργοποίησης χρησιμοποιούνται για να καθορίσουν αν ένας νευρώνας μπορεί να πυροδοτηθεί ή όχι, υπολογίζοντας το σταθμισμένο άθροισμα των εισόδων και των προκαταλήψεων. Στη συνέχεια, δημιουργείται μια έξοδος για το νευρωνικό δίκτυο, η οποία περιλαμβάνει τις παραμέτρους στην είσοδο, μετά από χειρισμό των δεδομένων μέσω κάποιας επεξεργασίας κλίσης όπως gradient descent. Αυτές οι συναρτήσεις αναφέρονται συχνά ως συνάρτηση μεταφοράς σε διάφορες βιβλιογραφίες.

Χωρίζεται σε γραμμική και μη γραμμική, ανάλογα με τη συνάρτηση που αναπαριστά, και χρησιμοποιείται για τον έλεγχο των εξόδων των νευρωνικών δικτύων σε διάφορους τομείς, από την αναγνώριση αντικειμένων και την ταξινόμηση, σε συστήματα εντοπισμού Καρκίνου, πρόβλεψη καιρού κ.α., όπου σύμφωνα με πρώιμα ερευνητικά αποτελέσματα, επικυρώνουν πως η σωστή επιλογή της συνάρτησης ενεργοποίησης βελτιώνει τα αποτελέσματα στον υπολογισμό νευρωνικών δικτύων.

Για ένα γραμμικό μοντέλο, μια γραμμική απεικόνιση μιας συνάρτησης εισόδου σε μια έξοδο, όπως πραγματοποιείται στα κρυφά στρώματα πριν από την τελική πρόβλεψη της βαθμολογίας της κλάσης για κάθε ετικέτα, δίνεται από τον μετασχηματισμό συγγένειας στις περισσότερες περιπτώσεις. Ο μετασχηματισμός των διανυσμάτων εισόδου x δίνεται από τη σχέση:

$$f(x) = w^T x + b$$

Όπου χ είναι η είσοδος, w τα βάρη και b τα biases.

Τα νευρωνικά δίκτυα παράγουν γραμμικά αποτελέσματα από τις αντιστοιχίσεις της παραπάνω εξίσωσης και προκύπτει η ανάγκη για τη συνάρτηση ενεργοποίησης, πρώτα για τη μετατροπή αυτών των γραμμικών εξόδων σε μη γραμμικά αποτελέσματα για περαιτέρω υπολογισμούς, ιδίως για την εκμάθηση μοτίβων στα δεδομένα. Η έξοδος αυτών των μοντέλων δίνεται από:

$$y = (w_1 x_1 + w_2 x_2 + .. + w_n x_n + b)$$

Αυτές οι έξοδοι κάθε στρώματος τροφοδοτούνται στο επόμενο στρώμα για πολυεπίπεδα δίκτυα όπως τα βαθιά νευρωνικά δίκτυα έως ότου επιτευχθεί η τελική έξοδος, αλλά είναι εξ ορισμού γραμμικά. Η αναμενόμενη έξοδος καθορίζει τον τύπο της συνάρτησης ενεργοποίησης που θα να αναπτυχθεί σε ένα δεδομένο δίκτυο.

Ωστόσο, δεδομένου ότι η έξοδος είναι γραμμική, οι μη γραμμικές συναρτήσεις ενεργοποίησης απαιτούνται για τη μετατροπή αυτών των γραμμικών εισόδων σε μη γραμμικές εξόδους. Αυτές οι συναρτήσεις ενεργοποίησης είναι συναρτήσεις μεταφοράς που εφαρμόζονται στις εξόδους των γραμμικών μοντέλων για την παραγωγή των μετασχηματισμένων μη γραμμικών εξόδων. Η μη γραμμική έξοδος μετά την εφαρμογή της συνάρτησης ενεργοποίησης δίνεται από τη σχέση:

$$y = \alpha(w_1 x_1 + w_2 x_2 + .. + w_n x_n + b)$$

Όπου α είναι η συνάρτηση ενεργοποίησης.

Τέλος, μια ειδική ιδιότητα των μη-γραμμικών συναρτήσεων ενεργοποίησης είναι ότι είναι διαφορίσιμες, διαφορετικά δεν μπορούν να λειτουργήσουν κατά την οπισθοδιάδοση (backpropagation) των βαθιών νευρωνικών δικτύων.

3.6.1.1. Σιγμοειδής Συνάρτηση

Υπάρχουν τρεις παραλλαγές αυτής της μη γραμμικής συνάρτησης ενεργοποίησης, η οποία χρησιμοποιείται κυρίως σε νευρωνικά δίκτυα προσοτροφοδότησης (feedforward). Είναι μια περιορισμένη διαφορίσιμη πραγματική συνάρτηση, που ορίζεται για πραγματικές τιμές εισόδου, με θετικές παραγώγους παντού και κάποιο βαθμό εξομάλυνσης.

$$f(x) = \left(\frac{1}{(1 + exp^{-x})}\right)$$

Η σιγμοειδής συνάρτηση εμφανίζεται στα στρώματα εξόδου των αρχιτεκτονικών Deep Learning και χρησιμοποιείται για την πρόβλεψη εξόδου με βάση την πιθανότητα και έχει εφαρμοστεί με επιτυχία σε προβλήματα δυαδικής ταξινόμησης, μοντελοποίηση εργασιών λογιστικής παλινδρόμησης καθώς και σε άλλους τομείς νευρωνικών δικτύων.

Ωστόσο, η σιγμοειδής AF υποφέρει από σημαντικά μειονεκτήματα, τα οποία περιλαμβάνουν απότομα damp gradients κατά την οπισθοδιάδοση από βαθύτερα κρυμμένα στρώματα προς τα στρώματα εισόδου, κορεσμό κλίσης, αργή σύγκλιση και μη μηδενική κεντροποιημένη έξοδο προκαλώντας έτσι τις ενημερώσεις των κλίσεων να διαδίδονται προς διαφορετικές κατευθύνσεις.

1) Hard Σιγμοειδής Συνάρτηση

Μια από τις παραλλαγές η οποία δίνεται από την σχέση:

$$f(x) = max\left(0, min\left(1, \frac{(x+1)}{2}\right)\right)$$

2) Sigmoid-Weighted Linear Units (SiLU)

Βασίζεται στην ενισχυτική μάθηση, προτάθηκε από τον Elfwing κ.α. [16], υπολογίζεται ως Σιγμοειδής πολλαπλασιασμένη με την είσοδό της, και δίνεται από την σχέση:

$$a_\kappa(s) = z_k a(z_k)$$

Όπου s είναι το διάνυσμα εισόδου, z_k η είσοδος στις κρυφές μονάδες k.

Η είσοδος για τα κρυφά επίπεδα δίνεται από την σχέση:

$$z_k = w_{ik}s_i + b_k$$

Με b_k το bias, w_{ik} το βάρος που συνδέεται στις κρυφές μονάδες αντίστοιχα.

Η συνάρτηση SiLU μπορεί να χρησιμοποιηθεί μόνο στα κρυφά στρώματα των βαθιών νευρωνικών δικτύων και μόνο για συστήματα βασισμένα στην ενισχυτική μάθηση.

3) Παράγωγος του Sigmoid-Weighted Linear Units (dSiLU)

Είναι η κλίση της συνάρτησης SiLU και αναφέρεται ως dSiLU. Η dSiLU χρησιμοποιείται για ενημερώσεις μάθησης με κλίση-κατάβαση (gradient-descend) για τις παραμέτρους βάρους του νευρωνικού δικτύου και η σχέση που την προσδιορίζει είναι:

$$a_k(s) = a(z_k)(1 + z_k(1 - a(z_k)))$$

3.6.1.2. Υπερβολική συνάρτηση Εφαπτομένης (Tanh)

Είναι μια ομαλότερη μηδενικού κέντρου συνάρτηση της οποίας το εύρος κυμαίνεται

μεταξύ -1 και 1, επομένως η έξοδος της συνάρτησης tanh δίνεται από τη σχέση:

$$f(x) = \left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right)$$

Η συνάρτηση tanh έγινε η προτιμώμενη συνάρτηση σε σύγκριση με τη σιγμοειδή συνάρτηση, καθώς παρέχει καλύτερες επιδόσεις εκπαίδευσης για νευρωνικά δίκτυα πολλαπλών επιπέδων. Ωστόσο, δεν μπορούσε να επιλύσει το πρόβλημα της εξαφανιζόμενης κλίσης που αντιμετώπιζαν και οι σιγμοειδείς συναρτήσεις. Το κύριο πλεονέκτημα που παρέχει η συνάρτηση είναι ότι παράγει μηδενικό κέντρο εξόδου βοηθώντας έτσι τη διαδικασία οπισθοδιάδοσης. Μια ιδιότητα της συνάρτησης tanh είναι ότι μπορεί να αποκτήσει κλίση 1 μόνο όταν η τιμή της εισόδου είναι 0, δηλαδή όταν το x είναι μηδέν. Αυτό κάνει τη συνάρτηση tanh να παράγει μερικούς νεκρούς νευρώνες κατά τη διάρκεια του υπολογισμού. Ο νεκρός νευρώνας είναι μια κατάσταση όπου το βάρος ενεργοποίησης, χρησιμοποιείται σπάνια ως αποτέλεσμα της μηδενικής κλίσης. Αυτός ο περιορισμός της συνάρτησης tanh ώθησε σε περαιτέρω έρευνα στις συναρτήσεις ενεργοποίησης για την επίλυση του προβλήματος και έτσι δημιουργήθηκε η συνάρτηση ενεργοποίησης rectified linear unit (ReLU) που θα αναλύσουμε παρακάτω.

3.6.1.3. Rectified Linear Unit (ReLU)

Προτάθηκε από τον Nair και Hinton [17], η οποία για εφαρμογές που χρησιμοποιούν Deep Learning, είναι η πιο διαδεδομένη συνάρτηση ενεργοποίησης και προσφέρει καλύτερη απόδοση σε σχέση με την Σιγμοειδή και την Υπερβολική Συνάρτηση Εφαπτομένης [18]. Προσφέρει μια γραμμική συνάρτηση διατηρώντας έτσι τις ιδιότητες των γραμμικών μοντέλων κάνοντας την εύκολη στη βελτιστοποίηση, μέσω μεθόδων Gradient-Descent. Μια συνάρτηση κατωφλίου χρησιμοποιείται σε κάθε στοιχείο εισόδου, όπου οι τιμές μικρότερες του μηδενός μηδενίζονται και δίνεται από τη σχέση:

$$f(x) = \max(0,x) = \begin{cases} x_i, \alpha\nu\ x_i \geq 0 \\ 0, \alpha\nu\ x_i < 0 \end{cases}$$

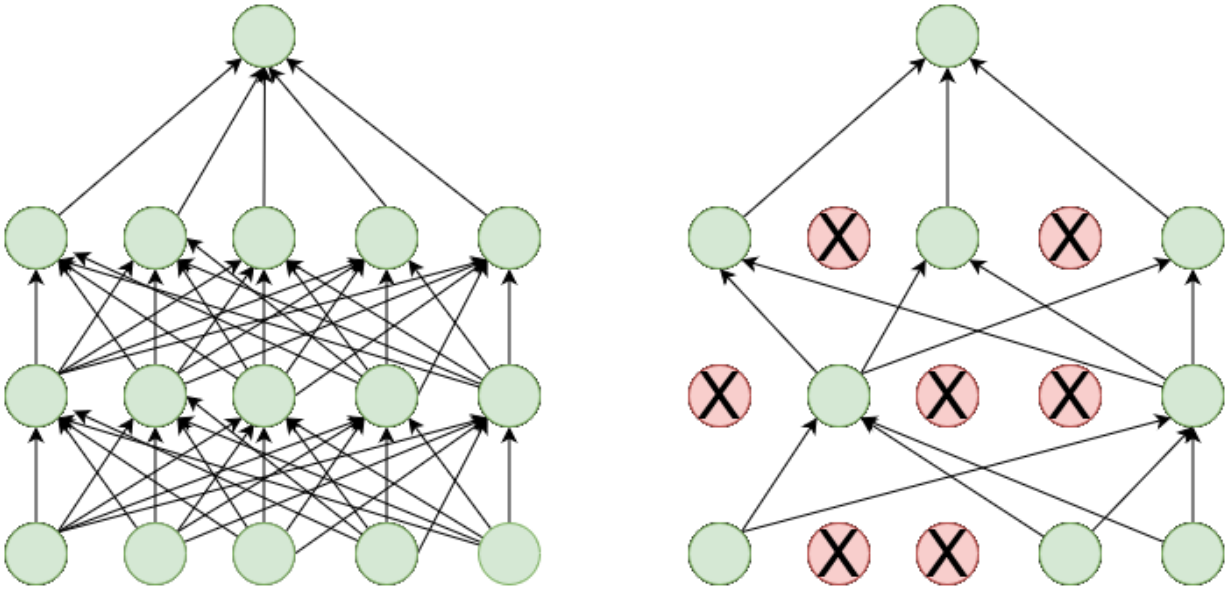
Η συνάρτηση αυτή μηδενίζει τις εισόδους με τιμή μικρότερη του μηδενός και διορθώνει έτσι το πρόβλημα της εξαφανιζόμενης κλίσης που παρατηρείται σε προηγούμενους τύπους συναρτήσεων ενεργοποίησης. Το κύριο πλεονέκτημα της ReLU είναι η ταχύτητα υπολογισμού, αφού δεν λαμβάνει υπόψιν εκθετικές συναρτήσεις και διαιρέσεις. Μια άλλη ιδιότητα της ReLU είναι ότι εισάγει αραιότητα στις κρυφές μονάδες, καθώς συμπυκνώνει το εύρος των τιμών από μηδέν έως μέγιστο. Όμως, διαθέτει το πρόβλημα της υπερπροσαρμογής (overfit), ακόμη και με την υλοποίηση της τεχνικής εγκατάλειψης (dropout), η οποία χρησιμοποιείται για να μειωθεί η επίδραση της υπερπροσαρμογής των ReLUs.

3.6.1.4. Dropout

Πρόκειται για μια τεχνική που συνδυάζει εκθετικά πολλές διαφορετικές αρχιτεκτονικές νευρωνικών δικτύων, ενώ παράλληλα αντιμετωπίζει την υπερπροσαρμογή. Η απόρριψη μονάδων σε ένα νευρωνικό δίκτυο, τόσο κρυφών όσο και ορατών, ορίζεται ως “dropout”. Μια μονάδα "απορρίπτεται" όταν οι συνδέσεις της τόσο με το εισερχόμενο όσο και με το εξερχόμενο δίκτυο διακόπτονται προσωρινά και η επιλογή αυτών που θα εγκαταλειφθούν γίνεται τυχαία. Στο πιο απλό σενάριο, κάθε μονάδα έχει πιθανότητα διατήρησης p που είναι ανεξάρτητη από όλες τις άλλες μονάδες, όπου p μπορεί να επιλεγεί χρησιμοποιώντας ένα σύνολο επικύρωσης, με βέλτιστη πιθανότητα διατήρησης 0.5 για το μεγαλύτερο μέρος δικτύων. Η καλύτερη πιθανότητα διατήρησης, ωστόσο, είναι συνήθως πιο κοντά στο 1 για τις μονάδες εισόδου.

Η εφαρμογή του dropout ισοδυναμεί με τη δειγματοληψία ενός "αραιωμένου" δικτύου από αυτό. Όλες οι μονάδες που κατάφεραν να «επιζήσουν» από την εγκατάλειψη αποτελούν το αραιωμένο δίκτυο. Μπορείτε να θεωρήσουμε ένα νευρωνικό δίκτυο με n μονάδες ως συνδυασμό

2n πιθανών αραιών νευρωνικών δικτύων, τα οποία μοιράζονται όλα τα βάρη, ώστε να εξακολουθούν να υπάρχουν συνολικά λιγότερες ή ίσες με O(n2) παράμετροι. Για κάθε παρουσίαση κάθε περίπτωσης εκπαίδευσης, γίνεται δειγματοληψία και εκπαίδευση ενός νέου αραιωμένου δικτύου. Έτσι, η εκπαίδευση ενός νευρωνικού δικτύου με dropout μπορεί να θεωρηθεί ως εκπαίδευση μιας συλλογής 2n αραιωμένων δικτύων με εκτεταμένο διαμοιρασμό βαρών, όπου κάθε αραιωμένο δίκτυο εκπαιδεύεται πολύ σπάνια, ίσως και καθόλου.



Εικόνα 7. Απλό Νευρωνικό Δίκτυο – Νευρωνικό Δίκτυο μετά από την εφαρμογή Dropout

Ας υποθέσουμε πως έχουμε ένα νευρωνικό δίκτυο με L κρυφά επίπεδα, $l \in \{1, \dots, L\}$ ο δείκτης του κρυφού επιπέδου δικτύου, $z^{(l)}$ το διάνυσμα των εισόδων στο επίπεδο l και $y^{(l)}$ το διάνυσμα των εξόδων από το επίπεδο l ($y^{(0)} = x$ *είναι η είσοδος*). $w^{(l)}$ *και* $b^{(l)}$ είναι τα βάρη και τα biases στο επίπεδο l αντίστοιχα. Η λειτουργία feed-forward ενός απλού νευρωνικού δικτύου περιγράφεται ως:

$$z_i^{(l+1)} = w_i^{(l+1)} y^l + b_i^{(l+1)}$$

$$y_i^{(l+1)} = f\left(z_i^{(l+1)}\right)$$

Όπου *f* μια οποιαδήποτε συνάρτηση ενεργοποίησης και *i* η κρυφή μονάδα.

Η παραπάνω σχέση, μέσω του dropout, μετατρέπεται σε:

$$r_j^{(l)} \sim Bernoulli(p),$$

$$\tilde{y}^{(l)} = r^{(l)} * y^{(l)},$$

$$z_i^{(l+1)} = w_i^{(l+1)} \tilde{y}^l + b_i^{(l+1)},$$

$$y_i^{(l+1)} = f\left(z_i^{(l+1)}\right).$$

Όπου * είναι το γινόμενο των στοιχείων και για κάθε επίπεδο *l*, $r^{(l)}$είναι το διάνυσμα των ανεξάρτητων Bernoulli τυχαίων μεταβλητών, οι οποίες έχουν πιθανότητα *p* να ισούνται με 1. Αυτό το διάνυσμα δειγματοληπτείται και πολλαπλασιάζεται με τις εξόδους αυτού του επιπέδου, $y^{(l)}$, για να δημιουργηθούν οι αραιωμένες έξοδοι $\tilde{y}^{(l)}$. Οι αραιωμένες έξοδοι χρησιμοποιούνται στη συνέχεια ως είσοδος στο επόμενο επίπεδο με την διαδικασία αυτή να εφαρμόζεται σε κάθε επίπεδο. Αυτό ισοδυναμεί με δειγματοληψία ενός υποδικτύου από ένα μεγαλύτερο δίκτυο. Για τη μάθηση, οι παράγωγοι της συνάρτησης απωλειών διαδίδονται μέσω του υποδικτύου. Τέλος, τα βάρη κλιμακώνονται ως $W_{test}^{(l)} = pW^{(l)}$, κατά την διάρκεια του test, και το τελικό νευρωνικό δίκτυο χρησιμοποιείται χωρίς το dropout.

3.6.2. Αλγόριθμος Βελτιστοποίησης Adam

Πρόκειται για έναν αλγόριθμο στοχαστικής βελτιστοποίησης που χρησιμοποιείται για την εκπαίδευση deep learning μοντέλων και προτάθηκε από τους Kingma και Ba [19] το 2014. Τροποποιεί τα χαρακτηριστικά του νευρωνικού δικτύου, όπως τα βάρη και τον ρυθμό μάθησης και βοηθά στη μείωση της συνολικής απώλειας και στη βελτίωση της ακρίβειας (accuracy).

Ψευδοκώδικας αλγορίθμου:

Απαιτείται: α: Μέγεθος Βήματος

Απαιτείται: $\beta_1, \beta_2 \in [0,1)$: Εκθετικά φθίνοντα ποσοστά εκτιμήσεων στιγμής

Απαιτείται: $f(\theta)$: Στοχαστική αντικειμενική συνάρτηση με παραμέτρους θ

Απαιτείται: θ_0: Αρχικό διάνυσμα παραμέτρων

$m_0 \leftarrow 0$ (Αρχικοποίηση διανύσματος 1ης στιγμής)

$v_0 \leftarrow 0$ (Αρχικοποίηση διανύσματος 2ης στιγμής)

$t \leftarrow 0$ (Αρχικοποίηση χρονικού βήματος)

Όσο θ_t δεν έχει συγκλίνει

$t \leftarrow t + 1$

$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Λήψη κλίσεων ως προς τον στοχαστικό στόχο στο χρονικό βήμα t)

$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Ενημέρωση biased εκτίμησης 1ης στιγμής)

$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot {g^2}_t$ (Ενημέρωση biased εκτίμησης 2ης ακατέργαστης στιγμής)

$\widehat{m}_t \leftarrow m_t / (1 - {\beta^t}_1)$ (Υπολογισμός διόρθωσης του bias για την εκτίμηση της 1ης στιγμής)

$\hat{v}_t \leftarrow v_t / (1 - {\beta^t}_2)$ (Υπολογισμός διόρθωσης του bias για την εκτίμηση

της $2^{ης}$ ακατέργαστης στιγμής)

$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon)$ (Ενημέρωση των παραμέτρων)

Τέλος όσο

Επέστρεψε θ_t (Παράμετροι που προέκυψαν)

Όπου $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ και $\epsilon = 10^{-8}$. Έστω $f(\theta)$ μια στοχαστική κλιμακωτή συνάρτηση που είναι διαφορίσιμη ως προς τις παραμέτρους θ. Ο στόχος είναι να ελαχιστοποιήσουμε την $E[f(\theta)]$ σε σχέση με τις παραμέτρους θ. Το $f_1(\theta), \dots, f_T(\theta)$ μετρά την επιτυχία της στοχαστικής συνάρτησης σε μελλοντικά βήματα $1, \dots, T$. Το $g_t = \nabla_\theta f_t(\theta)$ συμβολίζει την κλίση, η οποία είναι το διάνυσμα των μερικών παραγώγων της f_t σε σχέση με το θ που υπολογίζεται στο χρονικό βήμα t.

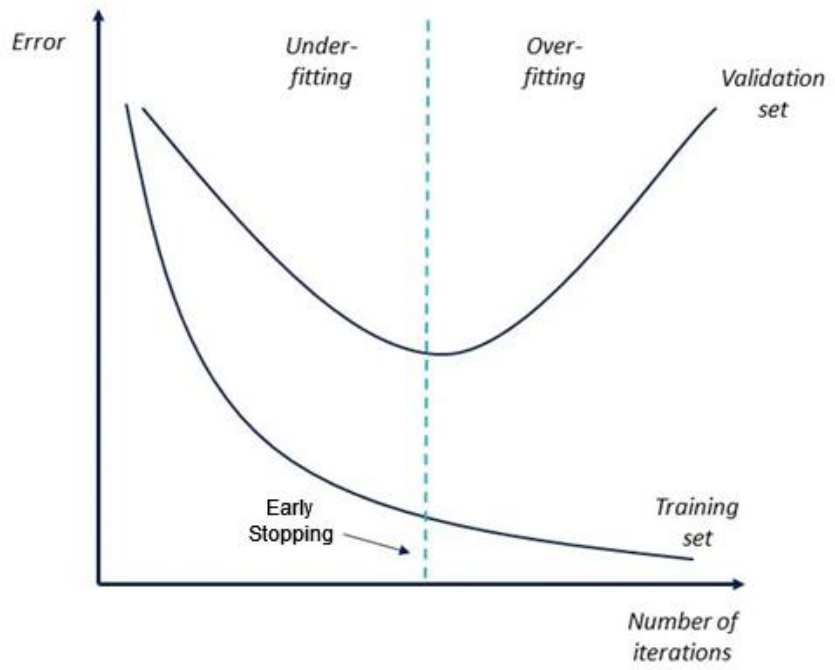
Οι εκθετικοί κινητοί μέσοι όροι της τετραγωνικής κλίσης v_t και της κλίσης m_t ενημερώνονται από τον αλγόριθμο, με τους υπερ-παραμέτρους $\beta_1, \beta_2 \in [0,1)$ να ελέγχουν τους ρυθμούς εκθετικής αποσύνθεσης των όρων που χρησιμοποιούνται στους κινητούς μέσους όρους.

Οι εκτιμήσεις της πρώτης (του μέσου όρου) και της δεύτερης ακατέργαστης στιγμής (η μη συγκεντρωτική διακύμανση) παρέχονται από αυτούς τους κινητούς μέσους όρους. Ωστόσο, αρχικοποιούνται ως διανύσματα ίσα με το 0, οδηγώντας σε εκτιμήσεις που έχουν bias προς το μηδέν, ιδίως κατά τη διάρκεια των αρχικών χρονικών βημάτων, και ιδίως όταν οι ρυθμοί αποσύνθεσης είναι μικροί.

3.6.3. Early Stopping

Είναι μια τεχνική κανονικοποίησης για βαθιά νευρωνικά δίκτυα που σταματά την εκπαίδευση όταν οι ενημερώσεις των παραμέτρων δεν αρχίζουν πλέον να βελτιώνονται σε ένα σύνολο επικύρωσης (overfitting). Αποθηκεύουμε και ενημερώνουμε τις τρέχουσες καλύτερες

παραμέτρους κατά τη διάρκεια της εκπαίδευσης, και όταν οι ενημερώσεις παραμέτρων δεν αποφέρουν πλέον βελτίωση (μετά από έναν καθορισμένο αριθμό επαναλήψεων) σταματάμε την εκπαίδευση και χρησιμοποιούμε τις τελευταίες καλύτερες παραμέτρους.



Εικόνα 8. Early Stopping. Πηγή: ibm

Η πιο απλή υλοποίηση Early Stopping είναι η εξής:

1. Διαχωρισμός των δεδομένων εκπαίδευσης σε ένα σύνολο εκπαίδευσης και ένα σύνολο επικύρωσης, συνήθως σε αναλογία 2 προς 1.
2. Εκπαίδευση μόνο στο σύνολο εκπαίδευσης και αξιολόγηση του σφάλματος ανά δείγμα στο σύνολο επικύρωσης μια στο τόσο, π.χ. μετά από κάθε πέμπτη εποχή.
3. Τερματισμός εκπαίδευσης εφόσον το σφάλμα στο σύνολο επικύρωσης είναι υψηλότερο από ό,τι ήταν την τελευταία φορά που ελέγχθηκε.

4. Χρήση τω βαρών που είχε το δίκτυο στο προηγούμενο βήμα ως αποτέλεσμα της εκπαίδευσης.

Αυτή η προσέγγιση χρησιμοποιεί το σύνολο επικύρωσης για να προβλέψει τη συμπεριφορά σε πραγματικά δεδομένα (ή σε ένα σύνολο δοκιμών), υποθέτοντας ότι το σφάλμα και στα δύο θα είναι παρόμοιο. Το σφάλμα επικύρωσης χρησιμοποιείται ως εκτίμηση του σφάλματος γενίκευσης.

3.6.4. Loss Functions

Οι συναρτήσεις απώλειας χρησιμοποιούνται για τον υπολογισμό του σφάλματος μεταξύ της εξόδου των αλγορίθμων και της καθορισμένης τιμής στόχου.. Με απλά λόγια, η συνάρτηση απώλειας εκφράζει πόσο μακριά από το στόχο είναι η υπολογισμένη μας έξοδος.

3.6.4.1. L1 Loss Function ή Least Absolute Deviations

Η συνάρτηση απώλειας L1, επίσης γνωστή ως απόλυτη απώλεια σφάλματος, είναι η απόλυτη διαφορά μεταξύ μιας πρόβλεψης και της πραγματικής τιμής, η οποία υπολογίζεται για κάθε παράδειγμα σε ένα σύνολο δεδομένων. Το άθροισμα όλων αυτών των τιμών απώλειας ονομάζεται συνάρτηση κόστους, όπου η συνάρτηση κόστους για την L1 είναι συνήθως η MAE (Mean Absolute Error).

$$L1LossFunction = \left|y_{actual} - y_{predicted}\right|$$

3.6.4.2. L2 Loss Function ή Squared Error Loss

Η συνάρτηση απώλειας L2, επίσης γνωστή ως τετραγωνική απώλεια σφάλματος, είναι η τετραγωνική διαφορά μεταξύ μιας πρόβλεψης και της πραγματικής τιμής, που υπολογίζεται για κάθε παράδειγμα σε ένα σύνολο δεδομένων. Σε αντίθεση με την L1, η συνάρτηση κόστους για την L2 είναι συνήθως η MSE (Mean of Squared Errors).

$$L2LossFunction = \left(y_{actual} - y_{predicted}\right)^2$$

3.7. Υλοποίηση LSTM – Εκπαίδευση

Το LSTM που υλοποιήθηκε χρησιμοποιεί μια Dropout, δύο Linear, μια ReLU και μια Σιγμοειδή συνάρτηση ενεργοποίησης. Τα σύνολα εκπαίδευσης θα χωριστούν ως:

1. Σύνολο εκπαίδευσης: το αρχικό 90% του συνόλου δεδομένων, όπου διαχωρίζεται επιπλέον σε 75% για εκπαίδευση και 25% για επικύρωση.
2. Σύνολο δοκιμής: το τελευταίο 10% του συνόλου δεδομένων

Χρησιμοποιήθηκε επίσης τεχνητή διόγκωση των δεδομένων μέσω της τεχνικής Ολισθαίνοντος Παραθύρου (sliding window) και ο αλγόριθμος βελτιστοποίησης που χρησιμοποιήθηκε είναι ο Adam με ρυθμό εκμάθησης 0.01.

Κεφάλαιο 4: Μετρικές – Αξιολόγηση Αλγορίθμων

Η αξιολόγηση μοντέλων είναι ζωτικής σημασίας και μας βοηθά να κατανοήσουμε την απόδοση του μοντέλου και διευκολύνει την παρουσίαση του μοντέλου σε άλλους ανθρώπους. Υπάρχουν πολλές διαφορετικές μετρικές αξιολόγησης αλλά λίγες είναι κατάλληλες για να χρησιμοποιηθούν για παλινδρόμηση (regression).

4.1. Μέσο Τετραγωνικό Σφάλμα

Το μέσο τετραγωνικό σφάλμα (MSE) μετρά πόσο κοντά είναι μια γραμμή παλινδρόμησης σε ένα σύνολο σημείων δεδομένων. Ορίζεται ως ο μέσος όρος του τετραγώνου της διαφοράς μεταξύ των πραγματικών και των εκτιμώμενων τιμών.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Όπου N ο αριθμός του δείγματος, $\hat{y}_i$ είναι η πρόβλεψη της τιμής του y και y_i η πραγματική τιμή.

4.2. Μέσο Απόλυτο Σφάλμα

Η απόλυτη διαφορά μεταξύ πραγματικών και προβλεπόμενων τιμών στο σύνολο των δεδομένων ορίζεται ως μέσο απόλυτο σφάλμα (MAE).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

Όπου N ο αριθμός του δείγματος, $\hat{y}_i$ είναι η πρόβλεψη της τιμής του y και y_i η πραγματική τιμή.

4.3. Μέσο Απόλυτο Ποσοστιαίο Σφάλμα

Το μέσο απόλυτο ποσοστιαίο σφάλμα (MAPE) ορίζεται ως ο μέσος όρος του απόλυτου σφάλματος εκφρασμένου σε ποσοστό για ένα δείγμα.

$$\text{M}APE = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{|y_i|} * 100$$

Όπου N ο αριθμός του δείγματος, $\hat{y}_i$ είναι η πρόβλεψη της τιμής του y και y_i η πραγματική τιμή.

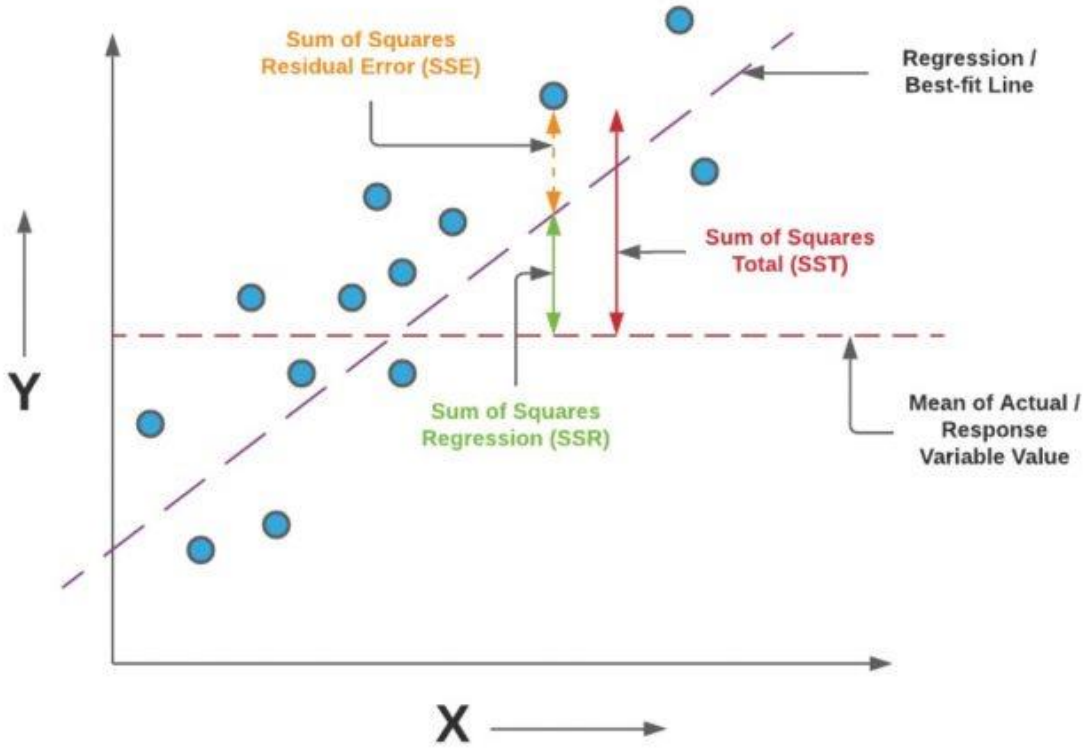
4.4. R-Squared

Ο συντελεστής προσδιορισμού είναι ο λόγος του αθροίσματος των τετραγώνων παλινδρόμησης (SSR) και του αθροίσματος των τετραγώνων συνολικά (SST). Το άθροισμα τετραγώνων παλινδρόμησης (SSR) αντιπροσωπεύει τη συνολική απόκλιση όλων των προβλεπόμενων τιμών που βρίσκονται στην ευθεία ή στο επίπεδο παλινδρόμησης από τη μέση τιμή όλων των τιμών των μεταβλητών απόκρισης. Το άθροισμα των συνολικών τετραγώνων (SST) αντιπροσωπεύει τη συνολική διακύμανση των πραγματικών τιμών από τη μέση τιμή όλων των τιμών των μεταβλητών απόκρισης. Η τιμή R-squared χρησιμοποιείται για τη μέτρηση της καλής

προσαρμογής ή της γραμμής καλύτερης προσαρμογής και όσο αυξάνεται τόσο καλύτερο είναι το μοντέλο παλινδρόμησης και είναι ανάμεσα $(-\infty, 1]$. [20]

$$R^2 = 1 - \frac{SSR}{SST},$$

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2}$$



Εικόνα 9. R-Squared Πηγή: Vitalflux

4.5. Συσχέτιση (Correlation)

Η συσχέτιση είναι μια στατιστική τεχνική η οποία υλοποιείται για τη μέτρηση της γραμμικής σχέσης μεταξύ δύο μεταβλητών και τον υπολογισμό της συσχέτισής τους. Εάν μια αύξηση (ή μείωση) σε μια μεταβλητή προκαλεί αντίστοιχη αύξηση (ή μείωση) σε μια άλλη, τότε οι δύο

μεταβλητές λέγεται ότι συσχετίζονται άμεσα. Αντίστοιχα, εάν η αύξηση της μίας προκαλεί μείωση της άλλης ή το αντίστροφο, τότε οι μεταβλητές λέγεται ότι συσχετίζονται έμμεσα. Εάν μια μεταβολή σε μια ανεξάρτητη μεταβλητή δεν προκαλεί μεταβολή στην εξαρτημένη μεταβλητή, τότε είναι ασυσχέτιστες. Έτσι, η συσχέτιση μπορεί να είναι θετική (άμεση συσχέτιση), αρνητική (έμμεση συσχέτιση) ή μηδενική. Η σχέση αυτή δίνεται από τον συντελεστή συσχέτισης (correlation coefficient).

Κεφάλαιο 5: Αποτελέσματα

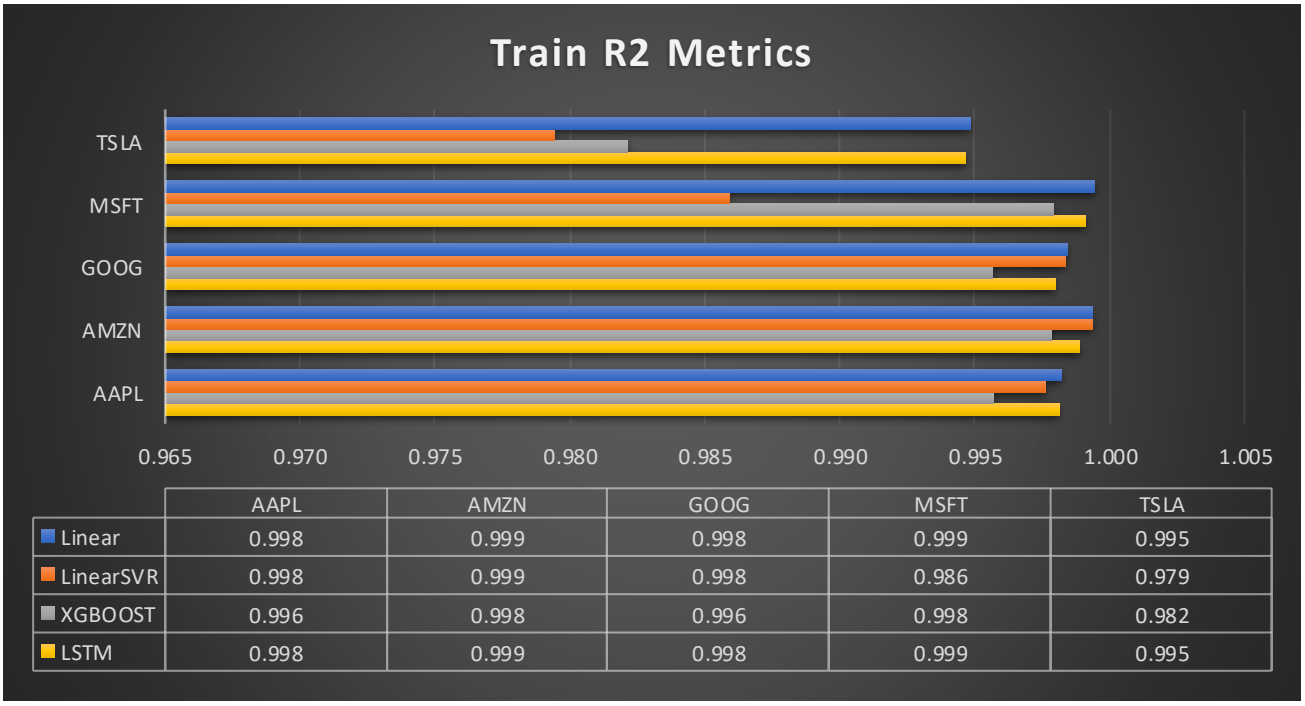
Αφού εκπαιδεύτηκαν τα μοντέλα, προσπαθήσαμε να τα χρησιμοποιήσουμε για την πρόβλεψη τιμής σε σημερινά δεδομένα. Δυστυχώς αυτή η προσπάθεια απέτυχε καθώς το σύνολο των δεδομένων που εκπαιδεύτηκαν τα μοντέλα ήταν κατά την περίοδο 2015 – 2020 με τις τιμές των μετοχών τώρα να έχουν πολύ μεγάλη απόκλιση από τις μετοχές τότε (για παράδειγμα η τιμή μιας μετοχής Tesla ήταν 29.53$ στις 3/1/2020 ενώ τώρα η τιμή της κυμαίνεται στα 300$). Έτσι, καταλήξαμε να χρησιμοποιούμε τα τελευταία 10% του συνόλου δεδομένων που διαθέταμε για την πρόβλεψη τιμής.

Στα γραφήματα κατά την εκπαίδευση, επαλήθευση και δοκιμή (εκτός των μετρικών) θα παρουσιαστεί η μετοχή της Tesla (TSLA), αλλά παρόμοια γραφήματα έχουν εξαχθεί για όλες τις μετοχές.

5.1. Εκπαίδευση

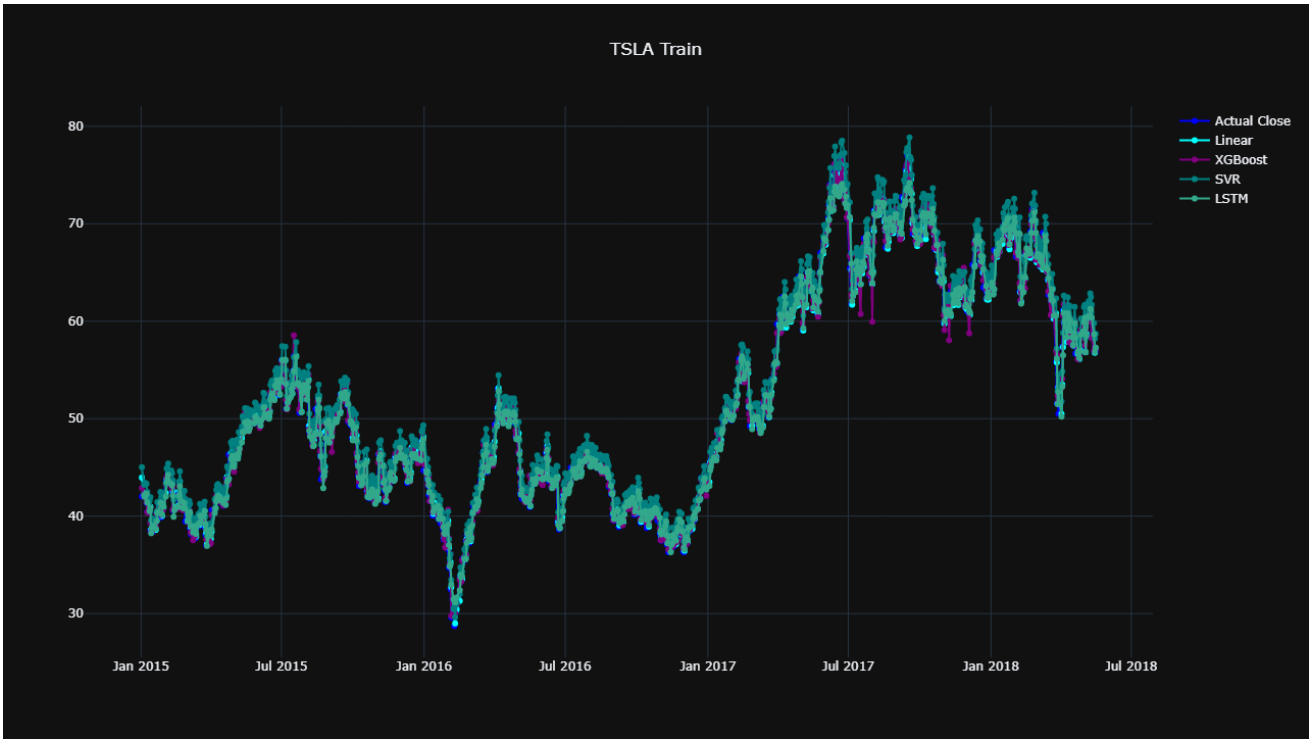
	AAPL			AMZN			GOOG			MSFT			TSLA		
	mae	mape	mse	mae	mape	mse	mae	mape	mse	mae	mape	mse	mae	mape	mse
Linear	0.153	0.004	0.144	5.358	0.006	165.34	3.496	0.004	71.316	0.285	0.004	0.363	0.355	0.007	0.620
LinearSVR	0.305	0.008	0.192	5.495	0.005	160.54	4.445	0.005	74.771	2.876	0.039	9.504	1.440	0.027	2.477
XGBOOST	0.386	0.010	0.349	13.513	0.013	552.57	8.805	0.010	197.94	0.765	0.010	1.407	1.006	0.019	2.152
LSTM	0.242	0.007	0.151	11.499	0.012	283.40	5.893	0.007	92.392	0.540	0.007	0.596	0.423	0.008	0.643

Εικόνα 10. Αποτελέσματα μετρικών εκπαίδευσης

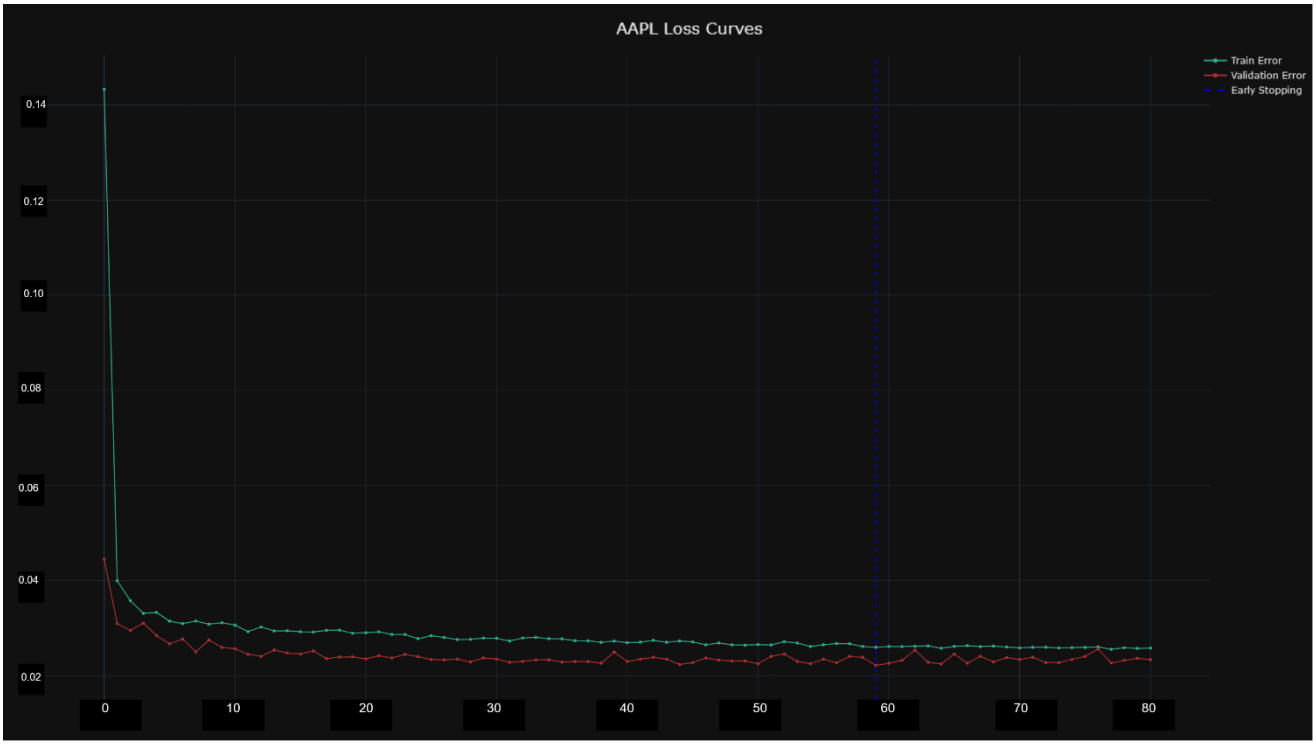


	AAPL	AMZN	GOOG	MSFT	TSLA
Linear	0.998	0.999	0.998	0.999	0.995
LinearSVR	0.998	0.999	0.998	0.986	0.979
XGBOOST	0.996	0.998	0.996	0.998	0.982
LSTM	0.998	0.999	0.998	0.999	0.995

Εικόνα 11. Αποτελέσματα μετρικής R2 εκπαίδευσης



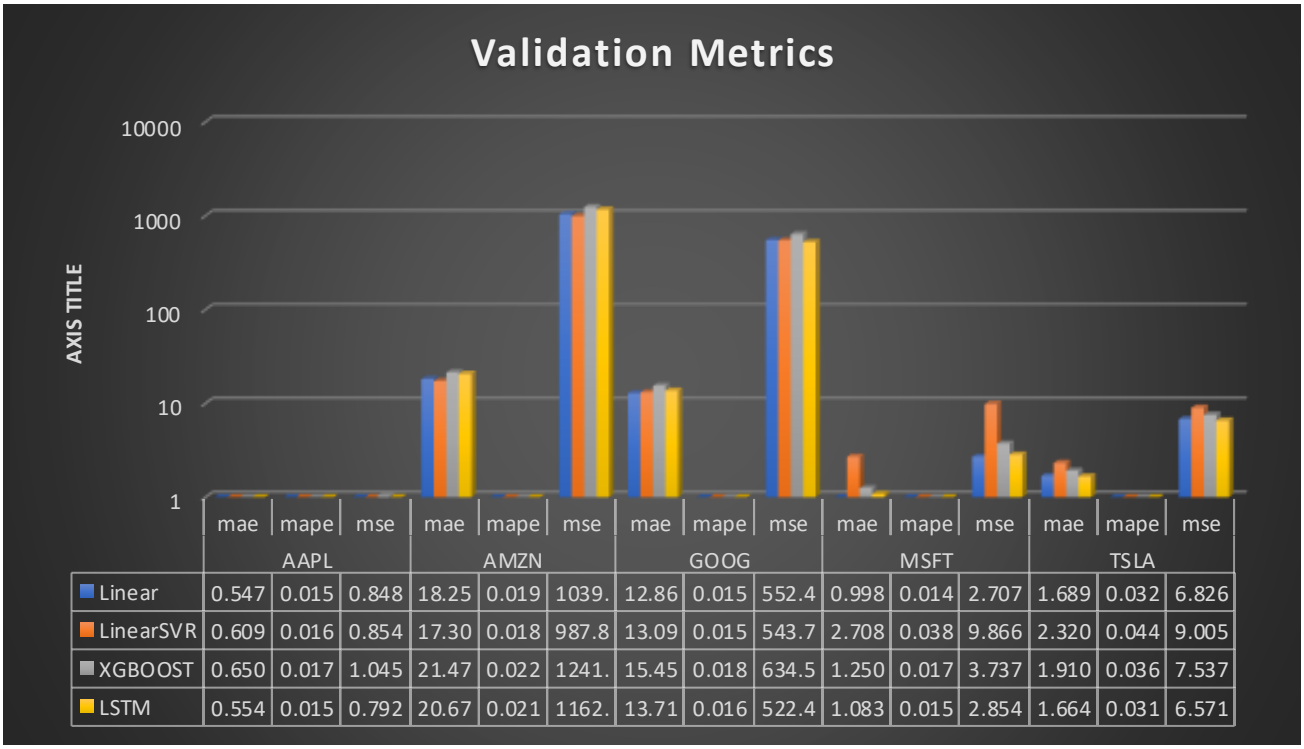
Εικόνα 12. Αποτελέσματα Εκπαίδευσης της μετοχής TSLA



Εικόνα 13. Αποτελέσματα Εκπαίδευσης της μετοχής TSLA – Early Stopping

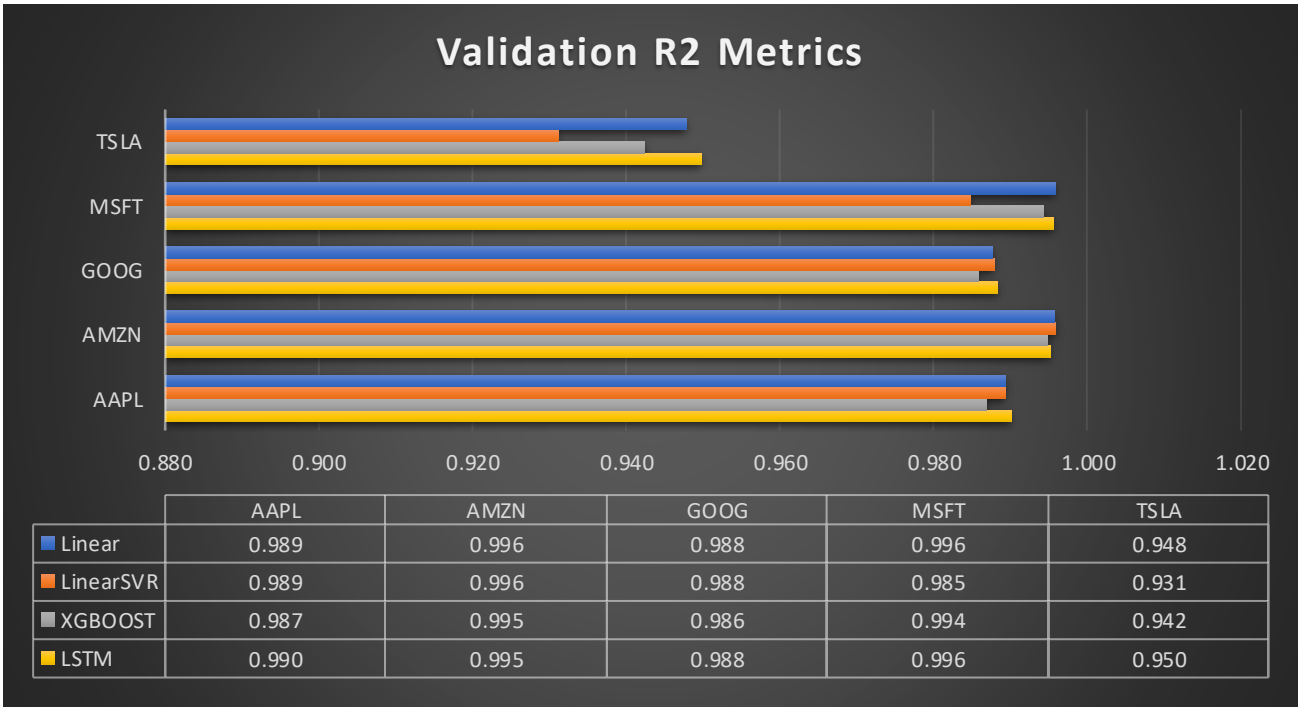
Στην *Εικόνα 12*, παρουσιάζεται η εκπαίδευση των μοντέλων και στην *Εικόνα 13* παρατηρούμε αρχικά έναν ραγδαίο ρυθμό εκμάθησης του LSTM μοντέλου όπου λόγω του Early Stopping σταματάει την εκπαίδευση στις 60 περίπου εποχές (με Train Error = 0.024).

5.2. Επαλήθευση



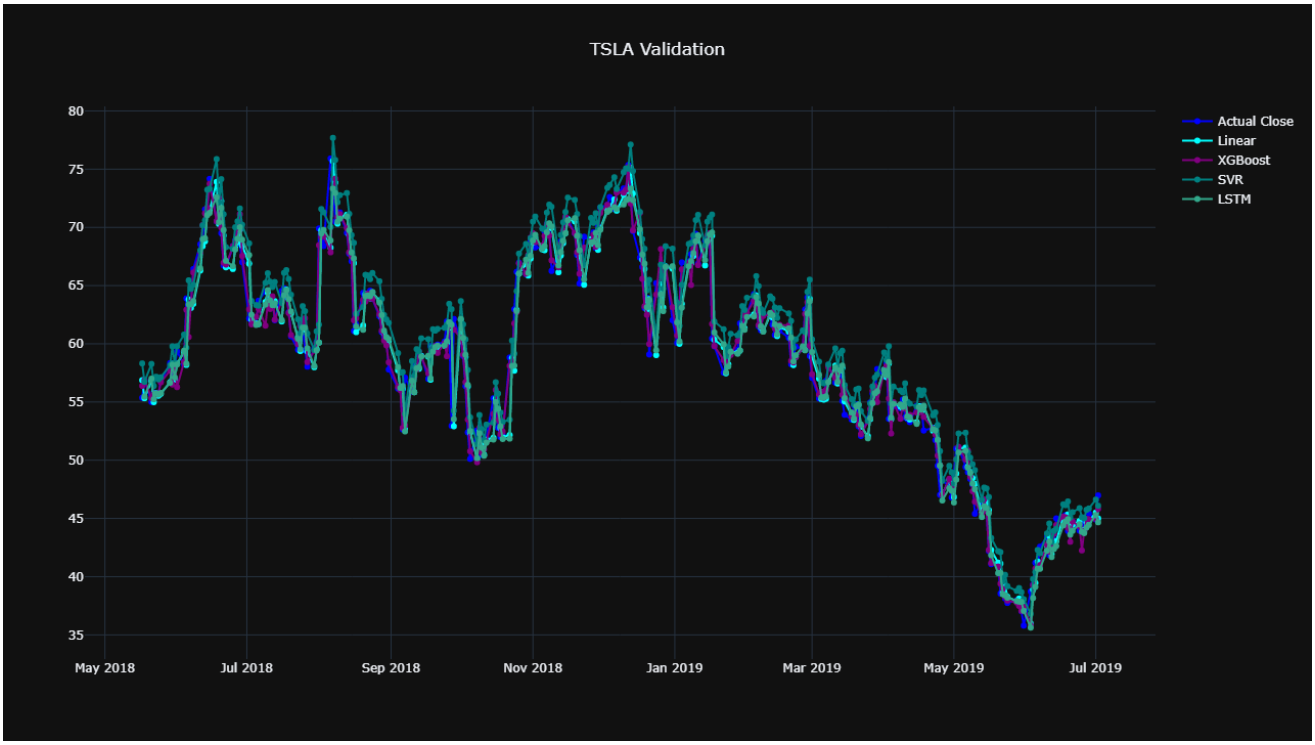
	AAPL			AMZN			GOOG			MSFT			TSLA		
	mae	mape	mse	mae	mape	mse	mae	mape	mse	mae	mape	mse	mae	mape	mse
Linear	0.547	0.015	0.848	18.25	0.019	1039.	12.86	0.015	552.4	0.998	0.014	2.707	1.689	0.032	6.826
LinearSVR	0.609	0.016	0.854	17.30	0.018	987.8	13.09	0.015	543.7	2.708	0.038	9.866	2.320	0.044	9.005
XGBOOST	0.650	0.017	1.045	21.47	0.022	1241.	15.45	0.018	634.5	1.250	0.017	3.737	1.910	0.036	7.537
LSTM	0.554	0.015	0.792	20.67	0.021	1162.	13.71	0.016	522.4	1.083	0.015	2.854	1.664	0.031	6.571

Εικόνα 14. Αποτελέσματα μετρικών επαλήθευσης



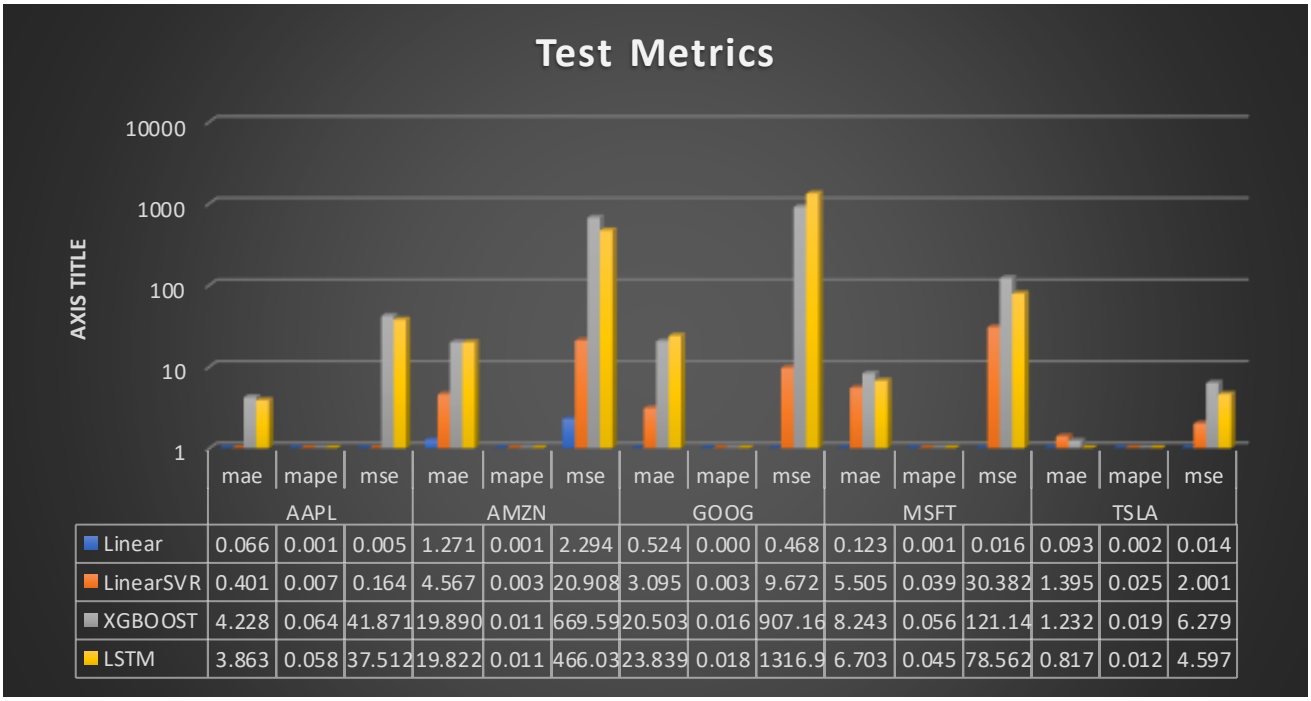
	AAPL	AMZN	GOOG	MSFT	TSLA
Linear	0.989	0.996	0.988	0.996	0.948
LinearSVR	0.989	0.996	0.988	0.985	0.931
XGBOOST	0.987	0.995	0.986	0.994	0.942
LSTM	0.990	0.995	0.988	0.996	0.950

Εικόνα 15. Αποτελέσματα μετρικής R2 επαλήθευσης



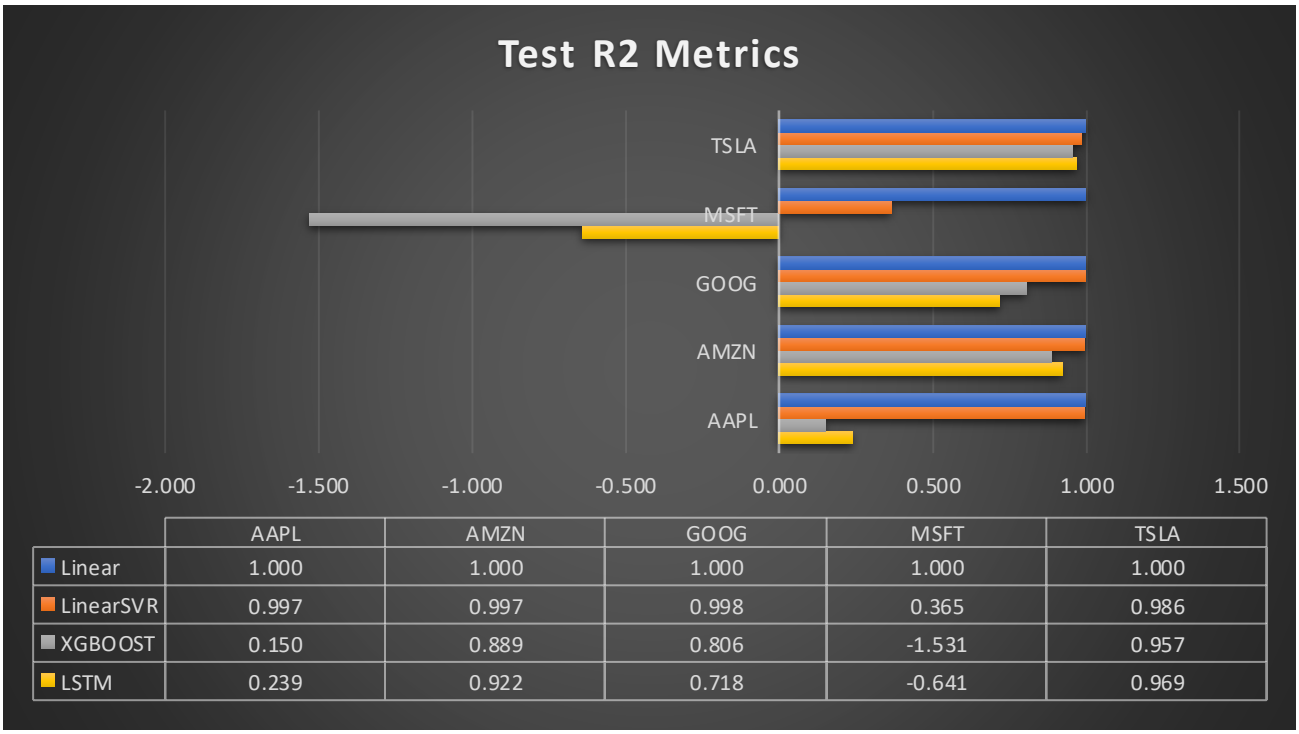
Εικόνα 16. Αποτελέσματα επαλήθευσης της μετοχής TSLA

5.3. Δοκιμή



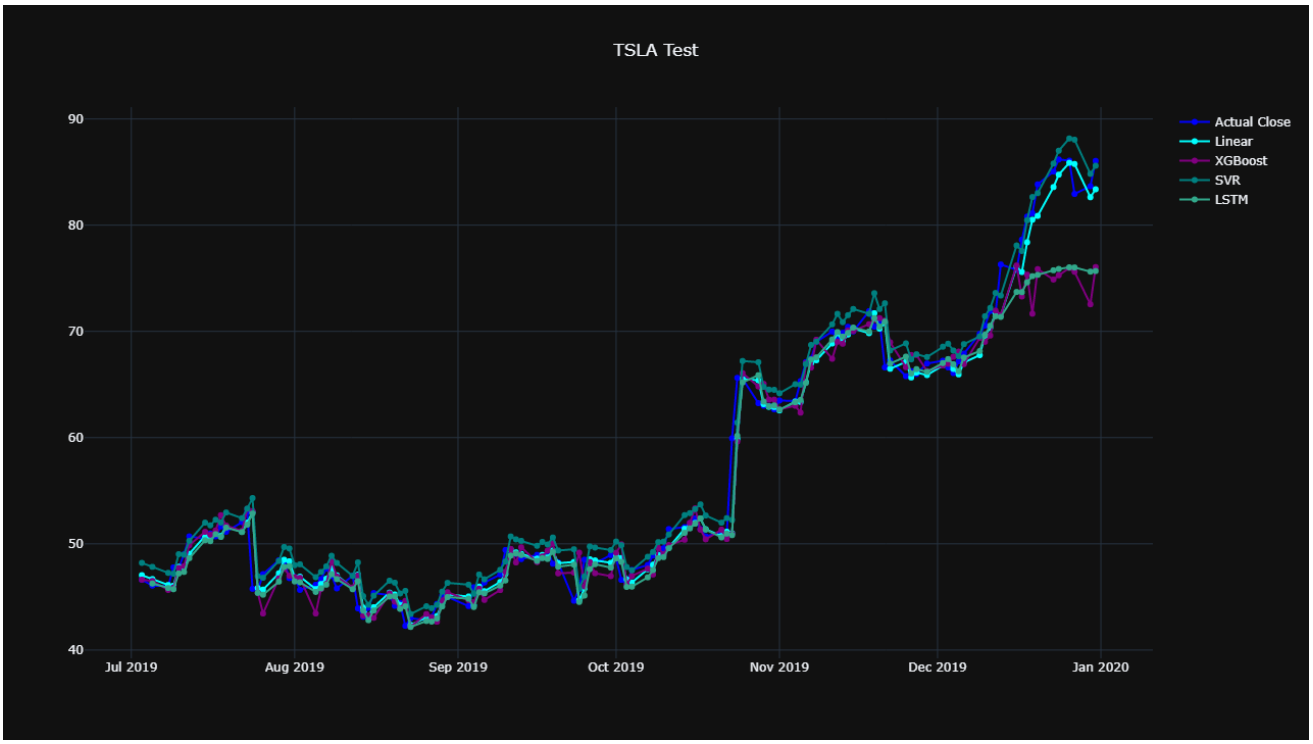
	AAPL			AMZN			GOOG			MSFT			TSLA		
	mae	mape	mse	mae	mape	mse	mae	mape	mse	mae	mape	mse	mae	mape	mse
Linear	0.066	0.001	0.005	1.271	0.001	2.294	0.524	0.000	0.468	0.123	0.001	0.016	0.093	0.002	0.014
LinearSVR	0.401	0.007	0.164	4.567	0.003	20.908	3.095	0.003	9.672	5.505	0.039	30.382	1.395	0.025	2.001
XGBOOST	4.228	0.064	41.871	19.890	0.011	669.59	20.503	0.016	907.16	8.243	0.056	121.14	1.232	0.019	6.279
LSTM	3.863	0.058	37.512	19.822	0.011	466.03	23.839	0.018	1316.9	6.703	0.045	78.562	0.817	0.012	4.597

Εικόνα 17. Αποτελέσματα μετρικών δοκιμής



	AAPL	AMZN	GOOG	MSFT	TSLA
Linear	1.000	1.000	1.000	1.000	1.000
LinearSVR	0.997	0.997	0.998	0.365	0.986
XGBOOST	0.150	0.889	0.806	-1.531	0.957
LSTM	0.239	0.922	0.718	-0.641	0.969

Εικόνα 18. Αποτελέσματα μετρικής R2 δοκιμής



Εικόνα 19. Αποτελέσματα δοκιμής της μετοχής TSLA

Παρατηρείται πως όλα τα μοντέλα έχουν εκπαιδευτεί επαρκώς και προσεγγίζουν αρκετά τις πραγματικές τιμές, με XGBoost και LSTM να έχουν μια μικρή απόκλιση στα τελευταία δεδομένα που εξηγείται ενδεχομένως από την απότομη αύξηση της μετοχής (περίπου 80% σε 6 μήνες). Επίσης, τα γραμμικά μοντέλα φαίνεται να έχουν καλύτερα αποτελέσματα στην πρόβλεψη της τιμής της μετοχής για την επόμενη μέρα, με τα υπόλοιπα μοντέλα να προσεγγίζουν καλύτερα στην προσπάθεια πρόβλεψης παραπάνω από 1 ημέρες (π.χ. 3 ημέρες).

Κεφάλαιο 6: Συμπεράσματα

Στην διπλωματική αυτή δημιουργήθηκαν μοντέλα πρόβλεψης της τιμής μια μετοχής με την βοήθεια της ανάλυσης συναισθήματος. Σύμφωνα με τα αποτελέσματα που εξήχθησαν παρατηρείται πως το συναίσθημα του κοινού (στην περίπτωση μας στο twitter) μπορεί να επηρεάσει την τιμή μιας μετοχής, αλλά συνήθως γίνεται το αντίστροφο δηλαδή η τιμή μιας μετοχής

θα επηρεάσει και το συναίσθημα του κοινού. Μεγάλη συσχέτιση παρατηρείται στην μετοχή της Tesla, όπου ανά τα χρόνια ο Διευθύνων Σύμβουλος της Elon Musk μέσω μηνυμάτων του στο Twitter είχε επιρροή στην τιμή της μετοχής της εταιρείας του. Αντίθετα, η Amazon φαίνεται να μην επηρεάζεται τόσο από το συναίσθημα του Twitter.

Τέλος, αξίζει να σημειωθεί πως η ανάλυση που πραγματοποιήθηκε δεν λαμβάνει υπόψιν πολλούς παράγοντες. Για παράδειγμα το σύνολο των δεδομένων μας δεν έχει χαρτογραφήσει το πραγματικό συναίσθημα του κοινού, αλλά μόνο αγγλόφωνους ανθρώπους οι οποίοι είναι χρήστες του twitter. Τα άτομα που επενδύουν σε μετοχές δεν έχουν άμεση συσχέτιση με τα άτομα που σχολιάζουν στο Twitter, αλλά έμμεση καθώς οι επενδυτικές αποφάσεις μπορεί να επηρεάζονται από το γενικότερο συναίσθημα του κοινού.

Όλα τα παραπάνω παραμένουν τομείς μελλοντικής έρευνας, επεκτείνοντας ίσως την παρούσα διπλωματική με την υλοποίηση ενός καλύτερου μοντέλου ανάλυσης συναισθήματος, την δημιουργία ενός dataset σε τωρινά tweets ή ακόμη και της υλοποίησης ενός Ensemble μοντέλου που συνδυάζει όλα τα μοντέλα που χρησιμοποιήθηκαν.

Κεφάλαιο 7: Παράρτημα – Κώδικας

Η υλοποίηση της παραπάνω διπλωματικής έγινε με την βοήθεια των βιβλιοθηκών pyTorch, Scikit-learn, pandas, plotly και Natural Language ToolKit (NLTK).

7.1. Προεργασία δεδομένων

Μετατροπή του epoch σε μορφή Date στα δεδομένα:

```
def convertEpochToDate(epoch):
    return dt.datetime.utcfromtimestamp(epoch).strftime("%Y-%m-%d")
```

Συγχώνευση των δύο συνόλων δεδομένων σε ένα και εξαγωγή μόνο των τελευταίων 300.000 tweets για κάθε μετοχή:

```
def merge_tables_get_last_tweets(tweets_num):
    tweets = pd.read_csv("data/OriginalDataset/Tweet.csv", index_col=False)
    ids = pd.read_csv("data/OriginalDataset/Company_Tweet.csv",
index_col=False)

    df = pd.merge(tweets, ids, on="tweet_id").dropna()  # merge the csv's on
tweet_id

    df["body"] = df["body"].apply(cleanText)
    df["post_date"] = df["post_date"].apply(convertEpochToDate)  # convert to
date
    create_folder('data/tweets_by_stock')

    # get last 300.000 tweets for each company
    for company_name in ["AMZN", "AAPL", "GOOG", "MSFT", "TSLA"]:
        df.loc[df.ticker_symbol == company_name].dropna().iloc[-
tweets_num:].to_csv(
            f"data/tweets_by_stock/{company_name}_tweets.csv", index=False)
```

Καθαρισμός δεδομένων (data cleaning):

```
def cleanMessage(message):
    message = re.sub('@[A-Za-z0-9]+', '', message)  # Remove @mentions
    message = re.sub('#', '', message)  # Remove '#' hash tag
    message = re.sub('RT[\s]+', '', message)  # Remove RT (retweets tag)
    message = re.sub('https?:\/\/\S+', '', message)  # Remove links
    message = re.sub('/^\s*$/', '', message)  # Remove empty tweets or tweets
with only space
    message = re.sub('^\d+$', '', message)  # Remove tweets containing only
numbers
    return message
```

Εξαγωγή λίστας με ημέρες που το χρηματιστήριο είναι κλειστό:

```
def missing_dates(company):
    df = pd.read_csv(f"data/{company}_stock_prices.csv")
    df = df.set_index('Date')
    df.index = pd.to_datetime(df.index)


    missing_dates_list = pd.date_range(start="2015-01-01", end="2019-12-
30").difference(df.index)
    # print(df_copy.loc[df_copy.Date == "2014-12-31"])  # missing_dates[0] -
timedelta(days=1))
    return missing_dates_list
```

7.2. Ιστορικά δεδομένα τιμών μετοχής

Εξαγωγή ιστορικών δεδομένων των μετοχών μέσω της βιβλιοθήκης yfinance.

```
import yfinance as yf
import pandas as pd
import config
from utils import create_folder

def stock_prices():
    create_folder("data/new_data/stock_prices")
    for company_name in config.company_names:
        data = yf.download(f"{company_name}", start="2014-12-31",
                                                end="2021-12-18").reset_index()
        data["Date"] = pd.to_datetime(data["Date"].dt.strftime('%Y/%m/%d'))

       data.to_csv(f"data/new_data/stock_prices/{company_name}_stock_price.cs
v", index=False)
```

7.3. Ανάλυση συναισθήματος

Ιστορικότητα συναισθήματος:

```
def get_historical_sentiment():
    for company_name in config.company_names:
        sentiment_avg(company_name, sentiment_Analysis_Parallel(company_name))
# sentiment analysis and write to csv
```

Ανάλυση λέξεων:

```
def wordAnalyser(df, sentiment, threadNum):
    sents = []

    for tweet in df["body"]:
        analyzer = SentimentIntensityAnalyzer().polarity_scores(str(tweet))
        neg = analyzer['neg']
        pos = analyzer['pos']
        if neg > pos:
            sents.append(-1)
        elif pos > neg:
            sents.append(1)
        elif pos == neg:
            sents.append(0)
    sentiment[threadNum] = sents
    print(f"Process {threadNum}: finished")
```

Ανάλυση συναισθήματος με παράλληλη επεξεργασία (processes):

```
def sentiment_Analysis_Parallel(company_name):
    df = pd.read_csv(f"data/tweets_by_stock/{company_name}_tweets.csv")
    df = df[["post_date", "body"]]
    manager = Manager()
    sentiments = manager.dict()
    threads = [0] * NUMBER_OF_PROCESSES
    arr = []
    for i in range(NUMBER_OF_PROCESSES-1):
        arr.append(df.iloc[i*(len(df)//NUMBER_OF_PROCESSES):(i + 1) *
int(len(df) / NUMBER_OF_PROCESSES)])
    arr.append(df.iloc[(i+1)*(len(df)//NUMBER_OF_PROCESSES):])
    for process in range(NUMBER_OF_PROCESSES):
        threads[process] = Process(target=wordAnalyser,
                                   args=(arr[process],
                                         sentiments, process,))
        threads[process].start()

    for process in range(NUMBER_OF_PROCESSES):
        threads[process].join()

    temp_sentiments = np.zeros(0)
    for process in range(NUMBER_OF_PROCESSES):
        temp_sentiments = np.append(temp_sentiments,
np.array(sentiments[process]))
    df["sentiment"] = pd.DataFrame(temp_sentiments).to_numpy().flatten()
    # drop rows where sentiment is 0 (neutral)
    df = df[df.sentiment != 0]
    return df.reset_index(drop=True)
```

Εξαγωγή μέσου όρου συναισθήματος για κάθε μέρα:

```
def sentiment_avg(company, df):
    # df = pd.read_csv(f"data/new_data/{company}_sentiment.csv")
    pos_sentiment = 0
    neg_sentiment = 0
    data = []
    for date in df["post_date"].unique():
        # Get all rows of positive sentiment (1) for specific date
        pos_sentiment = len(df.loc[(df.post_date == date) & (df.sentiment ==
1)].index)
        # Get all rows of negative sentiment (-1) for specific date
        neg_sentiment = len(df.loc[(df.post_date == date) & (df.sentiment == -
1)].index)
        # print(f"Date = {date}, Pos = {pos_sentiment}, Neg =
{neg_sentiment}")
        # print("Positive percentage = " + str(percentage(pos_sentiment,
(pos_sentiment + neg_sentiment))))
        data.append([date, str(percentage(pos_sentiment, (pos_sentiment +
neg_sentiment)))])
    new_df = pd.DataFrame(data, columns=['Date', 'Sentiment'])
    create_folder("data/new_data/stock_daily_avg_sentiment/v2")

new_df.to_csv(f"data/new_data/stock_daily_avg_sentiment/v2/{company}_avg_senti
ment.csv", index=False)
```

7.4. LSTM

Υλοποίηση LSTM:

```
class LSTM(nn.Module):
    def __init__(self, input_size, hidden_size, num_layers, output_dim,
drop=0.1, batch_first=True):
        super(LSTM, self).__init__()
        self.lstm = nn.LSTM(input_size, hidden_size, num_layers, batch_first)
        self.drop = Dropout(drop)
        self.seq = nn.Sequential(
            nn.Linear(hidden_size, hidden_size),
            nn.ReLU(),
            nn.Linear(hidden_size, output_dim),
            nn.Sigmoid()
        )
```

```
def forward(self, x):
    output, _ = self.lstm(x)
    output = self.drop(output)
    output = self.seq(output)
    return output
```

Μέθοδος ολισθαινόντων παραθύρων:

```
def sliding_windows(features, labels, batch_size, window_step=1):
    features_array = []
    features = features.astype(float)
    labels_array = []
    labels = labels.astype(float)
    for i in range(0, features.shape[0] - batch_size + 1, window_step):
        features_array.append(features[i: i + batch_size].values)
        labels_array.append(labels[i: i + batch_size].values)

    features_array = np.array(features_array)
    labels_array = np.array(labels_array)

    return features_array, labels_array
```

7.5. Linear, XGBoost, LinearSVR Pipeline

Αποτελέσματα - Μετρικές και παραγωγή γραφημάτων/εικόνων:

```
def model_results(company, x, y, mode, model_name):
    results = pd.DataFrame()
    prediction = load_pickle(company, model_name, "models").predict(x)
    results["Actual Close"] = y.shift(1).dropna()
    results["Prediction"] = pd.Series(prediction, index=y.index)

    metrics = pd.DataFrame()


    metrics["mae"] = [mean_absolute_error(results["Actual Close"],
results["Prediction"])]
    metrics["mape"] = [mean_absolute_percentage_error(results["Actual Close"],
results["Prediction"])]
    metrics["mse"] = [mean_squared_error(results["Actual Close"],
results["Prediction"])]
    metrics["r2"] = [r2_score(results["Actual Close"], results["Prediction"])]

    create_folder(f"data/metrics/{mode}")
    metrics.to_csv(f"data/metrics/{mode}/{model_name}_metrics_{company}.csv")

    layout = go.Layout(
        autosize=False,
        width=1280,
        height=720
    )
    fig = go.Figure(layout=layout)
    fig.update_layout(template=pio.templates['plotly_dark'])
    for column,color in zip(results.columns,["#32a88b","#a83232"]):
        fig.add_trace(go.Scatter(x=results.index, y=results[column],
mode='lines+markers', line=dict(color=color), name=column))
    create_folder(f"charts/{mode}")
    fig.write_html(f"charts/{mode}/{model_name}_pred_{company}.html")
    fig.write_image(f"charts/{mode}/{model_name}_pred_{company}.png")
```

Εκπαίδευση LSTM και παραγωγή γραφημάτων:

```
def train(x_train, x_val, y_train, y_val, company):
    batch_size = 5

    feature_scaler = MinMaxScaler(feature_range=(0, 1))
    label_scaler = MinMaxScaler(feature_range=(0, 1))
    # Transform only stock market Data a.e no Sentiment
    x_train.loc[:, x_train.columns != 'Sentiment'] = feature_scaler.fit_trans-
form(x_train.loc[:, x_train.columns != 'Sentiment'])
    x_val.loc[:, x_val.columns != 'Sentiment'] = feature_scaler.trans-
form(x_val.loc[:, x_val.columns != 'Sentiment'])

    y_train = pd.DataFrame(label_scaler.fit_transform(y_train.values.reshape(-
1, 1)), columns=["Label"],
                           index=y_train.index)
    y_val = pd.DataFrame(label_scaler.transform(y_val.values.reshape(-1, 1)),
columns=["Label"],
                         index=y_val.index)

    x_train, y_train = sliding_windows(x_train, y_train, batch_size)
    x_val, y_val = sliding_windows(x_val, y_val, batch_size)

    model = LSTM(x_train.shape[2], 64, 2, 1)
    loss_function = nn.L1Loss()
    optimizer = Adam(model.parameters(), lr=0.001)

    train_error = np.empty(0)
    val_error = np.empty(0)

    train_loader = DataLoader(TimeseriesValues(x_train, y_train), batch_size,
shuffle=False, drop_last=True)
    validation_loader = DataLoader(TimeseriesValues(x_val, y_val), batch_size,
shuffle=False, drop_last=True)

    epoch = 0
    best_epoch = None
    while True:
        print(f"Epoch = {epoch}")
        model.train()
        error = []
        # train model
        for x, y in train_loader:
            y_train_prediction = model(x.float())
            loss = loss_function(y_train_prediction.squeeze(),
y.squeeze().float())
            error.append(loss.detach().item())
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()
        train_error = np.append(train_error, (sum(err) / len(err)))

        model.eval()
        error = []
```

```
        # validate model
        for x, y in val_loader:
                y_validation_prediction = model(x.float())
                loss = loss_function(y_validation_prediction.squeeze(),
y.squeeze().float())
                error.append(loss.detach().item())
        val_error = np.append(val_error, (sum(err) / len(err)))
        print(f"Test_error = {(sum(err) / len(err))}")
        if epoch > 50:
                if val_error[-1] <= val_error[51:].min():
                        best_model = copy.deepcopy(model)
                        best_epoch = epoch
                if best_epoch is not None and epoch - best_epoch > 20 or epoch
> 300:
                        break
        epoch += 1

# save best model
    save_pickle(company, "lstm", best_model, "models")

# save scalers
    save_pickle(company, "feature_scaler", feature_scaler, "scalers")
    save_pickle(company, "label_scaler", label_scaler, "scalers")

    print(f"Best_epoch = {best_epoch}, Best_error= {val_error.min()}")
    df = pd.DataFrame(train_error.reshape(-1, 1), columns=['Train Error'])
    df['Validation Error'] = val_error.reshape(-1, 1)
    layout = go.Layout(
        autosize=False,
        width=1280,
        height=720
)
    fig = go.Figure(layout=layout)
    fig.update_layout(template=pio.templates['plotly_dark'], title=company+"
Loss Curve")
    for column, color in zip(df.columns, ["#32a88b", "#a83232"]):
        fig.add_trace(
            go.Scatter(x=df.index, y=df[column], mode='lines+markers',
line=dict(color=color), name=column))
    create_folder(f"charts/loss_curves")
    fig.add_vline(best_epoch, line_dash="dash", line_color="blue")
    fig.write_html(f"charts/loss_curves/lstm_training_{company}.html")
    fig.write_image(f"charts/loss_curves/lstm_training_{company}.png")
```

Αποτελέσματα LSTM:

```
def lstm_results(company, data, true_next_close, mode):
    """
    Feeds data to lstm model and returns the prediction
    :param mode: chart type
    :param company: name of company
    :param data: Features
    :param true_next_close: Labels
    :return: predictions
    """
    feature_scaler = load_pickle(company, "feature_scaler", "scalers")
    label_scaler = load_pickle(company, "label_scaler", "scalers")
    model = load_pickle(company, "lstm", "models")
    results = pd.DataFrame()
    results["Actual Close"] = true_next_close.shift(1)
    index = data.index
    data.loc[:, data.columns != 'Sentiment'] =
feature_scaler.transform(data.loc[:, data.columns != 'Sentiment'])
    data = data.values
    loader = DataLoader(TimeseriesValues(data), index.shape[0], shuffle=False,
drop_last=True)

    model.eval()
    for i in loader:
        i = i.reshape(1, -1, data.shape[1])
        predictions = model(i.float())

    results["Prediction"] =
label_scaler.inverse_transform(predictions.detach().numpy().reshape(1, -
1)).reshape(-1, 1)

    results = results.dropna()
    metrics = pd.DataFrame()

    metrics["mae"] = [mean_absolute_error(results["Actual Close"],
results["Prediction"])]
    metrics["mape"] = [mean_absolute_percentage_error(results["Actual Close"],
results["Prediction"])]
    metrics["mse"] = [mean_squared_error(results["Actual Close"],
results["Prediction"])]
    metrics["r2"] = [r2_score(results["Actual Close"], results["Prediction"])]

    create_folder(f"data/metrics/{mode}")
    metrics.to_csv(f"data/metrics/{mode}/lstm_metrics_{company}.csv")

    layout = go.Layout(
        autosize=False,
        width=1280,
        height=720
    )
    fig = go.Figure(layout=layout)
    fig.update_layout(template=pio.templates['plotly_dark'])
    for column, color in zip(results.columns, ["#32a88b", "#a83232"]):
        fig.add_trace(
            go.Scatter(x=results.index, y=results[column], mode='lines+markers',
line=dict(color=color), name=column))
    create_folder(f"charts/{mode}")
    fig.write_html(f"charts/{mode}/lstm_pred_{company}.html")
```

7.6. Σειριοποίηση – Αποσειριοποίηση δεδομένων

```
def save_pickle(company_name, model_type, model, path):
    create_folder(path)
    with open(f"{path}/{company_name}_{model_type}.pickle", 'wb') as file:
        pickle.dump(model, file)
```

```
def load_pickle(company_name, model_type, path):
    create_folder(path)
    return pickle.load(open(f"{path}/{company_name}_{model_type}.pickle",
'rb'))
```

7.7. Main μέθοδος

Εξαγωγή ιστορικών δεδομένων, τιμών και έπειτα εκπαίδευση των μοντέλων και παραγωγή αποτελεσμάτων:

```
if __name__ == "__main__":
   get_historical_sentiment()
   stock_prices()
   train_models_and_get_results()
```

Διάβασμα αρχείων, διαχωρισμός σε training, test και validation set:

```
def train_models_and_get_results():
   # Linear, XGBOOST, SVR train/test
   for company_name in config.company_names:
      company_sentiment = pd.read_csv(f"data/sentiment/{company_name}.csv",
index_col=0)
      company_data =
pd.read_csv(f"data/stock_prices/{company_name}_stock_price.csv", index_col=0)
      company_data['Sentiment'] = company_sentiment
      company_data['Label'] = company_data['Close'].shift(-1)
      company_data = company_data.dropna().round(5)

      # Lessen Dimensions for Non DL models
      linear_data = company_data[['Close', 'Sentiment', 'Label']]
      # get 90% of data for training/validation
      train_data = linear_data.iloc[:int(linear_data.shape[0] * 0.9)]
      # get 10% of data for prediction:)
      test_data = linear_data.iloc[int(linear_data.shape[0] * 0.9):]
      x_train, x_val, y_train, y_val = train_test_split(train_data.loc[:,
                                                linear_data.columns != 'Label'],
                                                train_data['Label'],
                                                train_size=0.75,
                                                random_state=11)
```

Εκπαίδευση Linear, XGBoost, LinearSVR και παραγωγή αποτελεσμάτων:

```
    models = {"Linear": LinearRegression(), "XGBoost": XGBRegressor(), "SVR":
LinearSVR()}
    for model_name in models:
      model_training(x_train, y_train, company_name, model_name,
models[model_name])

      model_results(company_name, x_train.sort_index(), y_train.sort_index(),
"train", model_name)
      model_results(company_name, x_val.sort_index(), y_val.sort_index(),
"val", model_name)
      model_results(company_name, test_data.loc[:, linear_data.columns !=
'Label'], test_data['Label'], "test",model_name)
```

Διαχωρισμός training-test set, εκπαίδευση LSTM και παραγωγή αποτελεσμάτων:

```
# get 90% of data for training/validation
train_data = company_data.iloc[:int(company_data.shape[0] * 0.9)]
# get 10% of data for prediction:)
test_data = company_data.iloc[int(company_data.shape[0] * 0.9):]
x_train, x_val, y_train, y_val = train_test_split(train_data.loc[:,
company_data.columns != 'Label'],
                                    train_data['Label'], train_size=0.75,
random_state=11)
lstm_trainer.train(x_train.copy(), x_val.copy(), y_train.copy(), y_val.copy(),
company_name)
lstm_results(company_name, x_train.sort_index(), y_train.sort_index(),
"train")
lstm_results(company_name, x_val.sort_index(), y_val.sort_index(), "val")
# test lstm
lstm_results(company_name, test_data.loc[:, company_data.columns != 'Label'],
test_data['Label'], "test")
```

References

[1] A. Chandra, "Decision Making in the Stock Market: Incorporating Psychology with Finance," *ERN: Behavioral Economics (Topic),* 2008.

[2] J. Bollen, H. Mao and X.-J. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science, 2(1), March 2011, Pages 1-8,* October 2010.

[3] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI),* 2017.

[4] T. H. Nguyen and K. Shirai, "Topic Modeling based Sentiment Analysis on Social Media for Stock Market Prediction," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015.

[5] S. Asur and B. A. Huberman, "Predicting the Future with Social Media," 2010.

[6] Y. Mao, W. Wei and B. Wang, "Twitter volume spikes," in *Proceedings of the 7th Workshop on Social Network Mining and Analysis - SNAKDD \textquotesingle13*, 2013.

[7] E. J. Ruiz, V. Hristidis, C. Castillo, A. Gionis and A. Jaimes, "Correlating financial time series with micro-blogging activity," in *Proceedings of the fifth ACM international conference on Web search and data mining - WSDM \textquotesingle12,* 2012.

[8] T. Upton, "Tweets about the Top Companies from 2015 to 2020".

[9] H. K. Sul, A. R. Dennis and L. I. Yuan, "Trading on Twitter: The Financial Information Content of Emotion in Social Media," in *2014 47th Hawaii International Conference on System Sciences*, 2014.

[10] C. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 8, pp. 216-225, May 2014.

[11] H. Ma, E.-P. Li, A. C. Cangellaris and X. Chen, "Support Vector Regression-Based Active Subspace (SVR-AS) Modeling of High-Speed Links for Fast and Accurate Sensitivity Analysis," *IEEE Access*, vol. 8, p. 74339–74348, 2020.

[12] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," March 2016.

[13] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, p. 1735–1780, November 1997.

[14] C.-R. Ko and H.-T. Chang, "LSTM-based sentiment analysis for stock price forecast," *PeerJ Computer Science*, vol. 7, p. e408, March 2021.

[15] S. Pal, S. Ghosh and A. Nag, "Sentiment Analysis in the Light of LSTM Recurrent Neural Networks," *International Journal of Synthetic Emotions*, vol. 9, p. 33–39, January 2018.

[16] S. Elfwing, E. Uchibe and K. Doya, "Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning," February 2017.

[17] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann

Machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, Madison, 2010.

[18] G. E. Dahl, T. N. Sainath and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[19] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," December 2014.

[20] D. Chicco, M. J. Warrens and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Computer Science,* vol. 7, p. e623, July 2021.

[21] H. Zhao, O. Gallo, I. Frosio and J. Kautz, "Loss Functions for Neural Networks for Image Processing," November 2015.

[22] M. Wiering and M. van Otterlo, Eds., Reinforcement Learning, Springer Berlin Heidelberg, 2012.

[23] Q. Wang, Y. Ma, K. Zhao and Y. Tian, "A Comprehensive Survey of Loss Functions in Machine Learning," *Annals of Data Science,* vol. 9, p. 187–212, April 2020.

[24] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks," September 2019.

[25] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.,* vol. 15, pp. 1929-1958, 2014.

[26] A. Schneider, G. Hommel and M. Blettner, "Linear Regression Analysis," *Deutsches Ärzteblatt international,* November 2010.

[27] L. Prechelt, "Early Stopping — But When?," in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2012, p. 53–67.

[28] L. Prechelt, "Early Stopping - But When?," in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 1998, p. 55–69.

[29] V. S. Pagolu, K. N. R. Challa, G. Panda and B. Majhi, "Sentiment Analysis of Twitter Data for Predicting Stock Market Movements," October 2016.

[30] V. S. Pagolu, K. N. Reddy, G. Panda and B. Majhi, "Sentiment analysis of Twitter data for predicting stock market movements," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 2016.

[31] D. W. Otter, J. R. Medina and J. K. Kalita, "A Survey of the Usages of Deep Learning in Natural Language Processing," July 2018.

[32] A. I. A. Osman, A. N. Ahmed, M. F. Chow, Y. F. Huang and A. El-Shafie, "Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia," *Ain Shams Engineering Journal,* vol. 12, p. 1545–1556, June 2021.

[33] C. Nwankpa, W. Ijomah, A. Gachagan and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," November 2018.

[34] L. L. Nathans, F. L. Oswald and K. Nimon, "Interpreting Multiple Linear Regression: A Guidebook of Variable Importance," 2012.

[35] A. Mittal, "Stock Prediction Using Twitter Sentiment Analysis," 2011.

[36] M. Makrehchi, S. Shah and W. Liao, "Stock Prediction Using Event-Based Sentiment Analysis," in *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, 2013.

[37] R. Karlemstrand and E. Leckström, "Using Twitter Attribute Information to Predict Stock Prices," May 2021.

[38] C. J. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," 2015.

[39] F. A. Gers, J. Schmidhuber and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," *Neural Computation*, vol. 12, p. 2451–2471, October 2000.

[40] S. F. Crone, J. Guajardo and R. Weber, "A study on the ability of Support Vector Regression and Neural Networks to Forecast Basic Time Series Patterns," in *Artificial Intelligence in Theory and Practice*, Boston, 2006.

[41] B. Chen, P. Lin, Y. Lai, S. Cheng, Z. Chen and L. Wu, "Very-Short-Term Power Prediction for PV Power Plants Using a Simple and Effective RCC-LSTM Model Based on Short Term Multivariate Historical Datasets," *Electronics*, vol. 9, p. 289, February 2020.

[42] R. Caruana, S. Lawrence and C. Giles, "Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping," in *Advances in Neural Information Processing Systems*, 2000.

[43] A. Amin, I. Hossain, A. Akther and K. M. Alam, "Bengali VADER: A Sentiment Analysis

Approach Using Modified VADER," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2019.