



Πανεπιστήμιο Μακεδονίας

Πρόβλεψη τιμών κρυπτονομισμάτων με μεθόδους μηχανικής μάθησης και νευρωνικών δικτύων

Μεταπτυχιακή Διπλωματική εργασία

Δ.Π.Μ.Σ. στη Διοίκηση Επιχειρήσεων (MBA)

Καλαφάτης Δημήτριος

Επιβλέπων: Ευστράτιος Λιβάνης

Θεσσαλονίκη, 2024

Πίνακας περιεχομένων

1. Εισαγωγή [2,7,8,9,12].....	6
1.1. Τι είναι τα κρυπτονομίσματα;.....	6
1.2. Πρόβλεψη τιμών	6
1.3. Η λειτουργία.....	7
1.4. Παραδείγματα κρυπτονομισμάτων	7
1.4.1. Bitcoin.....	7
1.4.2. Ethereum	7
1.4.3. Litecoin	7
1.4.4. Ripple.....	8
1.4.5. Dogecoin.....	8
1.5. Πιθανές αγορές με κρυπτονομίσματα σήμερα.....	8
2. Κρυπτονομίσματα [2,7,8,9,12].....	10
2.1. Πως παράγονται τα κρυπτονομίσματα.....	10
2.2. Blockchain & Hash	12
2.3. Πλεονεκτήματα και κίνδυνοι	12
2.4. Trading και Staking.....	14
3. Χρονολογικές σειρές [6,14,17].....	15
3.1. Χαρακτηριστικά.....	15
4. Τεχνητά νευρωνικά δίκτυα & βαθιά μάθηση [3,10,11,13,14,15,16].....	16
4.1.1. Νευρωνικά δίκτυα και κλασικός προγραμματισμός.....	16
4.1.2. Εφαρμογές τεχνητών δικτύων	17
4.1.3. Διαδικασία μάθησης.....	17
4.2. Είδη τεχνητών νευρικών δικτύων	18
4.2.1. Εμπρόσθιας τροφοδοσίας ενός επιπέδου	18
4.2.2. Εμπρόσθιας τροφοδότησης πολλαπλών επιπέδων.....	19
4.2.3. Ανατροφοδοτούμενα	19
4.5. Εκπαίδευση	20
4.6. Βαθιά μάθηση (Deep learning)	20
4.6.1. LSTM Δίκτυο (Long Short-Term Memory).....	21
4.6.2. Μέθοδος ADAM	23
5. Εφαρμογή της διαδικασίας και κώδικας Python [1,3,4,5,6,10,11,13,15,16].....	24
5.1. Περιγραφή του προβλήματος και μέσα υλοποίησης	24

5.1.1.	Συλλογή δεδομένων.....	24
5.2.	Ανάλυση και επεξήγηση του κώδικα	24
5.2.1.	Εισαγωγή.....	25
5.2.2.	Εισαγωγή των δεδομένων.....	26
5.2.3.	Έλεγχος για μηδενικές ή μη υπάρχουσες τιμές στο αρχείο.....	28
5.2.4.	Ερευνητική ανάλυση δεδομένων.....	28
5.2.5.	Ανάλυση Bitcoin 2014-2024	34
5.2.6.	Δημιουργία μοντέλου πρόβλεψης LSTM	36
5.2.6.1.	Δεδομένα.....	36
5.2.6.2.	Κανονικοποίηση των δεδομένων και χωρισμός	40
5.2.6.3.	Διαμόρφωση του μοντέλου	42
5.2.6.4.	Αποτελέσματα	46
6.	Συμπεράσματα	48
7.	Παράρτημα.....	49
8.	Βιβλιογραφία.....	60

Περίληψη

Αντικείμενο της παρούσας μεταπτυχιακής διπλωματικής εργασίας αποτελεί η μελέτη των κρυπτονομισμάτων, η ανάλυση των νευρικών δικτύων και της βαθιάς μάθησης, οι χρονολογικές σειρές, η πρόβλεψή τους και η αξιολόγηση αυτών.

Στη σημερινή εποχή, η πρόβλεψη τιμών και συμπεριφορών κρίνεται αναγκαία, αφού προσφέρει γνώση που δεν προϋπάρχει και γεννά την δυνατότητα της πρώιμης ενέργειας και της προετοιμασίας. Μέσω της ραγδαίας τεχνολογικής εξέλιξης η οποία παρατηρείται τα τελευταία χρόνια, οι μέθοδοι μηχανικής μάθησης επιτρέπουν πλέον μια ικανοποιητική πρόβλεψη μελλοντικών χρονοσειρών.

Σκοπός αυτής της εργασίας αποτελεί η ενημέρωση και η εξοικείωση περί της νέας οικονομίας και τα χαρακτηριστικά των κρυπτονομισμάτων, της βαθιάς μάθησης, των χρονοσειρών και η πρόβλεψη μελλοντικών τιμών αυτών μέσω μοντέλων στη γλώσσα προγραμματισμού Python.

Το πρώτο μέρος της εργασίας περιλαμβάνει το απαραίτητο θεωρητικό υπόβαθρο για τα κρυπτονομίσματα, όπως την λειτουργία τους και την δημιουργία τους. Ακόμα περιέχει πληροφορίες για τις χρονοσειρές, την βαθιά μάθηση και τα τεχνητά νευρωνικά δίκτυα.

Το δεύτερο μέρος της εργασίας περιέχει το πρακτικό και πειραματικό κομμάτι, στο οποίο θα επιδιωχθεί και θα αξιολογηθεί η πρόβλεψη μελλοντικών τιμών του κρυπτονομίσματος Bitcoin, όντας το πιο διαδομένο και άμεσα αξιοποιήσιμο στην αγορά. Αυτή η διαδικασία περιλαμβάνει την οικονομική ανάλυση του Bitcoin ανά τα χρόνια, καθώς και την εκπαίδευση ενός νευρωνικού δικτύου μακροπρόθεσμης μνήμης LSTM. Τέλος, τα αποτελέσματα τα οποία θα προκύψουν θα σχολιαστούν ποιοτικά καθώς και με μετρητικούς δείκτες όπως το μέσο τετραγωνικό σφάλμα, η ρίζα του μέσου τετραγωνικού σφάλματος, ενώ παράλληλα θα γίνει και η αξιολόγηση της χρηστικότητας ενός τέτοιου μοντέλου τη σημερινή εποχή.

Abstract

The subject of this postgraduate thesis is the technical study of cryptocurrencies, the analysis of artificial neural networks and deep learning, the definition of time lines and the possibility for the prediction of future values of them, as well as the evaluation of the whole practice.

Nowadays, the prediction of a large variety of values and behaviors is considered critical, as it delivers knowledge which was not available beforehand as well as the ability to prepare, adapt and take action. Due to the rapid growth of science and technology the past few years, machine and deep learning methods have seen a rise in popularity, as they deliver the ability to predict future values in time series.

The purpose of this thesis is to educate and familiarize the reader about the new economy surrounding cryptocurrencies, their attributes, about machine learning and neural networks, in addition to experiment with the possibility to predict future values of time series using the Python programming language.

The first part of the thesis consists of all the necessary information about crypto, like their functions and the procedures behind their crafting. Also, machine learning, neural networks and time series will be extensively analyzed.

The second part consists of the practical share, in which the prediction of future values of Bitcoin will be generated and assessed. This procedure consists of the exploratory economic analysis and the training of an artificial long short-term memory artificial neural network. Lastly, the outcome of this experiment will be evaluated based on certain performance metrics such as mean squared error and rooted mean squared error, along with the implantation of such a model in today's society.

1. Εισαγωγή [2,7,8,9,12]

1.1. Τι είναι τα κρυπτονομίσματα;

Τα κρυπτονομίσματα αποτελούν ένα κρυπτογραφημένο ηλεκτρονικό εργαλείο ανταλλαγής το οποίο αξιοποιεί συναρτήσεις για την εκτέλεση και διεκπεραίωση χρηματοοικονομικών συναλλαγών. Αυτές οι συναλλαγές δεν υπόκεινται σε κάποια ρυθμιστική αρχή αλλά αξιοποιούν ένα αποκεντρωμένο σύστημα για την καταγραφή των συναλλαγών και την δημιουργία νέων. Αποτελεί μια ψηφιακή μορφή συστήματος πληρωμών η οποία δεν εξαρτάται από τράπεζες για την πιστοποίηση των συναλλαγών αλλά σε ένα σύστημα Peer to peer, το οποίο επιτρέπει στον καθένα οπουδήποτε βρίσκεται γεωγραφικά να λάβει και στείλει πληρωμές. Τέτοιου είδους πληρωμές αντικαθιστούν τον πατροπαράδοτο τρόπο των φυσικών χαρτονομισμάτων και πορτοφολιών και εισάγουν μια νέα πρακτική κατά την οποία οι πληρωμές υπάρχουν αποκλειστικά ως ψηφιακές εντολές σε διαδικτυακές βάσεις δεδομένων, όπου και περιγράφεται αναλυτικά η συναλλαγή. Όλες οι συναλλαγές καταγράφονται σε "ledger", τα οποία μπορούν να παρομοιαστούν με ένα δημόσιο λογιστικό βιβλίο στο οποίο όλοι έχουν πρόσβαση, ενώ τα ίδια τα κρυπτονομίσματα αποθηκεύονται σε ψηφιακά πορτοφόλια.

Η ονομασία δόθηκε στα νομίσματα αυτά λόγω της κρυπτογράφησης (encrytion) που αυτά προσφέρουν κατά την διεκπεραίωση των συναλλαγών. Αυτό σημαίνει πως από πίσω υπάρχει προηγμένη κωδικοποίηση και προγραμματισμός που συμβαίνει κατά τη ανταλλαγή δεδομένων μεταξύ των ψηφιακών πορτοφολιών και των λογιστικών βιβλίων. Ο απώτερος στόχος όλου αυτού του κινήματος είναι η προσφορά ανώτερης ασφάλειας.

Το πρώτο κρυπτονόμισμα αποτέλεσε το Bitcoin, το οποίο εμφανίστηκε για πρώτη φορά το 2009 και ακόμα και σήμερα αποτελεί το πιο εξεζητημένο από όλα τα υπόλοιπα.

1.2. Πρόβλεψη τιμών

Σύνηθες μορφή επένδυσης στις μέρες μας αποτελεί η αγορά κρυπτονομισμάτων, η βραχυπρόθεσμη ή μακροπρόθεσμη κατοχή τους και η πώληση τους σε υψηλότερη αξία από αυτή της αγοράς για προσωπικό όφελος του χρήστη. Μια τέτοια επένδυση μπορεί να προσφέρει από μικρά έως τεράστια κέρδη, τα οποία μέχρι σήμερα ήταν πρωτόγνωρα να συμβούν από διαφόρου τύπου επενδύσεις.

1.3. Η λειτουργία

Τα κρυπτονομίσματα λειτουργούν πάνω σε ένα άμεσα προσβάσιμο και δημόσιο λογιστικό βιβλίο που ονομάζεται blockchain, το οποίο όπως αναφέρθηκε κρατάει το ιστορικό των συναλλαγών που μεσολαβούν ανάμεσα στους χρήστες. Οι διάφορες μονάδες των κρυπτονομισμάτων δημιουργούνται από διαδικασία η οποία ονομάζεται mining, κατά την οποία ένας υπολογιστής αξιοποιείται για την επίλυση περίπλοκων μαθηματικών προβλημάτων που δημιουργούν τα νομίσματα. Οι χρήστες μπορούν επίσης να προμηθευτούν νομίσματα από διάφορους εξειδικευμένους διαδικτυακούς ιστοτόπους, να αποθηκεύσουν και να καταναλώσουν τα νομίσματα αυτά μέσω ειδικών ηλεκτρονικών πορτοφολιών.

Η ιδιοκτησία κρυπτονομισμάτων δεν αναφέρεται σε κάτι απτό, αλλά αντιθέτως σε κάτι το οποίο υπάρχει μόνο στη θεωρία. Στην πραγματικότητα ο κάθε χρήστης έχει στη κατοχή του ένα κλειδί το οποίο του επιτρέπει να μεταφέρει μια καταγραφή ή ένα στοιχείο μέτρησης από τον έναν χρήστη στον άλλον, χωρίς να υπάρχει επίβλεψη ή ανάγκη για μια επιπλέον ρυθμιστική αρχή.

1.4. Παραδείγματα κρυπτονομισμάτων

1.4.1. Bitcoin

Εμφανίστηκε πρώτη φορά το 2009 και αποτέλεσε την αρχή της γένεσης των κρυπτονομισμάτων, ενώ πλέον αποτελεί το νόμισμα με την περισσότερη κινητικότητα. Το νόμισμα δημιουργήθηκε από τον Satoshi Nakamoto, το οποίο πολλοί θεωρούν ένα ψευδώνυμο και πως η πραγματική ταυτότητα του ιδρυτή παραμένει ακόμα άγνωστη. Πλέον αποτελεί τον κολοσσό του κλάδου και εργαλείο για την λειτουργία όλου του υπολοίπου συστήματος.

1.4.2. Ethereum

Η αρχή του δεύτερου μεγαλύτερου κρυπτονομίσματος έγινε το 2015, κατά το οποίο το νόμισμα αυτό δημιούργησε μια νέα δικιά του οικονομία. Πλέον αποτελεί και αυτό βασικό εργαλείο για την αγορά κρυπτονομισμάτων, την ανταλλαγή τους, ακόμα και για φυσικές αγορές.

1.4.3. Litecoin

Αυτό το κρυπτονόμισμα βασίστηκε και έχει αρκετές ομοιότητες με το Bitcoin αλλά έχει πιο ραγδαία ανάπτυξη και θεωρείται πιο πρωτοπόρο. Χαρακτηρίζεται από γρηγορότερες συναλλαγές και πληρωμές, με στόχο την αύξηση των επιτρεπόμενων συναλλαγών.

1.4.4. Ripple

Το Ripple αποτελεί ένα διαμοιρασμένο «λογιστικό βιβλίο» το οποίο πρωτοεμφανίστηκε το 2012. Η χρήση του αποτελεί την καταγραφή διαφόρων συναλλαγών οι οποίες δεν έχουν απαραίτητα σχέση με κρυπτονομίσματα. Η εταιρία που βρίσκεται πίσω από το νόμισμα έχει συνάψει συνεργασίες με αρκετές τράπεζες και διάφορους οικονομικούς οργανισμούς.

1.4.5. Dogecoin

Αποτελεί ένα νόμισμα το οποίο εμφανίστηκε για πρώτη φορά το 2013 και δημιουργήθηκε εξ αρχής ως ένα αστείο και πολλές φορές χρησιμοποιούταν ως μορφή φιλοδωρήματος. Απέκτησε μεγάλη φήμη και έγινε διάσημο μέσω διασημοτήτων που ασχολήθηκαν με αυτό ενώ επέτρεψε σε αρκετούς πρόωρους επενδυτές να έχουν αμύθητα κέρδη από ένα νόμισμα το οποίο δεν έχει κάποιο ουσιαστική λειτουργία και σκοπό.

1.5. Πιθανές αγορές με κρυπτονομίσματα σήμερα

Με την πρώτο λανσάρισμα του Bitcoin, ο σκοπός αυτού αποτελούσε να εξελιχθεί σε μια χρηματική οικονομία ευρέως διαδεδομένη όπου ο καθένας θα μπορούσε να αγοράσει με αυτά από ένα καφέ ή έναν υπολογιστή μέχρι και εκτάσεις γης ή ένα σπίτι. Αν και σίγουρα υπάρχει πρόοδος από το 2015 που εμφανίστηκε το πρώτο κρυπτονόμισμα, σήμερα το 2022, δεν έχει εκπληρωθεί πλήρως αυτό το όραμα. Αν και όλο και περισσότερες εταιρίες και οργανισμοί γίνονται αποδέκτες των νομισμάτων για παροχή υπηρεσιών ή την αγορά αγαθών, η ανταλλαγή τους για αντικείμενα μεγάλης αξίας είναι αρκετά σπάνια. Παρακάτω αναφέρονται κάποια παραδείγματα ενδεικτικά:

- Εταιρείες εμπορίου ηλεκτρονικών ειδών

Διάφοροι κολοσσοί πώλησης ηλεκτρονικών ειδών και υπηρεσιών άρχισαν να δέχονται από νωρίς τα κρυπτονομίσματα όπως η Microsoft, η AT&T η οποία αποτελεί εταιρία κινητής τηλεφωνίας στην Αμερική και το Newegg, το οποίο αποτελεί έμπορο πάσης φύσεως ηλεκτρονικών συσκευών. Την αρχή έκαναν η εταιρείες Overstock, Shopify και Rakuten.

- Πολυτελή αγαθά

Διάφορες πλατφόρμες οι οποίες κατέχουν αγαθά υψηλής αξίας και συγκεκριμένα ρολογιών συνηθίζουν να διαθέτουν πληρωμή με κρυπτονομίσματα, αφενός λόγω της αμεσότητας της συναλλαγής και την μη ύπαρξη ορίου όπως θέτουν αρκετές τράπεζες, αφετέρου διότι τα οι συναλλαγές με κρυπτονομίσματα μπορούν να προσφέρουν την ανωνυμία σε υψηλού προφίλ πελάτες.

- Αυτοκίνητα

Διαδομένη μορφή αγοράς αυτοκινήτων αποτελούν τα κρυπτονομίσματα, κυρίως στην Αμερική, τα οποία εκτείνονται από απλά προσιτά αυτοκίνητα μέχρι αγωνιστικά και πολυτελείας.

- Ασφάλεια

Μία πρόσφατη πρακτική η οποία εφαρμόστηκε από την ασφαλιστική AXA είναι η πληρωμή διαφόρων ασφαλειών, εκτός της ασφάλειας ζωής λόγω ορισμένων νομοθετικών όρων στην Αμερική, με κρυπτονομίσματα.

- Υπηρεσίες

Σύνηθες φαινόμενο αποτελεί πλέον η παροχή υπηρεσιών από τρίτους είτε από εταιρίες με το αντίτιμο να μπορεί να καταβληθεί σε κρυπτονομίσματα. Συνήθως, τα κρυπτονομίσματα που χρησιμοποιούνται είναι μικρής σχετικά αξίας, ενώ για πολλά χρόνια υπήρχε και η κουλτούρα του φιλοδωρήματος σε κάποιον ο οποίος παρείχε μια υπηρεσία ή μια διευκόλυνση, συνήθως με την χρήση του κρυπτονομίσματος Dogecoin.

Αξίζει να σημειωθεί πως πλέον υπάρχουν διάφοροι φορείς όπως η Bitpay, η οποία επιτρέπει στον χρήστη να εκδώσει μια χρεωστική κάρτα συνδεδεμένη με το ηλεκτρονικό του πορτοφόλι, η οποία μπορεί να χρησιμοποιηθεί σε εμπόρους και υπηρεσίες ακόμα και εάν αυτή δεν δέχονται απευθείας την πληρωμή με κρυπτονομίσματα, αφού γίνεται αυτόματα η συναλλαγή στο εκάστοτε νόμισμα με την ανάλογη μικρή προμήθεια της εταιρίας για τις υπηρεσίες που προσφέρει.

2. Κρυπτονομίσματα [2,7,8,9,12]

Όπως αναφέρθηκε και στην αρχή τα κρυπτονομίσματα αποτελούν μια εικονική μη απτή οικονομία οι οποία βασίζεται πάνω στην κρυπτογράφηση και στην επίλυση μαθητικών συναρτήσεων για διάφορες χρηματοοικονομικές συναλλαγές. Τα περισσότερα από αυτά βασίζονται στην τεχνολογία Blockchain και αποτελεί ένα σύστημα peer-to-peer, όροι που θα αναλυθούν περαιτέρω σε επόμενη ενότητα.

Είδαμε πως το Bitcoin αποτέλεσε το πρώτο κρυπτονομίσμα το 2015, το οποίο ξεκίνησε από μια ασήμαντη τιμή και κατάφερε να αγγίξει τα 68000 δολάρια το Νοέμβριο του 2021. Αποτέλεσε το λόγο που σήμερα το 2022 υπάρχουν παραπάνω από 2000 κρυπτονομίσματα στην αγορά.

Παρακάτω θα γίνει παράθεση ορισμένων όρων που θα χρησιμοποιηθούν για την καλύτερη κατανόηση του αναγνώστη.

- Peer-to-peer network: αποτελεί ένα δίκτυο υπολογιστών στο οποίο το κάθε μηχάνημα διακατέχει την ίδια αξία στην λειτουργία του συστήματος.
- Public key: ένας συνδυασμός γραμμάτων και αριθμών που προσδιορίζει την ηλεκτρονική διεύθυνση ενός χρήστη και αξιοποιείται για την διενέργεια συναλλαγών.
- Private key: Αποτελεί ίδια μορφή με το δημόσιο κλειδί, όμως είναι γνωστό μόνο στον εκάστοτε χρήστη και ουσιαστικά φέρει την ηλεκτρονική υπογραφή του
- Wallet: Μαζί με το ηλεκτρονικό κλειδί αποτελεί ουσιαστικά ένα ηλεκτρονικό πορτοφόλι όπου ο χρήστης αποθηκεύει τα κρυπτονομίσματά του

2.1. Πως παράγονται τα κρυπτονομίσματα

Υπάρχουν 4 τρόποι με τους οποίους παράγονται τα κρυπτονομίσματα:

- Mining

Το Bitcoin και πολλά άλλα κρυπτονομίσματα αποτελούν αποκεντρωμένα συστήματα τα οποία με τη βοήθεια υπολογιστών από όλο το κόσμο επιβεβαιώνουν τις συναλλαγές που διεξάγονται σε παγκόσμια κλίμακα με την επίλυση περίπλοκων μαθηματικών εξισώσεων. Αυτό αποτελεί διαδικασία η οποία απαιτεί μεγάλη υπολογιστική ισχύ. Όλοι οι υπολογιστές συνεισφέρουν στην επίλυση αυτών των προβλημάτων και στο τέλος ενός προκαθορισμένου αριθμού λεπτών (πχ. 10 λεπτά για το Bitcoin), μοιράζονται κρυπτονομίσματα στην ομάδα που βοήθησε στη διαδικασία, ανάλογα με το ποσοστό το οποίο συνεισφέρει η κάθε υπολογιστική μονάδα. Αυτές οι ομάδες ονομάζονται pools, και βοηθάνε τους χρήστες να επιλύουν πιο γρήγορα εξισώσεις και με αυτόν τον τρόπο να λαμβάνουν μεγαλύτερη ανταμοιβή.

Στη θεωρία ο καθένας με έναν απλό υπολογιστή μπορεί να κάνει mining κρυπτονομισμάτων, όμως με τη πάροδο του χρόνου οι συναρτήσεις και τα προβλήματα γίνονται όλο και πιο περίπλοκα και απαιτούν μεγαλύτερη υπολογιστική ισχύ. Για να έχει κέρδος κάποιος από

αυτήν την διαδικασία απαιτείτε μια ισορροπία ανάμεσα στο hardware, την σωστή ρύθμιση αυτού και την εκάστοτε κατανάλωση ρεύματος που χρησιμοποιεί το μηχάνημα. Η αρχή έγινε με ηλεκτρονικούς υπολογιστές ενώ με την πάροδο των χρόνων εμφανίστηκαν και ειδικές συσκευές οι οποίες ήταν κατάλληλες και πιο αποδοτικές μόνο για αυτήν την συγκεκριμένη διαδικασία.

- Αγορά

Πλέον υπάρχει μεγάλος αριθμός ηλεκτρονικών εμπόρων-ανταλλακτηρίων όπου ο χρήστης μπορεί να αγοράσει κρυπτονομίσματα κάθε φορά με την αντίστοιχη ισοτιμία, πληρώνοντας ένα μικρό φόρο στον έμπορο για την υπηρεσία που του προσφέρει. Συνήθως, τέτοιες πρακτικές υλοποιούνται με στόχο την βραχυπρόθεσμη ή μακροπρόθεσμη κατοχή ενός κρυπτονομίσματος από τον χρήστη και την μετέπειτα μεταπώληση του σε υψηλότερη αξία για προσωπικό του όφελος.

- Διανομή

Μία ακόμα περίπτωση παραγωγής κρυπτονομισμάτων είναι η δωρεάν διανομή τους από τον οργανισμό που τα δημιουργεί και βρίσκεται πίσω από αυτά. Συνήθως αποτελούν κρυπτονομίσματα μικρότερης αξίας τα οποία πολλές φορές δεν έχουν τις δυνατότητες που θα τα κάνουν ελκυστικά.

- Κατοχή

Αν και δεν αποτελεί σύνηθες φαινόμενο, υπάρχουν κρυπτονομίσματα τα οποία η παραγωγή του έγινε μέσω mining είτε μέσω διανομής και οργανισμός που βρίσκεται πίσω από το νόμισμα αποφάσισε να βάλει ένα όριο στον αριθμό των παραχθέντων νομισμάτων. Από εκεί και πέρα, ο οργανισμός επιβραβεύει στην ουσία τους κατόχους των νομισμάτων του με νέα νομίσματα ανάλογα με τη ποσότητα την οποία ήδη κατέχει ο εκάστοτε χρήστης.

2.2. Blockchain & Hash

Για να καταλάβει κάποιος πως λειτουργεί ένα σύστημα κρυπτονομισμάτων όπως το Bitcoin, αρκεί να φανταστεί μια αλυσίδα από κύβους η οποία διαρκώς επεκτείνεται. Κάθε ένας κύβος η αλλιώς Block, εκτός από τα δεδομένα που αυτός έχει, κατέχει και έναν δείκτη που ονομάζεται Hash, ο οποίος παραπέμπει στο προηγούμενο κύβο της αλυσίδας. Η ιδιότητα που έχει ένας δείκτης Hash, είναι ότι περιέχει όλη τη πληροφορία του προηγούμενου κύβου και έτσι δημιουργείται ένα αξιόπιστο σύστημα. Όταν έρθει η ώρα για την διεκπεραίωση μιας συναλλαγής, όλες οι παράμετροι οι οποίες είναι σημαντικές, όπως ο χρόνος της συναλλαγής, το μέγεθος της και τα συμβαλλόμενα μέρη, παράγουν ένα hash value που παριστάνει το transaction ID.

Για να γεννηθεί ένα Blockchain χρειαζόμαστε το πρώτο κύβο, γνωστό και ως Genesis Block το οποίο από μόνο του περιλαμβάνει ένα μοναδικό hash value. Για να συνεχίσει η αλυσίδα αυτή, όταν δημιουργηθεί ο δεύτερος κύβος, μαζί με τις πληροφορίες από τις εκάστοτε συναλλαγές θα περιέχει και το hash value του Genesis Block αρχίζοντας έτσι την αλυσίδα. Στην ίδια λογική, κάθε επόμενος κύβος θα περιέχει τις πληροφορίες του, μαζί με ένα hash το οποίο θα περιέχει όλες τις πληροφορίες του προηγούμενου κύβου.

2.3. Πλεονεκτήματα και κίνδυνοι

Όπως κάθε νέα πρόοδος της τεχνολογίας και γενικότερα κάθε νέα ανακάλυψη, τα κρυπτονομίσματα αποτελούν μια νέα εισαγωγή στην οικονομία και τις οικονομικές συναλλαγές με πολλαπλά οφέλη, αλλά και ταυτόχρονα πολλούς κινδύνους.

Μερικά από τα πλεονεκτήματα των δημοφιλέστερων κρυπτονομισμάτων όπως το Bitcoin ή το Ethereum, αλλά και των υπολοίπων τα οποία είναι ευρέως αποδεκτά από το κοινό και έχουν μεγάλο εύρος αξιοποίησης, παρουσιάζονται αναλυτικά παρακάτω:

- Αντίγραφα ασφαλείας: Κάθε χρήστης αποθηκεύει τα κρυπτονομισμάτα του σε εικονικά πορτοφόλια τα οποία κατοχυρώνονται και με σωστή πρακτική μπορούν να γίνουν απόρθητα. Ο χρήστης έχει δυνατότητα εκτός του κλασσικού κωδικού να εισάγει επιπλέον μέτρα ασφαλείας, όπως μοναδικούς κωδικούς από γεννήτριες, οι οποίες αλλάζουν ανά μερικά δευτερόλεπτα, κωδικούς βασισμένους σε βιομετρικά χαρακτηριστικά και άλλα.
- Διαφάνεια Συναλλαγών: Ένα από το μεγαλύτερα στοιχεία που εδραιώνουν τη παρουσία των κρυπτονομισμάτων στην παγκόσμια σκηνή είναι η διαφάνεια που αυτά προσφέρουν. Κάθε συναλλαγή η οποία εκτελείται ανά το κόσμο, ουσιαστικά αποθηκεύεται σε ένα αρχείο το οποίο είναι προσβάσιμο οποτεδήποτε, από οπουδήποτε, από τον οποιοδήποτε. Επίσης κάθε συναλλαγή η οποία γίνεται συνοδεύεται από το μέρη που ενεπλάκησαν, το ποσό, την τοποθεσία και την ώρα.
- Συμμετοχή της κοινότητας: Τα κρυπτονομίσματα αποτελούν μια οικονομία την οποία δρομολογούν όλοι του οι χρήστες. Οποιαδήποτε αλλαγή στους κανόνες ή στην λειτουργία γίνεται μόνο και εφόσον γίνει αποδεκτή από το κοινό.

- Αποκέντρωση: Ίσως από τα πιο σημαντικά προσόντα για ορισμένους, αφού τα κρυπτονομίσματα δεν απαιτούν κάποια κεντρική αρχή ελέγχου όπως μια τράπεζα ή έναν οργανισμό. Η μόνη περίπτωση να σταματήσουν να δουλεύει αυτό το τεράστιο δίκτυο είναι ο ταυτόχρονος τερματισμός όλων των υπολογιστών που συμμετέχουν σε αυτό ταυτόχρονα, πράγμα το οποίο καθίσταται αδύνατο.
- Ταχύτητα: Κάθε συναλλαγή με κρυπτονομίσμα είναι στιγμιαία ενώ παράλληλα καταγράφεται στο δημόσια προσβάσιμο αρχείο συναλλαγών. Δεν υπάρχει απαίτηση για ειδικά συστήματα και μηχανήματα.
- Κόστος συναλλαγών: Κάθε συναλλαγή έχει ένα πολύ μικρό κόστος, το οποίο διαμοιράζεται στους χρήστες οι οποίοι έλαβαν μέρος στην εκτέλεση αυτής της συναλλαγής.
- Ιδιωτικότητα: Κάθε χρήστης είναι ικανός να δημιουργήσει αμέτρητες διευθύνσεις από τις οποίες θα εκτελεί τις συναλλαγές του, προσόν το οποίο αρκετοί επιζητούν για να κρατήσουν την ανωνυμία τους. Αν και όλες οι συναλλαγές δημοσιεύονται και είναι άμεσα προσβάσιμες, πολλές φορές αυτό το χαρακτηριστικό αξιοποιείται και για την διενέργεια παρανομιών.

Εκτός από τα πλεονεκτήματα όμως κάθε νέα ανακάλυψη γεννά κινδύνους, οι οποίοι όμως με τη πάροδο του χρόνου είναι πιθανό να εξαλειφθούν.

- Απώλεια κλειδιών και κωδικών πρόσβασης: Εάν και ο χρήστης μπορεί να λάβει ένα μεγάλο αριθμό περιοριστικών μέτρων ώστε να αποτρέψει την πρόσβαση στα κρυπτονομίσματα του σε τρίτους, όταν κάποιος το αμελήσει ή όταν κάποιος τρίτος καταφέρει να αποκτήσει πρόσβαση σε αυτά, ουσιαστικά ο χρήστης χάνει τα πάντα. Αυτός που θα παραβιάσει τον χρήστη αυτομάτως αποκτά πρόσβαση σε όλα τα κρυπτονομίσματα του, με τη δυνατότητα να τα μεταφέρει όλα σε δικούς του λογαριασμούς, πιθανώς χωρίς εμφανή στοιχεία για την πορεία τους και χωρίς τη δυνατότητα επιστροφής.
- Νομικό πλαίσιο: Η αποκεντρωτική φύση των κρυπτονομισμάτων, μαζί με τις νέες παραμέτρους που εισάγουν, έχουν δημιουργήσει ένα κενό στο νομοθετικό πλαίσιο γύρω από αυτά. Αν και νομοθεσίες έχουν θεσπιστεί για να ορίσουν κάποια όρια, το τοπίο ακόμα θεωρείται θολό.
- Ασφάλεια δικτύου: Κατά καιρούς το γενικό σύστημα έχει υποστεί βλάβες και ανάλογα μια προβληματική λειτουργία, περιστάσεις όμως οι οποίες εύκολα διορθώθηκαν σε σύντομο χρονικό διάστημα, χωρίς να υπάρχει κάποια ουσιαστική επίπτωση. Πάραυτα, ο κίνδυνος για επιθέσεις ή κακόβουλες προθέσεις ως προς το γενικότερο δίκτυο, παραμένει ανοιχτό.

2.4. Trading και Staking

Ένας πολύ σημαντικός παράγοντας ο οποίος έπαιξε ρόλο στην διάδοση των κρυπτονομισμάτων είναι οι ιστορίες από ανθρώπους που πλούτισαν με αμύθητα ποσά, πολλές φορές και σε πολύ σύντομα χρονικά διαστήματα, όντως πρόωροι επενδυτές σε αυτά είτε αξιοποιώντας μεθόδους εξόρυξης με ειδικά σχεδιασμένα μηχανήματα και τις λεγόμενες φάρμες κρυπτονομισμάτων. Πλέον θεωρείται έως και κανονικό επάγγελμα η πώληση και αγορά κρυπτονομισμάτων σε μικρά χρονικά διαστήματα, κάτι ως επένδυση, όπως θα λειτουργούσε και η αγορά μιας μετοχής. Είναι ένας από τους λόγους που η κατάσταση έχει τόση βαρύτητα και πλέον πολλοί αναζητούν τρόπους για να έχουν κέρδος από αυτή τη διαδικασία. Παρακάτω θα παρουσιαστούν οι 2 δημοφιλέστεροι τρόποι με τους οποίους κάποιος μπορεί να επενδύσει και να έχει κέρδος από τα κρυπτονομίσματα:

- Trading

Με απλά λόγια είναι ένα παιχνίδι προσφοράς και ζήτησης. Στόχος είναι η αγορά τη σωστή στιγμή, όταν για παράδειγμα ένα κρυπτονόμισμα βγαίνει στην αγορά και δεν έχει ιδιαίτερη αξία, καθώς και η πώληση στη σωστή στιγμή, όταν για παράδειγμα ο οργανισμός πίσω από το κρυπτονόμισμα συνάπτει κάποια συμφωνία με μια μεγάλη εταιρία και η τιμή του ανεβαίνει. Τα κρυπτονομίσματα δηλαδή λειτουργούν με τον ίδιο τρόπο με τον οποίο λειτουργούν μετοχές στο χρηματιστήριο. Αποτελεί μια διαδικασία χρονοβόρα, με ανάγκη για πολύ καλή αντίληψη, γνώση, ενημέρωση. Κατά κύριο λόγο υπάρχουν δύο περιπτώσεις trading. Η πρώτη αφορά γρήγορες αγοραπωλησίες με στόχο το μικρό κέρδος από πολλαπλές περιπτώσεις (quick sell/buy), ενώ η δεύτερη αφορά τη μακροπρόθεσμη κατοχή με στόχο τον πολλαπλασιασμό της αξίας (hold).

- Staking

Το staking αναφέρεται στη διατήρηση της αξίας των χρημάτων που προέρχονται από κρυπτονομίσματα που βρίσκονται σε ένα ψηφιακό πορτοφόλι για την υποστήριξη της ασφάλειας και των λειτουργιών ενός δικτύου Blockchain. Ουσιαστικά, αποτελεί μια εναλλακτική του Mining, η οποία δεν απαιτεί υπολογιστική ισχύ παρά μόνο κεφάλαιο, αφού ο χρήστης με την προσφορά του και το κλείδωμα των κρυπτονομισμάτων, λαμβάνει χρηματικές ανταμοιβές για την διατήρηση της σωστής λειτουργίας όλου του δικτύου. Στην πράξη λειτουργεί όπως η κατάθεση ενός κεφαλαίου σε ένα λογαριασμό τραπεζής, το οποίο με την πάροδο του χρόνου δανείζεται θεωρητικά η τράπεζα για τις λειτουργίες και τις συναλλαγές της και επιβραβεύει τον ιδιοκτήτη με ένα επιτόκιο στο κεφάλαιο του, ή ένα ομόλογο το οποίο αποκαθιστά το κεφάλαιο επένδυσης συν τους τόκους με αντάλλαγμα τα δικαιώματα χρήσης των κεφαλαίων κατά της διάρκειά του.

3. Χρονολογικές σειρές [6,14,17]

Σε περίπτωση που θέλουμε να μελετήσουμε ένα φαινόμενο ή μια τιμή σε βάθος χρόνου, χρησιμοποιούμε ένα σύνολο παρατηρήσεων το οποίο συλλέγουμε και ονομάζουμε χρονολογική σειρά. Οι χρονολογικές σειρές μπορούν να διαχωριστούν με βάση τις μεταβλητές που χαρακτηρίζουν, από μία έως απροσδιόριστο αριθμό, ενώ και από την χρονική στιγμή της παρατήρησης, δηλαδή εάν είναι συνεχής στο φάσμα του χρόνου ή ανά ορισμένα χρονικά διαστήματα. Για να δημιουργήσουμε ένα γράφημα της χρονοσειράς, το λεγόμενο χρονοδιάγραμμα, αρκεί να συμβολίσουμε στον x ένα ορθοκανονικό σύστημα αξόνων την μεταβλητή X ως την στιγμή της χρονικής παρατήρησης και την μεταβλητή Y ως την τιμή αυτής. Με αυτόν τον τρόπο προκύπτει η εικόνα της εξέλιξης της τιμής ή του φαινομένου που μελετάμε.

Για την πρόβλεψη των τιμών μια χρονολογικής σειράς, εκτός από μαθηματικές μεθόδους, μπορούν να αξιοποιηθούν και μέθοδοι μηχανικής μάθησης. Αρχικά θα χρειαστούν δεδομένα μιας περιόδου των τιμών της χρονοσειράς τα οποία ενδεικτικά μπορούν να εκτείνονται από λεπτό έως χρόνο και δεκαετία. Αξιοποιώντας λοιπόν αυτά τα δεδομένα ως μια βάση εκπαίδευσης, το μοντέλο μηχανικής μάθησης μπορεί να προβλέψει τις μελλοντικές τιμές της χρονοσειράς στο σύντομο ή και μακρινό μέλλον.

3.1. Χαρακτηριστικά

Οι χρονολογικές σειρές πολλές φορές έχουν τόση μαζεμένη πληροφορία που δεν αρκεί η απλή εξέταση του χρονοδιαγράμματος τους για να προκύψει κάποιο συμπέρασμα. Υπάρχει λοιπόν η ανάγκη για περαιτέρω εμβάθυνση, για την ανακάλυψη ιδιοτήτων που βοηθήσουν στην ερμηνεία και την επεξεργασία τους. Τα ποιοτικά χαρακτηριστικά μιας χρονολογικής σειράς είναι πέντε και θα αναλυθούν στη συνέχεια.

Η τάση μιας χρονοσειράς, αποτελεί τη γενική εικόνα αυτής και ορίζεται ως « Η μακροχρόνια μεταβολή της μέσης τιμής της παρατηρούμενης μεταβλητής κατά τη διάρκεια μιας συγκεκριμένης χρονικής περιόδου». Αυτή μπορεί να είναι σταθερή, γραμμική ή μη γραμμική.

Όταν σε μια χρονολογική σειρά οι τιμές κυμαίνονται γύρω από μία μέση τιμή και δεν υπάρχει διακύμανση των τιμών σε σχέση με το χρόνο, τότε παρατηρούμε το χαρακτηριστικό της στασιμότητας.

Ένα ακόμα χαρακτηριστικό αποτελεί η εποχικότητα, η οποία αναφέρεται στην περιοδική διακύμανση με σταθερό μήκος μικρότερο του έτους.

Ακόμα, όταν οι παρατηρήσεις της χρονοσειράς έχουν μεγάλη απόκλιση από το μέσο όρο του συνόλου, παρουσιάζεται το χαρακτηριστικό των ακραίων τιμών.

Τέλος, όταν παρουσιάζονται αυξομειώσεις των τιμών της μεταβλητής λόγω εξωτερικών παραγόντων σε μη σταθερές περιόδους, έχουμε το χαρακτηριστικό της κυκλικότητας.

4. Τεχνητά νευρωνικά δίκτυα & βαθιά μάθηση [3,10,11,13,14,15,16]

Η κατανόηση των νευρωνικών δικτύων μπορεί να γίνει εύκολα κατανοητή μέσω της μελέτης λειτουργίας των φυσικών νευρώνων. Ο εγκέφαλος ο οποίος αποτελείται από νευρώνες, διαθέτει μια ικανότητα μάθησης και μνήμης η οποία οφείλεται στην ιδιότητα των συνάψεων να μεταβάλουν την αγωγιμότητά τους. Τα ηλεκτρικά σήματα που εισέρχονται στο σώμα μέσω των δενδριτών συνδυάζονται και αν το αποτέλεσμα ξεπερνάει κάποια τιμή κατωφλίου το σήμα διαδίδεται με τη βοήθεια του άξονα προς άλλους νευρώνες.

Ένα τεχνητό νευρωνικό δίκτυο ορίζεται ως ένας συμπαγής παράλληλος κατανεμημένος επεξεργαστής που έχει τη φυσική κλίση να αποθηκεύει εμπειριστατωμένη γνώση και να την διαθέσιμη προς χρήση. Το κύριο πλεονέκτημά του είναι η ανοχή σε δεδομένα εκπαίδευσης που περιστασιακά έχουν λανθασμένες ή κενές τιμές, ενώ το κύριο μειονέκτημα είναι πως από μόνος του αδυνατεί να εξηγήσει ποιοτικά τη γνώση που μοντελοποιεί.

Τα τεχνητά νευρωνικά δίκτυα σχεδιάστηκαν με βάση τη λειτουργία του ανθρωπίνου εγκεφάλου και προσπαθούν να συνδυάσουν τον τρόπο σκέψης αυτού με τον αφηρημένο μαθηματικό τρόπο σκέψης.

Ένα νευρωνικό δίκτυο χαρακτηρίζεται από την ιδιότητα του να επεξεργάζεται δεδομένα και να εξάγει αποτελέσματα, ενώ παράλληλα μπορεί να αποθηκεύει γνώση, να αξιοποιεί την αποθηκευμένη γνώση και να αποκτά νέα γνώση μέσω της εμπειρίας. Συχνά παρουσιάζει ανοχή σε λάθη αφού η επεξεργασία διαχέεται σε όλη την έκταση των συνδέσεων ενός δικτύου.

4.1.1. Νευρωνικά δίκτυα και κλασικός προγραμματισμός

Η μέθοδος με την οποία πραγματοποιούνται διάφοροι υπολογισμοί μέσω νευρωνικών δικτύων διαφέρει αρκετά με τον τρόπο που χρησιμοποιείται στον κλασικό προγραμματισμό. Κύρια διαφορά αποτελεί ότι το νευρωνικό δίκτυο μπορεί να εκπαιδευτεί χρησιμοποιώντας δεδομένα και αποτελέσματα μιας διαδικασίας να δημιουργήσει ένα τεχνητό μοντέλο αυτής αντί να προγραμματίζει το φυσικό μοντέλο της διαδικασίας το οποίο συνήθως δεν είναι γνωστό. Συνοπτικά, ένα νευρωνικό δίκτυο εμπίπτει σε ένα σύγχρονο μοντέλο λειτουργίας, το οποίο είναι ικανό για παράλληλη επεξεργασία, να εκπαιδευτεί με παραδείγματα έχοντας ανοχή στα σφάλματα, στο οποίο συνυπάρχουν η μνήμη και οι μονάδες επεξεργασίας. Αντιθέτως, ο κλασικός προγραμματισμός χαρακτηρίζεται από σειριακή επεξεργασία, με διαχωρισμό της μνήμης και των μονάδων επεξεργασίας χωρίς ανοχή στα σφάλματα.

4.1.2. Εφαρμογές τεχνητών δικτύων

Τα τεχνητά νευρωνικά δίκτυα προσφέρουν λύσεις σε προβλήματα που σχετίζονται με τον ανθρώπινο παράγοντα όπως αναγνώριση εικόνας, ομιλίας, κειμένου, δεδομένων και άλλα.

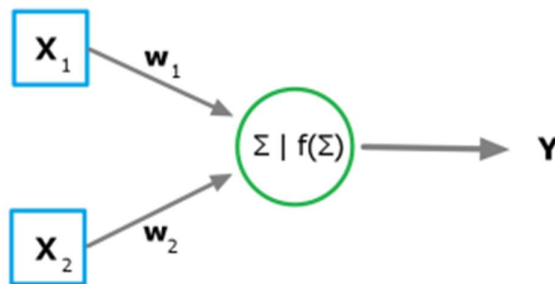
Η αξιοποίηση του ποικίλει και εκτείνεται σε διάφορες εφαρμογές όπως σχεδιασμός ενεργειών, χρονοπρογραμματισμός, ανάγνωση προτύπων, προβλέψεις η ακόμα να αποτελούν και μέρος συστημάτων ελέγχου που βασίζονται στην ασαφή λογική.

Ενδεικτικά, τεχνητά νευρωνικά δίκτυα αξιοποιούνται πλέον σε τομείς όπως η βιολογία, η ιατρική, στον στρατό, στα χρηματοοικονομικά, στο περιβάλλον και σε διάφορες βιομηχανίες.

4.1.3. Διαδικασία μάθησης

Τα τεχνητά νευρωνικά δίκτυα πραγματοποιούν δύο βασικές λειτουργίες, την εκπαίδευση και την ανάκληση. Η εκπαίδευση αποτελεί την τροποποίηση της τιμής των βαρών του δικτύου w_i , έχοντα συγκεκριμένο γνωστό διάνυσμα εισόδου, ώστε να παραχθεί συγκεκριμένο γνωστό διάνυσμα εξόδου και κατά επέκταση ένα μοντέλο. Συνήθως για την μάθηση αξιοποιείται ένα ποσοστό περίπου 60-70% από το διαθέσιμα δεδομένα εισόδου-εξόδου.

Η ανάκληση είναι η διαδικασία υπολογισμού ενός διανύσματος εξόδου για συγκεκριμένο διάνυσμα εισόδου με βάση τα βάρη που έχουν καθοριστεί από τη διαδικασία μάθησης. Πραγματοποιείται στο υπόλοιπο ποσοστό των διαθέσιμων δεδομένων εισόδου-εξόδου από αυτά που χρειάστηκαν για την εκπαίδευση.



Εικόνα 1. Κόμβοι εισόδου, βάρη νευρώνων, κόμβος εξόδου.

Υπάρχουν δύο είδη μάθησης, η υπό επίβλεψη και αυτή χωρίς επίβλεψη. Στην πρώτη στο δίκτυο δίνονται ζευγάρια διανυσμάτων εισόδου και επιθυμητής εξόδου, ενώ στη δεύτερη δίνονται μόνο δεδομένα εισόδου καθώς στόχος δεν αποτελεί η εκτίμηση των αποτελεσμάτων από τα δεδομένα εισόδου αλλά η επεξεργασία των δεδομένων εισόδου.

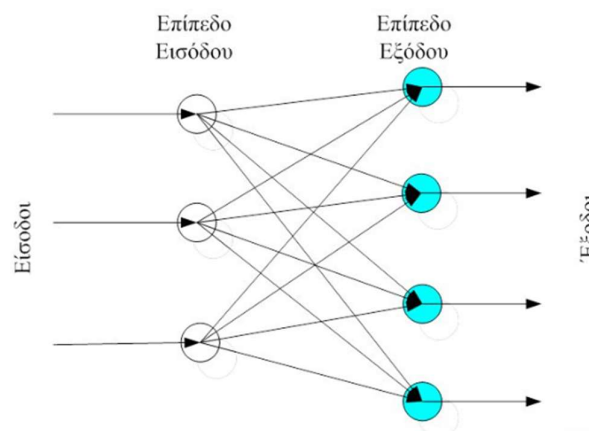
4.2. Είδη τεχνητών νευρικών δικτύων

Τοπολογία ή αρχιτεκτονική ενός τεχνητού νευρωνικού δικτύου ονομάζεται η τρόπος δόμησης των νευρώνων του και είναι άμεσα συνδεδεμένη με τον αλγόριθμο εκπαίδευσης που εφαρμόζεται στο δίκτυο. Οι νευρώνες μπορεί να θεωρηθεί ότι είναι τοποθετημένοι σε κάθετες στήλες τις οποίες προσδιορίζουμε ως στρώματα, ενώ ένα τεχνητό νευρωνικό δίκτυο αποτελείται τουλάχιστον από 2 στρώματα, το στρώμα εξόδου (input layer) και το στρώμα εξόδου (output layer).

Εάν ένα τεχνητό νευρωνικό δίκτυο αποτελείται από περισσότερα στρώματα από αυτά τα δύο, τότε τα ενδιάμεσα αυτά στρώματα τα ονομάζουμε κρυφά, καθώς και το δίκτυο πολυστρωματικό. Οι βασικές αρχιτεκτονικές τεχνητών νευρωνικών δικτύων είναι τρεις και θα αναλυθούν στην συνέχεια.

4.2.1. Εμπρόσθιας τροφοδοσίας ενός επιπέδου

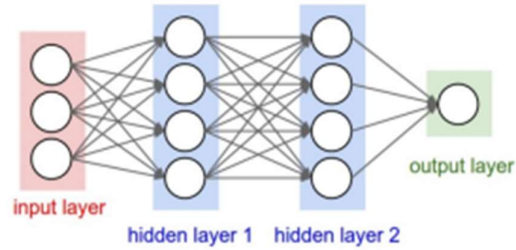
Αποτελεί την πιο απλή μορφή ενός τεχνητού νευρωνικού δικτύου και πρακτικά αποτελείται από ένα επίπεδο με κόμβους εισόδου και ένα επίπεδο με κόμβους εξόδου όπου γίνονται και οι υπολογισμοί. Το δίκτυο είναι αυστηρά εμπρόσθιας τροφοδότησης, δηλαδή κινείται αυστηρά από τους κόμβους εισόδου προς τους νευρώνες εξόδου. Η ορολογία του ενός επιπέδου αναφέρεται στο γεγονός πως υπάρχει μόνο ένα επίπεδο στο οποίο γίνονται οι υπολογισμοί, δηλαδή το επίπεδο εξόδου με τους νευρώνες.



Εικόνα 2. Τεχνητό νευρωνικό δίκτυο εμπρόσθιας τροφοδοσίας ενός επιπέδου.

4.2.2. Εμπρόσθια τροφοδότηση πολλαπλών επιπέδων

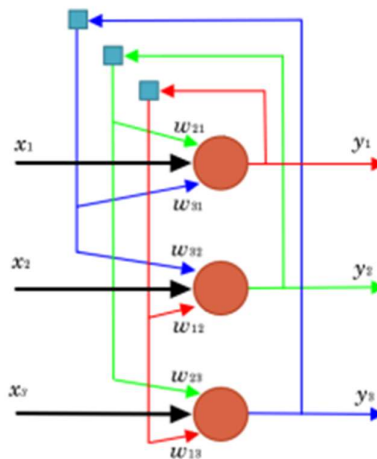
Στη δεύτερη περίπτωση τεχνητών νευρωνικών δικτύων υπάρχει η προσθήκη ενός ή περισσότερων κρυφών επιπέδων, με τους αντίστοιχους κρυφούς νευρώνες. Μεταξύ των κόμβων εισόδου και εξόδου, παρεμβάλλονται οι κρυφοί κόμβοι, οι οποίοι πρακτικά ανατροφοδοτούν τον επόμενο κρυφό κόμβο με νευρώνες-διανύσματα, μέχρι η πληροφορία να φτάσει στο κόμβο εξόδου. Το σύνολο των σημάτων εξόδου στο επίπεδο εξόδου αποτελεί και την απάντηση του δικτύου για τα δεδομένα που εισάγονται στους κόμβους εισόδου.



Εικόνα 3. Τεχνητό νευρωνικό δίκτυο εμπρόσθιας τροφοδοσίας πολλαπλών επιπέδων.

4.2.3. Ανατροφοδοτούμενα

Η διαφορά των ανατροφοδοτούμενων τεχνητών νευρωνικών δικτύων με τα εμπρόσθια τροφοδότησης, είναι η ύπαρξη τουλάχιστον ενός βρόγχου ανατροφοδότησης. Αυτό σημαίνει πως σε έναν τουλάχιστον νευρώνα μέσα στο δίκτυο το σήμα εξόδου επηρεάζει το σήμα που έρχεται στην είσοδο του νευρώνα.



Εικόνα 4. Ανατροφοδοτούμενο τεχνητό νευρωνικό δίκτυο.

4.5. Εκπαίδευση

Ένα τεχνητό νευρωνικό δίκτυο έχει ως βασική ιδιότητα να βελτιώνεται μέσω της εκπαίδευσής του από το περιβάλλον. Αυτό είναι απόρροια μιας διαδικασίας αλληλεπίδρασης για τον υπολογισμό των συναπτικών βαρών και των πολώσεων και έχει ως αποτέλεσμα την καλύτερη πληροφόρηση του δικτύου σχετικά με το περιβάλλον το οποίο βρίσκεται. Ένας τυπικός επίσημος όρος θεωρείται ο εξής:

Εκπαίδευση είναι μια διαδικασία κατά την οποία οι ελεύθεροι παράμετροι ενός τεχνητού νευρωνικού δικτύου προσαρμόζονται μέσω μιας διαδικασίας διέγερσης από το περιβάλλον, την οποία το δίκτυο ενσωματώνει. Ο τύπος της εκπαίδευσης είναι καθορισμένος από τον τρόπο με τον οποίο οι αλλαγές των παραμέτρων πραγματοποιούνται.

Όταν αναφερόμαστε στην εκπαίδευση του δικτύου, αυτή καθορίζεται αρχικά με το τεχνητό νευρωνικό δίκτυο να διεγείρεται από το περιβάλλον. Εν συνεχεία, το δίκτυο υποβάλλεται στις αλλαγές στις ελεύθερες παραμέτρους του, ως αποτέλεσμα της διέγερσης. Τέλος, το δίκτυο αποκρίνεται εκ νέου στο περιβάλλον του λόγω της αλλαγής στην εσωτερική δομή του.

Υπό σωστές συνθήκες και ορθώς ορισμένους κανόνες, καταλήγουμε στην λύση του προβλήματος της εκπαίδευσης, δηλαδή έναν αλγόριθμο εκπαίδευσης. Κάθε αλγόριθμος εκπαίδευσης διαφέρει στον τρόπο με τον οποίο ρυθμίζεται το συναπτικό βάρος ενός νευρώνα, για αυτό και οι αλγόριθμοι αυτοί ποικίλουν.

4.6. Βαθιά μάθηση (Deep learning)

Μια από τις μεγαλύτερες εισαγωγές στα νευρωνικά δίκτυα είναι η βαθιά μάθηση, η οποία αποτελεί κλάδο της μηχανικής μάθησης και επιτρέπει στους υπολογιστές να λειτουργούν όπως ένας άνθρωπος, να μαθαίνουν δηλαδή εμπειρικά. Με διάφορες υπολογιστικές λοιπόν, τους δίνεται η δυνατότητα να διαβάζουν τις πληροφορίες από τα δεδομένα, χωρίς να είναι απαραίτητα η παρουσία κάποιου προκαθορισμένης συνάρτησης που θα λειτουργεί ως μοντέλο. Τα μοντέλα βαθιά μάθησης μπορούν να πετύχουν μεγάλη ακρίβεια στην ταξινόμηση αντικειμένων.

4.6.1. LSTM Δίκτυο (Long Short-Term Memory)

Πλέον σε διάφορες εφαρμογές σε διάφορους κλάδους της επιστήμης, έχει αποδειχτεί ότι είναι πιο χρήσιμα τα νευρωνικά δίκτυα μακράς διάρκειας μνήμης. Αυτό δικαιολογείται στην ικανότητά τους να μάθουν μακροπρόθεσμες εξαρτήσεις αφού καταφέρνουν να ανακαλούν προηγούμενες καταστάσεις του δικτύου, χρησιμοποιώντας συνδέσεις ανατροφοδότησης.

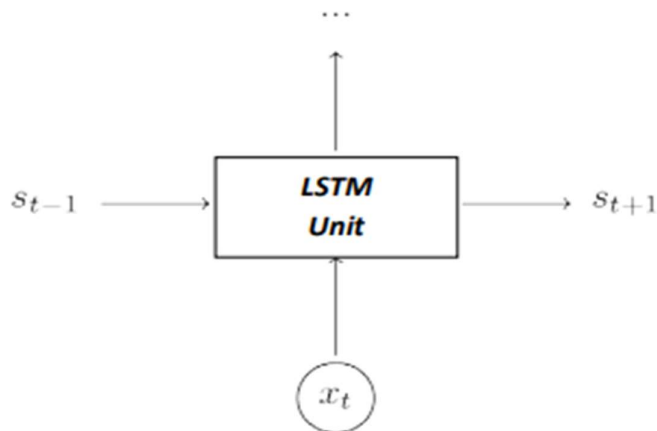
Πιο συγκεκριμένα τα LSTM δίκτυα αποτελούν μια εξελιγμένη εκδοχή των ανατροφοδοτούμενων νευρωνικών δικτύων (Recurrent Neural Networks – RNN), τα οποία εξαρχής δημιουργήθηκαν για την εξάλειψη του προβλήματος της ελλιπής μνήμης που έχουν τα απλά νευρωνικά δίκτυα τροφοδοσίας και είχαν καταστεί αναξιόπιστα σε αλληλουχίες και προβλήματα χρονοσειρών. Τα ανατροφοδοτούμενα νευρωνικά δίκτυα χρησιμοποιούν κυκλικές συνδέσεις στο κρυφό στρώμα για να έχουν μακροπρόθεσμη μνήμη και να διαχειρίζονται σωστά πληροφορίες από αλληλουχίες και δεδομένα χρονοσειρών.

Η διαφορά των κλασικών αυτών δικτύων RNN με τα LSTM δίκτυα είναι πως τα πρώτα περιορίζουν τα μοντέλα να μάθουν εξαρτήσεις μεγάλου εύρους, ενώ τα δεύτερα αποθηκεύουν πληροφορίες σε κελιά μνήμης, εξαλείφουν την περιττή πληροφορία και έτσι έχουν καλύτερη απόδοση και αξιοπιστία.

Για την αναλυτικότερη ερμηνεία ενός LSTM θα ερμηνευτεί η υπόσταση του μέσω συναρτήσεων.

$$\begin{aligned}i &= \sigma(x_t U^i + s_{t-1} W^i) \\f &= \sigma(x_t U^f + s_{t-1} W^f) \\o &= \sigma(x_t U^o + s_{t-1} W^o) \\g &= \tanh(x_t U^g + s_{t-1} W^g) \\c_t &= c_{t-1} \circ f + g \circ i \\s_t &= \tanh(c_t) \circ o \longrightarrow s_t = \tanh(U x_t + W s_{t-1})\end{aligned}$$

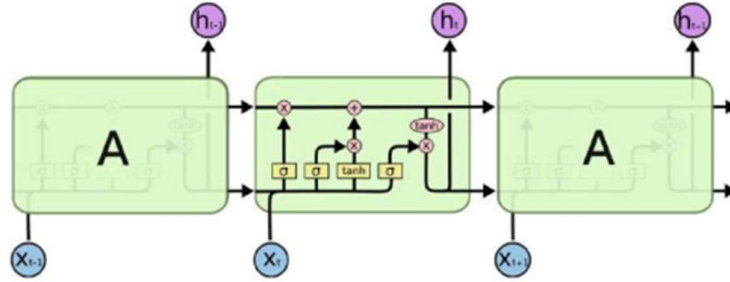
Εάν παρατηρήσουμε και αναλύσουμε τον υπολογισμό της κρυφής κατάστασης η οποία συμβολίζεται με S_t , αυτή εξαρτάται από την είσοδο μιας δεδομένης χρονικής στιγμής t , που συμβολίζεται με τον όρο X_t , αλλά και την κρυφή κατάσταση μιας προηγούμενης χρονικής στιγμής $t-1$, η οποία συμβολίζεται με S_{t-1} . Συμπερασματικά κάθε δομική μονάδα LSTM μπορεί να θεωρηθεί ως ένα ξεχωριστό κουτί όπου με βάση μια δοθείσας εισόδου αλλά και της προηγούμενης κρυφής κατάστασης, μπορούν να υπολογίσουν την επόμενη κρυφή κατάσταση.



Εικόνα 5. Το Black Box ενός LSTM μοντέλου

Παρακάτω θα ακολουθήσει μια αναλυτικότερη ανάπτυξη στην επεξήγηση των συναρτήσεων που λαμβάνουν χώρα σε ένα LSTM δίκτυο.

- Τα γράμματα i , f , o αποτελούν αρχικά για τις πύλες input, forget & output. Παρατηρείται πως οι εξισώσεις οι οποίες περιγράφουν αυτές τις πύλες είναι ίδιες, με εξαίρεση πως έχουν διαφορετικούς πίνακες μεταβλητών. Η ονομασία των πυλών προκύπτει από το ότι χαρακτηρίζονται από σιγμοειδής συναρτήσεις που συμπιέζουν αυτά τα διανύσματα σε τιμές μεταξύ του 0 και του 1, τα οποία όταν πολλαπλασιαστούν με άλλα διανύσματα γίνεται έλεγχος της επιρροής του διανύσματος στην έξοδο. Με λίγα λόγια, η πρώτη πύλη καθορίζει το ποσοστό της νέας υπολογισμένης κατάστασης που θα επηρεάσει την τρέχουσα είσοδο. Η δεύτερη πύλη το ποσοστό της προηγούμενης το οποίο θα αξιοποιηθεί, ενώ η τρίτη πύλη το ποσοστό της εσωτερικής κατάστασης που θα αξιοποιηθεί στο εξωτερικό δίκτυο.
- Η συνάρτηση ορισμένη ως g βασισμένη στην είσοδο της τρέχουσας χρονικής στιγμής και στην κρυφή κατάσταση της προηγούμενης χρονικής στιγμής, υπολογίζει την υποψήφια κρυφή κατάσταση.
- Η συνάρτηση ορισμένη ως c_t απαρτίζει την εσωτερική μνήμη της μονάδας. Η θεωρητική της επεξήγηση αποτελεί συνδυασμό της εσωτερικής μνήμης της προηγούμενης χρονικής στιγμής πολλαπλασιασμένη με την πύλη forget και της νέας κρυφής κατάστασης g πολλαπλασιασμένη με την πύλη εισόδου. Στην πραγματικότητα μπορεί να μεταφραστεί ως η προσπάθεια να ενωθεί η μνήμη της τρέχουσας χρονικής στιγμής με την μνήμη της προηγούμενης χρονικής στιγμής.
- Έχοντας ως δεδομένο την εσωτερική μνήμη c_t , καθίσταται εφικτός ο υπολογισμός της κρυφής κατάστασης εξόδου s_t πολλαπλασιάζοντας την εσωτερική μνήμη με την πύλη εξόδου.



Εικόνα 6. Long short term memory νευρωνικό δίκτυο.

4.6.2. Μέθοδος ADAM

Κατά τη διαδικασία εκπαίδευσης ενός τεχνητού νευρωνικού δικτύου είναι φυσικό και επόμενο να προκύπτουν σφάλματα, κατάσταση που αντιμετωπίζεται μέσω ενός αλγορίθμου βελτιστοποίησης. Στη συγκεκριμένη πρακτική εφαρμογή που θα ακολουθήσει, η γλώσσα προγραμματισμού Python μας επιτρέπει αυτόματα να τρέξουμε αυτόν τον αλγόριθμο βελτίωσης, ενώ στη συνέχεια θα ακολουθήσει μια επιφανειακή παρουσίαση της διαδικασίας λειτουργίας του.

Ο αλγόριθμος ADAM προσαρμόζει το ρυθμό εκπαίδευσης για κάθε παράμετρο, εξάγοντας ένα τρέχον τετραγωνικό μέσο όρο των δυναμικών αλλά ταυτόχρονα κρατώντας και έναν μέσο όρο παλαιών δυναμικών. Για αρχή, γίνεται υπολογισμός των παλαιών μέσων όρων m_t και των παλαιών τετραγωνικών δυναμικών u_t , οι οποίοι αποτελούν εκτιμήσεις του μέσου όρου και της διακύμανσης των δυναμικών.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$u_t = \beta_2 u_{t-1} + (1 - \beta_2) g_t^2$$

Οι όροι αυτοί αποτελούν διανύσματα μηδενικών στην αρχή, οι δημιουργοί του ADAM έχουν δώσει προσοχή στην προδιάθεση να μετακινηθούν προς το 0. Για την εξάλειψη του φαινομένου αυτού, υπολογίζονται οι όροι αφαιρώντας την προδιάθεση προς το 0 με αυτούς τους τύπους:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{u}_t = \frac{u_t}{1 - \beta_2^t}$$

Τελικά, οι όροι που αξιοποιούνται για τον υπολογισμό των παραμέτρων αυτών είναι οι εξής:

$$\Delta \theta_t = -\frac{\eta}{\sqrt{\hat{u}_t} + \epsilon} \hat{m}_t$$

$$\theta_{t+1} = \theta_t + \Delta \theta_t$$

5. Εφαρμογή της διαδικασίας και κώδικας Python [1,3,4,5,6,10,11,13,15,16]

5.1. Περιγραφή του προβλήματος και μέσα υλοποίησης

Θα ακολουθήσει η αναλυτική περιγραφή της μεθοδολογικής προσέγγισης για την πρόβλεψη τιμών του κρυπτονομίσματος Bitcoin, όντως το πιο δημοφιλές τα τελευταία χρόνια και έχοντας τεράστια εφαρμογή στις τωρινές οικονομικές συναλλαγές ανά τον κόσμο. Η Πρόβλεψη θα γίνει αξιοποιώντας το μοντέλο βαθιάς μάθησης LSTM – Νευρωνικών δικτύων μακροχρόνιας μνήμης ενώ το μοντέλο θα εκπαιδευτεί με τη μέθοδο Adam.

Ο Κώδικας υλοποιήθηκε στην Python 3.10 στο εικονικό περιβάλλον της Pycharm με τη χρήση των βιβλιοθηκών Keras και Tensorflow, ενώ θα παρουσιαστούν και οι μετρητές απόδοσης (Mean Absolute Error-MAE, Root Mean Square Error-RMSE, Mean Squared Error-MSE) του μοντέλου για την σύγκριση των αποτελεσμάτων και τον προσδιορισμό της αξίας της υλοποίησης.

5.1.1. Συλλογή δεδομένων

Τα δεδομένα για τις χρηματικές τιμές του Bitcoin αποκτήθηκαν μέσω της ιστοσελίδας Yahoo Finance από την καρτέλα Historical Data. Από αυτά τα δεδομένα μπορούμε να έχουμε όλα τα στοιχεία για το κρυπτονόμισμα για κάθε μέρα από τις 17/9/2014 μέχρι και σήμερα. Τα δεδομένα αυτά αποθηκεύτηκαν σε ένα αρχείο .csv και θα αξιοποιηθούν στο κώδικα. Για την απλότητα του πειράματος, η πρόβλεψη ορίστηκε μονάχα στην τιμή κλεισίματος.

5.2. Ανάλυση και επεξήγηση του κώδικα

Σε κάθε πλαίσιο στο οποίο θα προηγείται ο όρος "IN" θα εννοείται η γραμμή του κώδικα η οποία εκτελείται, ενώ σε κάθε πλαίσιο με τον όρο "OUT" θα εννοείται η έξοδος των εντολών που έχουν δοθεί. Για λόγους καλύτερης απεικόνισης, κάθε πλαίσιο εισαγωγής εντολών μπορεί να έχει αριθμημένα πακέτα εντολών με το σύμβολο της δίσησης, καθώς ανάλογα θα παρουσιάζονται οι έξοδοι τους, με τον αντίστοιχο αντιπροσωπευτικό αριθμό.

5.2.1. Εισαγωγή

IN:

```
#1
import os
import pandas as pd
import numpy as np
import math
import datetime as dt
import matplotlib.pyplot as plt

#2
from sklearn.metrics import mean_squared_error, mean_absolute_error,
explained_variance_score, r2_score
from sklearn.metrics import mean_poisson_deviance,
mean_gamma_deviance, accuracy_score
from sklearn.preprocessing import MinMaxScaler

#3
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.layers import LSTM

#4
import matplotlib.pyplot as plt
from itertools import cycle
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots
```

Αρχικά θα εισάγουμε τις απαραίτητες βιβλιοθήκες στον κώδικα. Το πρώτο σκέλος αναφέρεται σε γενικές βιβλιοθήκες, το δεύτερο σκέλος αποτελεί εργαλεία απαραίτητα για την αξιολόγηση του κώδικα, το τρίτο σκέλος εργαλεία για την δημιουργία του μοντέλου πρόβλεψης ενώ το τελευταίο σκέλος θα αξιοποιηθεί για την γραφική απεικόνιση των αποτελεσμάτων.

5.2.2. Εισαγωγή των δεδομένων

IN:

```
maindf = pd.read_csv(r"C:\Users\jimka\Desktop\opa\45\123.csv")
print('Total number of days present in the dataset: ',
maindf.shape[0])
print('Total number of fields present in the dataset: ',
maindf.shape[1])
```

OUT:

```
Total number of days present in the dataset: 3411
```

```
Total number of fields present in the dataset: 7
```

Διαβάζοντας το αρχείο με τις τιμές του Bitcoin, ο κώδικας συμπεραίνει πως του έχουμε εισάγει δεδομένα για 3411 μέρες.

Με τις επόμενες εντολές θα ζητήσουμε από τον κώδικα να μας προβάλει τις πρώτες τιμές που εμφανίζονται στο αρχείο αλλά και τις τελευταίες, για να επαληθεύσουμε πως όλες οι τιμές έχουν περαστεί σωστά. Επίσης θα ζητήσουμε να μας προβάλει τον αριθμό των γραμμών και των στηλών του αρχείου και την περιγραφή των στοιχείων.

IN:

```
maindf.shape
print(maindf.shape)

maindf.head()
print(maindf.head)
maindf.tail()
print(maindf.tail)

maindf.info()
print(maindf.info)
maindf.describe()
print(maindf.describe)
```

OUT:

of	Date	Open	...	Adj Close	Volume
0	2014-09-17	465.864014	...	457.334015	21056800
1	2014-09-18	456.859985	...	424.440002	34483200
2	2014-09-19	424.102997	...	394.795990	37919700
3	2014-09-20	394.673004	...	408.903992	36863600
4	2014-09-21	408.084991	...	398.821014	26580100
...
3406	2024-01-14	42842.261719	...	41796.269531	17521429522
3407	2024-01-15	41715.066406	...	42511.968750	22320220558
3408	2024-01-16	42499.335938	...	43154.945313	24062872740
3409	2024-01-17	43132.101563	...	42742.652344	20851232595
3410	2024-01-18	42742.312500	...	42406.933594	20322904064

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3411 entries, 0 to 3410
Data columns (total 7 columns):
# Column Non-Null Count Dtype
---  ---  ---
0 Date 3411 non-null object
1 Open 3411 non-null float64
2 High 3411 non-null float64
3 Low 3411 non-null float64
4 Close 3411 non-null float64
5 Adj Close 3411 non-null float64
6 Volume 3411 non-null int64
dtypes: float64(5), int64(1), object(1)
memory usage: 186.7+ KB
```

5.2.3. Έλεγχος για μηδενικές ή μη υπάρχουσες τιμές στο αρχείο

IN:

```
print('Null Values:', maindf.isnull().values.sum())
print('NA values:', maindf.isnull().values.any())
```

Ο κώδικας θα μας εμφανίσει πως ορθώς δεν υπάρχουν κενές ή μηδενικές τιμές στο αρχείο.

```
Null Values: 0
NA values: False
```

5.2.4. Ερευνητική ανάλυση δεδομένων

Στη συνέχεια, θα ερευνήσουμε την πρώτη και τελευταία ημερομηνία για τις οποίες έχουμε δεδομένα τιμών(#1), αλλά και μια αναλυτικότερη εμβάθυνση στις διαθέσιμες τιμές για το πρώτο έτος που κατέχουμε, δηλαδή το 2014. Για την απλοποίηση της ανάλυσης θα κρατήσουμε από τα δεδομένα του πίνακα την ημερομηνία, την τιμή ανοίγματος, κλεισίματος, την υψηλότερη καθώς και την χαμηλότερη τιμή κάθε ημέρας(#2).

IN:

```
#1
sd = maindf.iloc[0][0]
ed = maindf.iloc[-1][0]

print('Starting Date', sd)
print('Ending Date', ed)

#2
maindf['Date'] = pd.to_datetime(maindf['Date'], format='%Y-%m-%d')

y_2014 = maindf.loc[(maindf['Date'] >= '2014-09-17') &
                   (maindf['Date'] < '2014-12-31')]

y_2014.drop(y_2014[['Adj Close', 'Volume']], axis=1)

#3
monthwise = y_2014.groupby(y_2014['Date'].dt.strftime('%B'))[['Open',
'Close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June',
'July', 'August',
              'September', 'October', 'November', 'December']
monthwise = monthwise.reindex(new_order, axis=0)
monthwise
print(monthwise)
```

OUT: 1,2,3

Starting Date 2014-09-17
Ending Date 2024-01-18

/	Date	Open	High	Low	Close
0	2014-09-17	465.864014	468.174011	452.421997	457.334015
1	2014-09-18	456.859985	456.859985	413.104004	424.440002
2	2014-09-19	424.102997	427.834991	384.532013	394.795990
3	2014-09-20	394.673004	423.295990	389.882996	408.903992
4	2014-09-21	408.084991	412.425995	393.181000	398.821014
...
100	2014-12-26	319.152008	331.424011	316.627014	327.924011
101	2014-12-27	327.583008	328.911011	312.630005	315.863007
102	2014-12-28	316.160004	320.028015	311.078003	317.239014
103	2014-12-29	317.700989	320.266998	312.307007	312.670013
104	2014-12-30	312.718994	314.808990	309.372986	310.737000

	Open	Close
Date	NaN	NaN
January	NaN	NaN
February	NaN	NaN
March	NaN	NaN
April	NaN	NaN
May	NaN	NaN
June	NaN	NaN
July	NaN	NaN
August	NaN	NaN
September	412.654003	407.182428
October	365.748000	364.148873
November	364.850235	366.099799
December	344.146864	341.970366

Όπως βλέπουμε από τη στιγμή που δεν έχουμε πληροφορίες για ορισμένους μήνες στην αρχή του έτους, ο κώδικας μας ενημερώνει πως αυτές είναι μη διαθέσιμες. Επομένως για τις τιμές τις οποίες έχουμε θα δημιουργήσουμε 2 ραβδοδιαγράμματα που θα περιγράψουν την τιμή ανοίγματος και την τιμή κλεισίματος, καθώς και την υψηλότερη και χαμηλότερη τιμή αντίστοιχα, ανά μήνα για το 2014.

IN:

```
#1
#2014 open and close price
fig = go.Figure()

fig.add_trace(go.Bar(
    x=monthwise.index,
    y=monthwise['Open'],
    name='Bitcoin Open Price',
    marker_color='crimson'
))
fig.add_trace(go.Bar(
    x=monthwise.index,
    y=monthwise['Close'],
    name='Bitcoin Close Price',
    marker_color='lightsalmon'
))

fig.update_layout(barmode='group', xaxis_tickangle=-45,
                  title='Monthwise comparison between Bitcoin open
and close price')
fig.show()

#2
#2014 high low price
y_2014.groupby(y_2014['Date'].dt.strftime('%B'))['Low'].min()
monthwise_high =
y_2014.groupby(maindf['Date'].dt.strftime('%B'))['High'].max()
monthwise_high = monthwise_high.reindex(new_order, axis=0)

monthwise_low =
y_2014.groupby(y_2014['Date'].dt.strftime('%B'))['Low'].min()
monthwise_low = monthwise_low.reindex(new_order, axis=0)

fig = go.Figure()
fig.add_trace(go.Bar(
    x=monthwise_high.index,
    y=monthwise_high,
    name='Bitcoin high Price',
    marker_color='rgb(0, 153, 204)'
))
fig.add_trace(go.Bar(
    x=monthwise_low.index,
    y=monthwise_low,
    name='Bitcoin low Price',
    marker_color='rgb(255, 128, 0)'
))

fig.update_layout(barmode='group',
                  title=' Monthwise High and Low Bitcoin price')
fig.show()

#3
#2014 stock analysis chart
names = cycle(['Bitcoin Open Price', 'Bitcoin Close Price', 'Bitcoin
High Price', 'Bitcoin Low Price'])

fig = px.line(y_2014, x=y_2014.Date, y=[y_2014['Open'],
y_2014['Close'],
y_2014['High'],
```

```

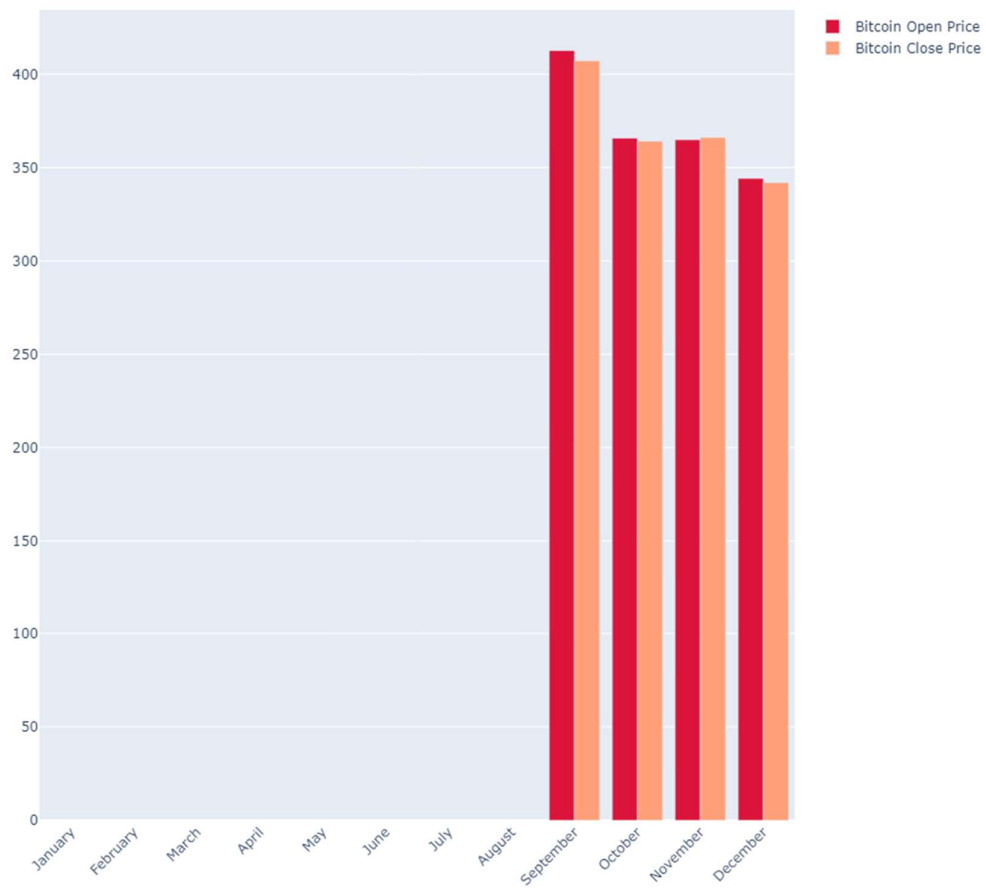
y_2014['Low']],
labels={'Date': 'Date', 'value': 'Bitcoin value'})
fig.update_layout(title_text='Bitcoin analysis chart', font_size=15,
font_color='black',
                    legend_title_text='Bitcoin Parameters')
fig.for_each_trace(lambda t: t.update(name=next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()

```

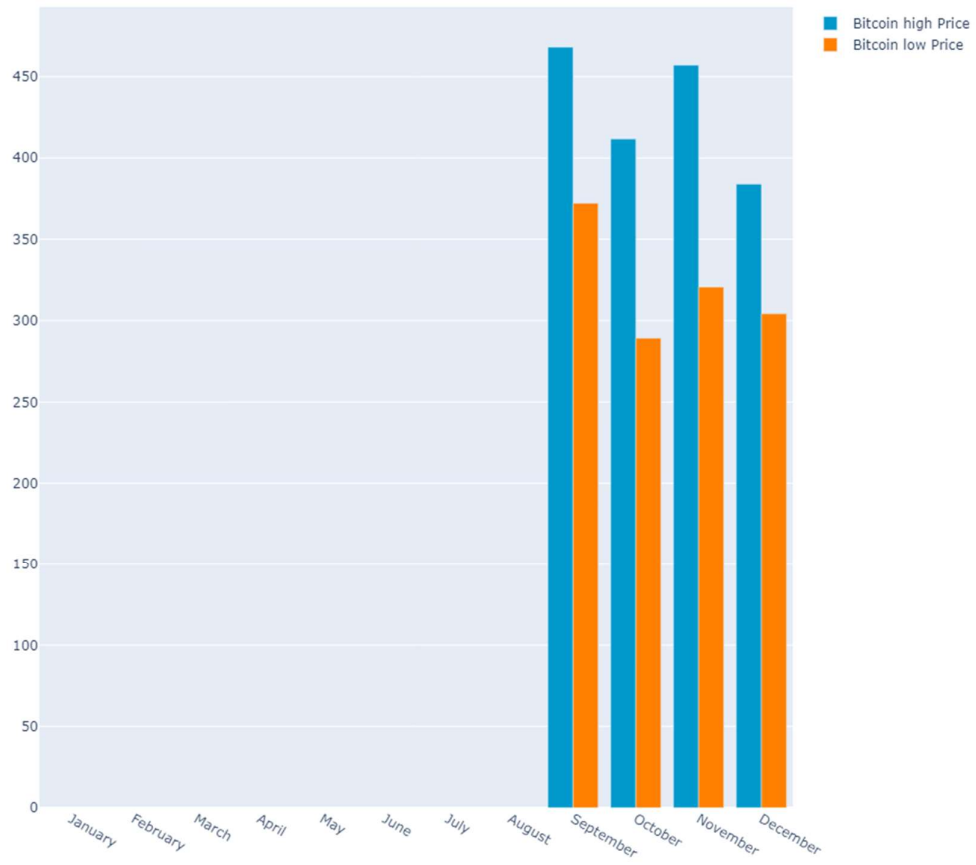
OUT: #1,2,3

Monthwise comparison between Bitcoin open and close price



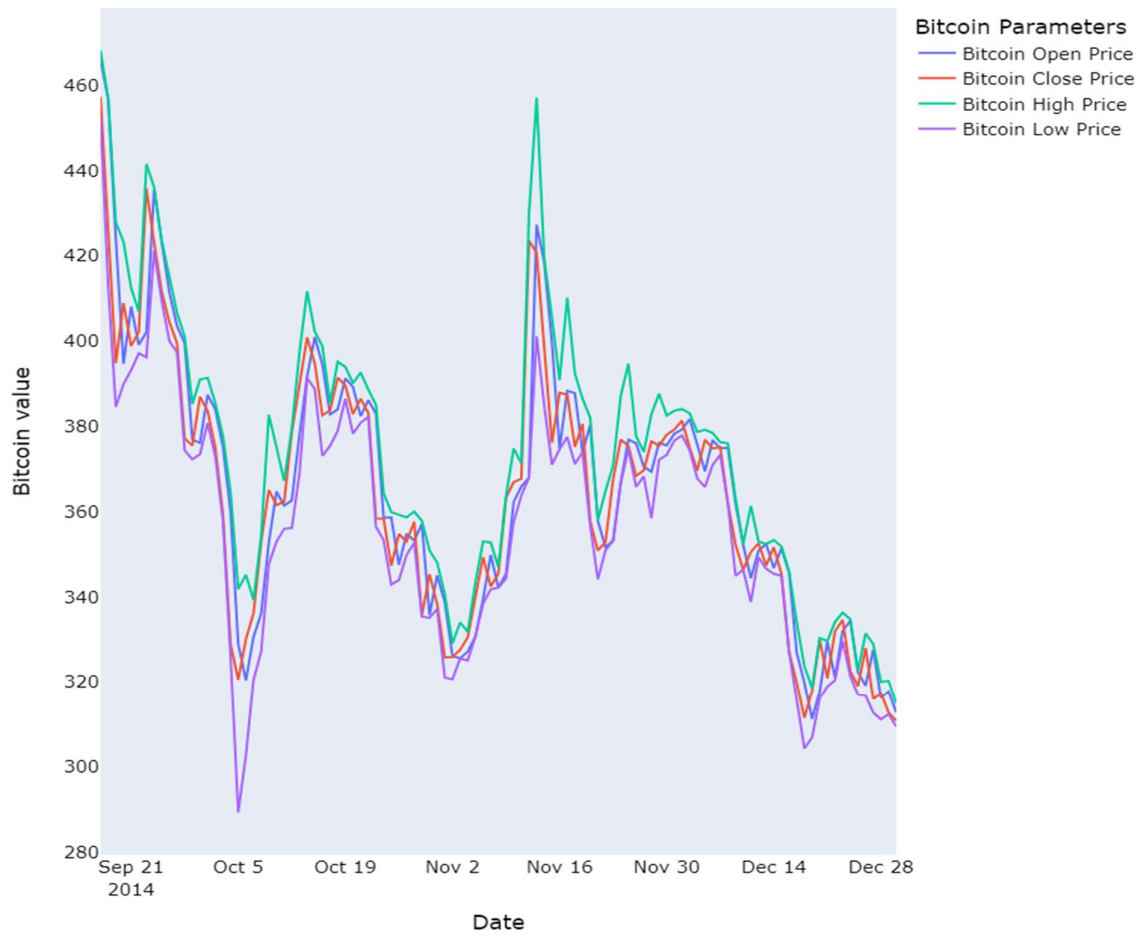
Εικόνα 7. Τιμή ανοίγματος-κλεισίματος του Bitcoin για το 2014.

Monthwise High and Low Bitcoin price



Εικόνα 8. Υψηλότερη και χαμηλότερη τιμή του Bitcoin για το 2014.

Bitcoin analysis chart



Εικόνα 9. Γραφική αναπαράσταση της τιμής ανοίγματος-κλεισίματος και της υψηλότερης-χαμηλότερης τιμής του Bitcoin για το 2014.

5.2.5. Ανάλυση Bitcoin 2014-2024

Η ανάλυση του κρυπτονομίσματος θα συνεχιστεί με μια συνοπτική διερεύνηση στο σύνολο των δεδομένων τιμών τις οποίες μπορεί κάποιος να βρει αναρτημένες στο διαδίκτυο, από την αρχή της καταγραφής έως και σήμερα. Ο κώδικας που αξιοποιήθηκε είναι ο ίδιος με την πειραματική διερεύνηση που έγινε πρώτα, με φυσικά τις αλλαγές στις παραμέτρους των ημερομηνιών.

IN:

```
#1
maindf['Date'] = pd.to_datetime(maindf['Date'], format='%Y-%m-%d')

y_overall = maindf.loc[(maindf['Date'] >= '2014-09-17')
                      & (maindf['Date'] <= '2024-01-18')]

y_overall.drop(y_overall[['Adj Close', 'Volume']], axis=1)

print(maindf)

monthwise=
y_overall.groupby(y_overall['Date'].dt.strftime('%B'))[['Open',
'Close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June',
'July', 'August',
             'September', 'October', 'November', 'December']
monthwise = monthwise.reindex(new_order, axis=0)
monthwise

#2
names = cycle(['Bitcoin Open Price', 'Bitcoin Close Price', 'Bitcoin
High Price', 'Bitcoin Low Price'])

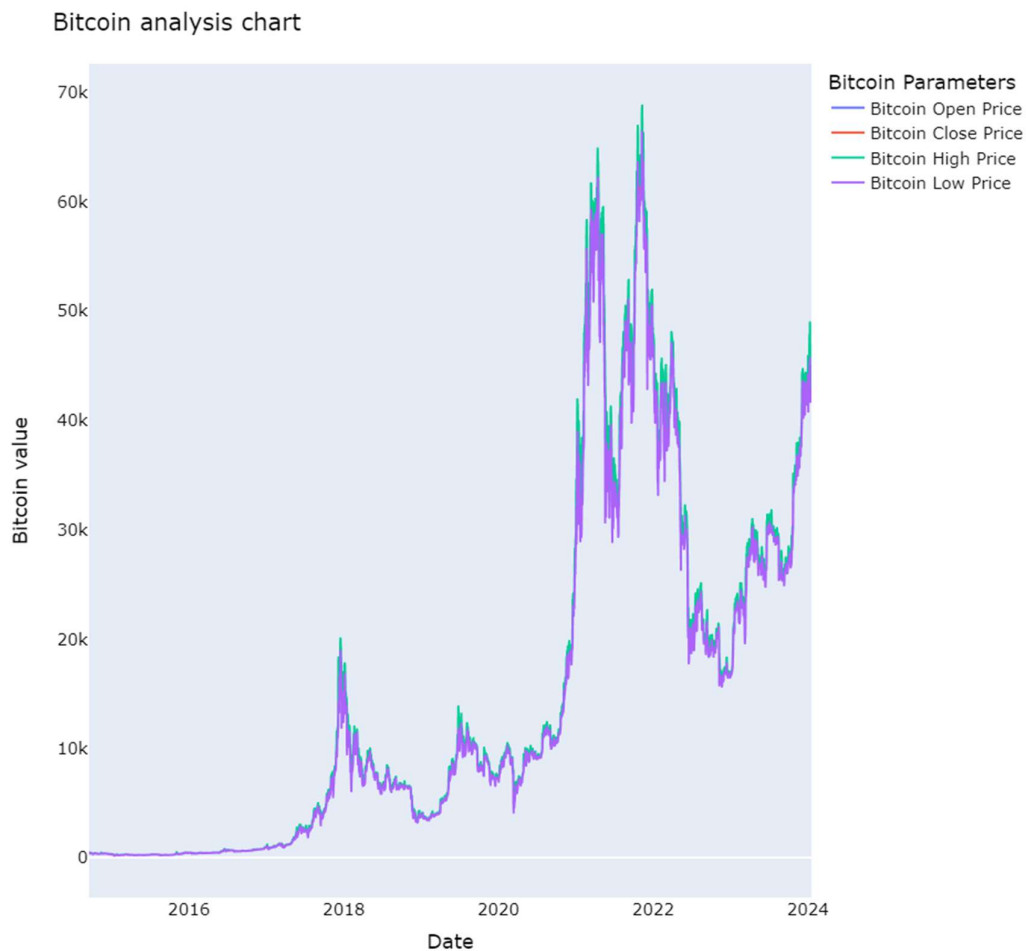
fig = px.line(y_overall, x=y_overall.Date, y=[y_overall['Open'],
y_overall['Close'],
                                     y_overall['High'],
y_overall['Low']],
             labels={'Date': 'Date', 'value': 'Bitcoin value'})
fig.update_layout(title_text='Bitcoin analysis chart', font_size=15,
font_color='black',
                 legend_title_text='Bitcoin Parameters')
fig.for_each_trace(lambda t: t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()
```

OUT: #1,2

	Date	Open	High	Low	Close
0	2014-09-17	465.864014	468.174011	452.421997	457.334015
1	2014-09-18	456.859985	456.859985	413.104004	424.440002
2	2014-09-19	424.102997	427.834991	384.532013	394.795990
3	2014-09-20	394.673004	423.295990	389.882996	408.903992
4	2014-09-21	408.084991	412.425995	393.181000	398.821014
...
3406	2024-01-14	42842.261719	43065.597656	41724.613281	41796.269531
3407	2024-01-15	41715.066406	43319.722656	41705.417969	42511.968750
3408	2024-01-16	42499.335938	43566.273438	42085.996094	43154.945313
3409	2024-01-17	43132.101563	43189.890625	42189.308594	42742.652344
3410	2024-01-18	42742.312500	42864.671875	42405.664063	42406.933594

[3411 rows x 7 columns]



Εικόνα 10. Γραφική αναπαράσταση της τιμής ανοίγματος-κλεισίματος και της υψηλότερης-χαμηλότερης τιμής του Bitcoin για τη περίοδο 2014 έως σήμερα.

5.2.6. Δημιουργία μοντέλου πρόβλεψης LSTM

5.2.6.1. Δεδομένα

Σε αυτό το σκέλος θα δημιουργήσουμε το μοντέλο πρόβλεψης. Θα χωρίσουμε τα δεδομένα μας σε ένα κομμάτι που θα αποτελεί το μέρος της εκπαίδευσης του αλγόριθμου (Training Data) και ένα κομμάτι θα αποτελεί το μέρος των δοκιμών (Test Data). Επειδή το Bitcoin εμφανίζει ειδικά στις αρχές του μεγάλες αποκλίσεις στις τιμές με την πάροδο του χρόνου, όπως για παράδειγμα από το 2014 στα 200 δολάρια εκτοξεύτηκε το 2018 στα 15000 δολάρια, θα επιλέξουμε για το Training set μονάχα 1 χρόνο από τα δεδομένα που έχουμε και συγκεκριμένα από τις 18-1-2023 έως και σήμερα. Για την απλοποίηση των δεδομένων και στην στοχευμένη καλύτερη απόδοση του αλγορίθμου, ο πίνακας των δεδομένων θα απλοποιηθεί στο να συμπεριλαμβάνει πέρα από τα χρονολογικά δεδομένα κάθε ημέρας, μόνο την τιμή κλεισίματος κάθε ημέρας.

IN:

```
closedf = maindf[['Date', 'Close']]
print("Shape of close dataframe:", closedf.shape)
```

OUT:

```
Shape of close dataframe: (3411, 2)
```

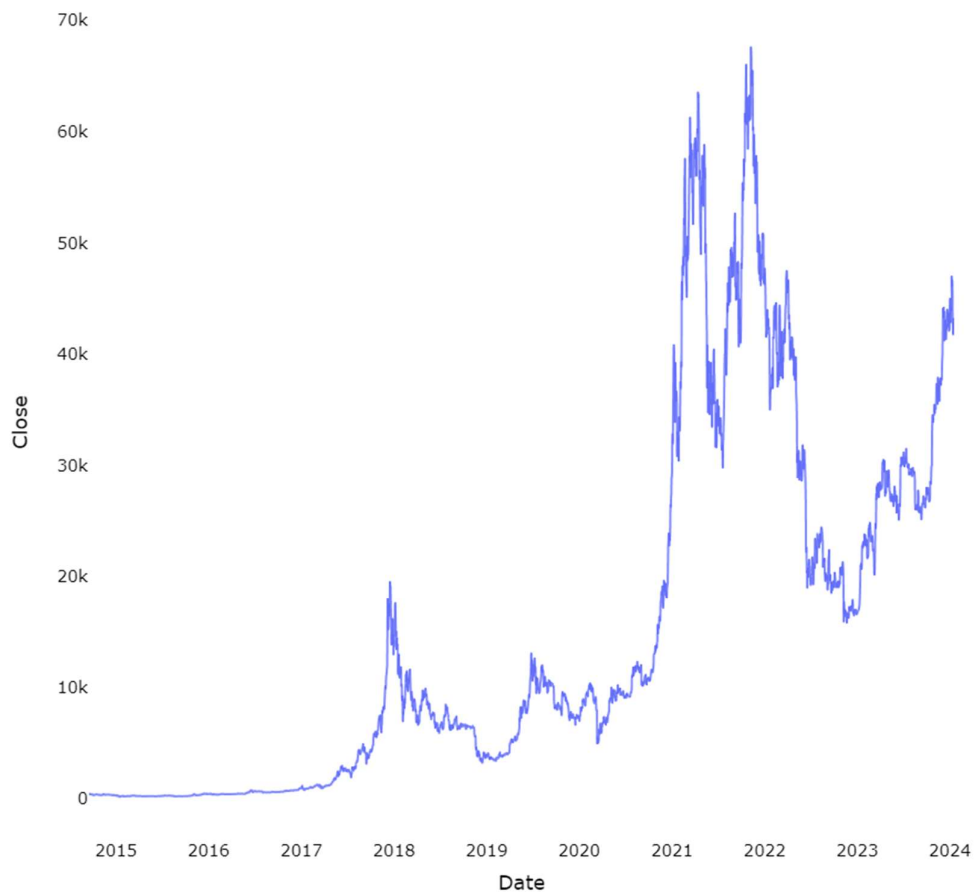
Στη συνέχεια θα αξιοποιήσουμε μια γραφική παράσταση για να προβάλλουμε την τιμή κλεισίματος ανά τα χρόνια.

IN:

```
fig = px.line(closedf, x=closedf.Date, y=closedf.Close,
              labels={'date': 'Date', 'close': 'Close Stock'})
fig.update_traces(marker_line_width=2, opacity=0.8,
                  marker_line_color='orange')
fig.update_layout(title_text='Whole period of timeframe of Bitcoin
close price 2014-2024', plot_bgcolor='white',
                  font_size=15, font_color='black')
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()
```

OUT:

Whole period of timeframe of Bitcoin close price 2014-2024



Εικόνα 11. Γραφική αναπαράσταση της τιμής κλεισίματος του Bitcoin από το 2014 έως σήμερα.

Θα γίνει η απομάκρυνση των περιττών τιμών ώστε οι διαθέσιμες τιμές για το μοντέλο να είναι αποκλειστικά αυτές των τελευταίων 365 ημερών.

IN:

```
closedf = closedf[closedf['Date'] > '2023-01-18']
close_stock = closedf.copy()
print("Total data for prediction: ", closedf.shape[0])
print(closedf)
```

OUT:

Total data for prediction: 365

	Date	Close
3046	2023-01-19	21086.792969
3047	2023-01-20	22676.552734
3048	2023-01-21	22777.625000
3049	2023-01-22	22720.416016
3050	2023-01-23	22934.431641
...
3406	2024-01-14	41796.269531
3407	2024-01-15	42511.968750
3408	2024-01-16	43154.945313
3409	2024-01-17	42742.652344
3410	2024-01-18	42406.933594

[365 rows x 2 columns]

Ακολουθεί το διάγραμμα των τιμών κλεισίματος των ημερών που θα χρησιμοποιηθούν.

IN:

```
fig = px.line(closedf, x=closedf.Date, y=closedf.Close,
labels={'date': 'Date', 'close': 'Close Stock'})
fig.update_traces(marker_line_width=2, opacity=0.8,
marker_line_color='orange')
fig.update_layout(title_text='Considered period to predict Bitcoin
close price',
plot_bgcolor='white', font_size=15,
font_color='black')
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()
```

OUT:

Considered period to predict Bitcoin close price



Εικόνα 12. Γραφική αναπαράσταση των απομονωμένων τιμών κλεισίματος έως και ένα χρόνο προγενέστερα του σήμερα για την εκπαίδευση του μοντέλου πρόβλεψης.

5.2.6.2. Κανονικοποίηση των δεδομένων και χωρισμός

Η κανονικοποίηση των δεδομένων αποτελεί πρακτική η οποία συχνά χρησιμοποιείται για την προετοιμασία των δεδομένων για χρήση σε μεθόδους μηχανικής μάθησης. Στόχος της είναι η αλλαγή των τιμών από της στήλες στις οποίες εμπεριέχονται στην βάση δεδομένων να αξιοποιούν μια κοινή κλίμακα, χωρίς να παραμορφώνει τυχόν διαφορές στη εμβέλεια των τιμών και στην αποφυγή απώλειας αυτών.

Στη συνέχεια θα εισαχθεί ο αλγόριθμος MinMaxScaler. Με αυτόν τον τρόπο όλες οι μεταβλητές στη βάση δεδομένων μας θα μεταμορφωθούν σε τιμές με εμβέλεια [0,1], το οποίο σημαίνει ότι κάθε μεταβλητή, δηλαδή στη προκειμένη περίπτωση οι τιμές κλεισίματος της κάθε ημέρας του Bitcoin, θα έχει σαν ελάχιστη τιμή το 0 και μέγιστη το 1.

Η ιδέα πίσω από αυτή τη πρακτική είναι πως η κλίμακα και η διαφορά ανάμεσα στις τιμές της βάσης δεδομένων μπορεί να διαφέρουν ποσοτικά σε μεγάλο βαθμό, με αποτέλεσμα να μην συμμετέχουν το ίδιο στον αλγόριθμο και την εκμάθηση του και να δημιουργούν αθέμητες τάσεις. Παρακάτω γίνεται παράθεση του τύπου που αξιοποιεί η συγκεκριμένη λειτουργία.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Για την περαιτέρω απλοποίηση κατά την εκμάθηση του μοντέλου πρόβλεψης θα αφαιρεθεί προσωρινά από την βάση δεδομένων και η στήλη που περιέχει τις ημερομηνίες.

IN:

```
del closedf['Date']
scaler = MinMaxScaler(feature_range=(0, 1))
closedf = scaler.fit_transform(np.array(closedf).reshape(-1, 1))
print(closedf.shape)
```

OUT:

```
(365, 1)
```

Στη συνέχεια θα χωριστούν τα δεδομένα ώστε το 60% να αποτελεί το Training set και το υπόλοιπο 40% το Testing set και θα οριστεί το time step (πόσες προηγούμενες ημέρες θα ληφθούν υπόψη για την πρόβλεψη τιμής της επόμενης ημέρας, ή αλλιώς η μνήμη του μοντέλου) στο 15.

IN:

```
training_size=int(len(closedf)*0.60)
test_size=len(closedf)-training_size
train_data, test_data=closedf[0:training_size, :],
closedf[training_size:len(closedf), :1]
print("train_data: ", train_data.shape)
print("test_data: ", test_data.shape)
```

OUT:

```
train_data: (219, 1)
```

```
test_data: (146, 1)
```

IN:

```
def create_dataset(dataset, time_step=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-time_step-1):
        a = dataset[i:(i+time_step), 0]    ###i=0, 0,1,2,3-----99
100
        dataX.append(a)
        dataY.append(dataset[i + time_step, 0])
    return np.array(dataX), np.array(dataY)

time_step = 15
X_train, y_train = create_dataset(train_data, time_step)
X_test, y_test = create_dataset(test_data, time_step)

print("X_train: ", X_train.shape)
print("y_train: ", y_train.shape)
print("X_test: ", X_test.shape)
print("y_test", y_test.shape)
```

OUT:

```
X_train: (203, 15)
```

```
y_train: (203,)
```

```
X_test: (130, 15)
```

```
y_test (130,)
```

Τέλος θα αναδιαμορφώσουμε τα δεδομένα.

IN:

```
X_train =X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)

print("X_train: ", X_train.shape)
print("X_test: ", X_test.shape)
```

OUT:

```
X_train: (203, 15, 1)
```

```
X_test: (130, 15, 1)
```

5.2.6.3. Διαμόρφωση του μοντέλου

Για την πρόβλεψη μελλοντικών τιμών του Bitcoin θα χρησιμοποιηθεί ένα νευρωνικό δίκτυο LSTM με τη μέθοδο βελτιστοποίησης Adam, ενώ παράλληλα θα υπολογιστούν οι μετρητές απόδοσης που έχουν αναφερθεί νωρίτερα για την αξιολόγηση της αποτελεσματικότητας των προβλέψεων. Οι φορές που θα «περάσει» ο αλγόριθμος από τη βάση δεδομένων ορίστηκε στις 200, ενώ για εύλογη παρουσίαση των αποτελεσμάτων θα παρατεθούν οι αρχικές και οι τελευταίες. Κρίνεται άξιο να σημειωθεί πως κάθε υλοποίηση του αλγόριθμου μπορεί να διαφέρει σε ένα μικρό βαθμό στα εξαγόμενα αποτελέσματα και πορίσματα.

IN:

```
model=Sequential()

model.add(LSTM(10, input_shape=(None, 1), activation="relu"))

model.add(Dense(1))

model.compile(loss="mean_squared_error", optimizer="adam")

history = model.fit(X_train, y_train, validation_data=(X_test,
y_test), epochs=200, batch_size=32, verbose=1)
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(loss))

plt.plot(epochs, loss, 'r', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend(loc=0)
plt.figure()

plt.show()

# Πρόβλεψη και απόδοση
train_predict=model.predict(X_train)
test_predict=model.predict(X_test)
train_predict.shape, test_predict.shape

# Μετατροπή σε αρχικά μορφή
train_predict = scaler.inverse_transform(train_predict)
test_predict = scaler.inverse_transform(test_predict)
original_ytrain = scaler.inverse_transform(y_train.reshape(-1, 1))
original_ytest = scaler.inverse_transform(y_test.reshape(-1, 1))
```

OUT:

```
Epoch 1/200
7/7 [=====] - 1s 41ms/step - loss: 0.0546 - val_loss: 0.2682

Epoch 2/200
7/7 [=====] - 0s 7ms/step - loss: 0.0456 - val_loss: 0.2378

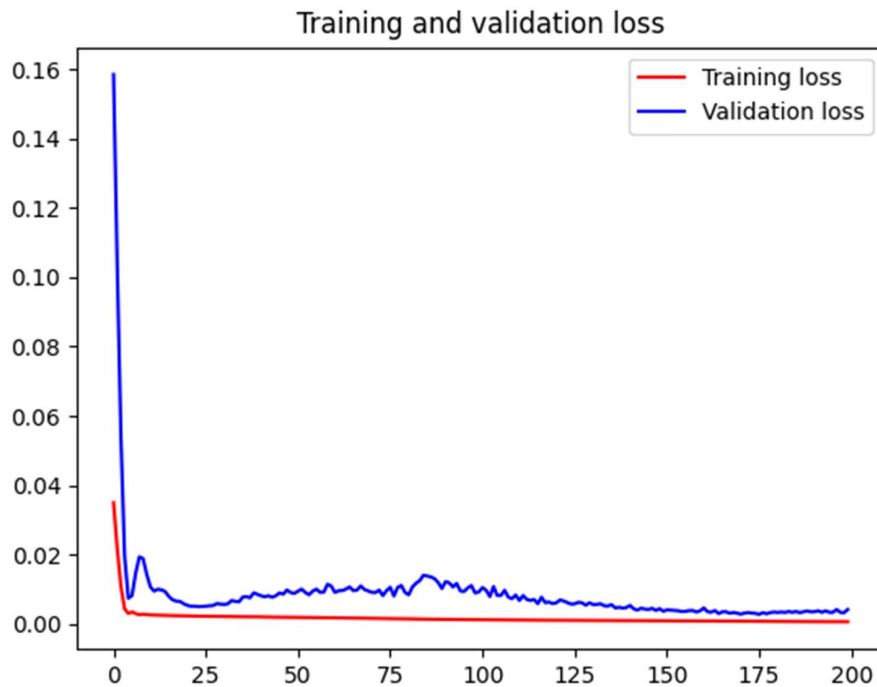
...

Epoch 199/200
7/7 [=====] - 0s 7ms/step - loss: 5.7131e-04 - val_loss: 0.0108

Epoch 200/200
7/7 [=====] - 0s 7ms/step - loss: 5.6840e-04 - val_loss: 0.0060

7/7 [=====] - 0s 1ms/step

5/5 [=====] - 0s 978us/step
```



Εικόνα 13. Το λάθος του αλγόριθμου σε κάθε επαναλημμένη προσπάθεια της εκπαίδευσης (200).

Αφού παρατηρήσουμε πως κάθε επόμενη υλοποίηση του αλγόριθμου μικραίνει το πιθανό σφάλμα ανάμεσα στην εκπαίδευση και την επαλήθευση του αλγορίθμου μπορούμε να προχωρήσουμε στους μετρητές απόδοσης του μοντέλου.

IN:

```
print("Train data RMSE: ",
      math.sqrt(mean_squared_error(original_ytrain, train_predict)))
print("Train data MSE: ", mean_squared_error(original_ytrain,
      train_predict))
print("Train data MAE: ", mean_absolute_error(original_ytrain,
      train_predict))
print("-----")
print("Test data RMSE: ",
      math.sqrt(mean_squared_error(original_ytest, test_predict)))
print("Test data MSE: ", mean_squared_error(original_ytest,
      test_predict))
print("Test data MAE: ", mean_absolute_error(original_ytest,
      test_predict))

print("Train data explained variance regression score:",
      explained_variance_score(original_ytrain, train_predict))
print("Test data explained variance regression score:",
      explained_variance_score(original_ytest, test_predict))

print("Train data R2 score:", r2_score(original_ytrain,
      train_predict))
print("Test data R2 score:", r2_score(original_ytest, test_predict))

print("Train data MGD: ", mean_gamma_deviance(original_ytrain,
      train_predict))
print("Test data MGD: ", mean_gamma_deviance(original_ytest,
      test_predict))
print("-----")
print("Train data MPD: ", mean_poisson_deviance(original_ytrain,
      train_predict))
print("Test data MPD: ", mean_poisson_deviance(original_ytest,
      test_predict))
```

OUT:

```
Train data RMSE: 636.260145114128
Train data MSE: 404826.9722606512
Train data MAE: 457.48197932450745
-----
Test data RMSE: 2066.033919720117
Test data MSE: 4268496.1574340705
Test data MAE: 1418.5054987961537
Train data explained variance regression score: 0.9429071614299025
Test data explained variance regression score: 0.9117803868669391
Train data R2 score: 0.9410441104324795
Test data R2 score: 0.9069492906533604
Train data MGD: 0.0005656687003065834
Test data MGD: 0.002261923117739336
-----
Train data MPD: 15.059653053782036
Test data MPD: 97.22769232660028
```

Από τις αποδόσεις των μετρητικών απόδοσης βγαίνει το συμπέρασμα πως το μοντέλο έχει ικανοποιητική απόδοση στην διαδικασία εκμάθησης, παρόλα αυτά η ακρίβεια στις προβλεπόμενες τιμές δεν κρίνεται επαρκή για κάποιον ο οποίος ασχολείται επενδυτικά με το Bitcoin, αφού αυτός θα αναζητούσε μια ακρίβεια πρόβλεψης κοντά στο 100% με πολύ μικρές αποκλίσεις της τάξεως των διψήφιων χρηματικών τιμών. Αντιθέτως, για τον μέσο χρήστη το μοντέλο κρίνεται επαρκές ενώ η ικανότητα του μοντέλου να προβλέπει τάσεις κρίνεται αξιοποιήσιμη.

5.2.6.4. Αποτελέσματα

Τέλος, αρχικά μέσω του κώδικα θα επιβεβαιώσουμε την ποσότητα των δεδομένων, ενώ θα γίνει γραφική αναπαράσταση των αποτελεσμάτων του κώδικα, αφού πρώτα επαναφέρουμε τα δεδομένα στην αρχική τους μορφή.

IN:

```
look_back=time_step
trainPredictPlot = np.empty_like(closedf)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[look_back:len(train_predict)+look_back, :] =
train_predict
print("Train predicted data: ", trainPredictPlot.shape)

#Μεταβολή μορφής
testPredictPlot = np.empty_like(closedf)
testPredictPlot[:, :] = np.nan
testPredictPlot[len(train_predict)+(look_back*2)+1:len(closedf)-1, :]
= test_predict
print("Test predicted data: ", testPredictPlot.shape)

names = cycle(['Original close price', 'Train predicted close price',
'Test predicted close price'])

plotdf = pd.DataFrame({'date': close_stock['Date'],
                       'original_close': close_stock['Close'],
                       'train_predicted_close':
trainPredictPlot.reshape(1, -1)[0].tolist(),
                       'test_predicted_close':
testPredictPlot.reshape(1, -1)[0].tolist()})

fig = px.line(plotdf, x=plotdf['date'], y=[plotdf['original_close'],
plotdf['train_predicted_close'],
plotdf['test_predicted_close']],
              labels={'value': 'Bitcoin price', 'date': 'Date'})
fig.update_layout(title_text='Comparison between original close price
vs predicted close price',
                  plot_bgcolor='white', font_size=15,
                  font_color='black', legend_title_text='Close Price')
fig.for_each_trace(lambda t: t.update(name=next(names)))

fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()
```

OUT:

Train predicted data: (365, 1)

Test predicted data: (365, 1)



Εικόνα 14. Γραφική αναπαράσταση της αληθινής τιμής κλεισίματος του Bitcoin για την εξεταζόμενη περίοδο, της τιμής πρόβλεψης κατά την εκπαίδευση και την τιμή πρόβλεψης ως έξοδο του κώδικα.

Αξιολογώντας την γραφική παράσταση γίνεται αντιληπτό πως το μοντέλο κρίνεται επαρκώς ικανό να προβλέψει τάσεις και λιγότερο ικανό στην ακρίβεια της πρόβλεψης, κάτι το οποίο παρατηρήθηκε έντονα μέσα από τις πολλαπλές εκτελέσεις του κώδικα. Αξίζει να σημειωθεί πως σε ορισμένες εκτελέσεις είτε λόγω περιορισμών των βιβλιοθηκών που χρησιμοποιήθηκαν είτε λόγω ασυμβατότητας της έκδοσης των βιβλιοθηκών σε ορισμένες ενέργειες με την έκδοση του κώδικα Pyhton, αλλά κυρίως λόγω της στοχαστικής φύσης του αλγορίθμου κατά τη διαδικασία της αποτίμησης, η ακρίβεια στην πρόβλεψη τιμής ήταν έως και 20% περίπου εσφαλμένη. Πάραυτα, η αρχική παρατήρηση της αξιοπιστίας του μοντέλου ως προς την πρόβλεψη της τάση της τιμής, παρέμενε αληθής καθ' όλες τις δοκιμές οι οποίες εκτελέστηκαν. Άλλωστε ο ακριβής προσδιορισμός μιας μελλοντικής τιμής ενός κρυπτονομίσματος, με δεδομένα την προγραμματιστική γνώση και την απαραίτητα υπολογιστική ισχύ, κρίνεται πιθανώς ακατόρθωτος.

6. Συμπεράσματα

Από την υλοποίηση του αλγόριθμου και τα εξαγόμενα αποτελέσματα, τα οποία έχουν παραστεί αναλυτικά στις προηγούμενες ενότητες, θα μπορούσε το πείραμα να χαρακτηριστεί ως επιτυχές υπό ορισμένες συνθήκες. Η επιλογή του συγκεκριμένου κρυπτονομίσματος έγινε με βάση την εδραίωσή του στην αγορά, και ως μέσο χρηματικών συναλλαγών, αλλά και ως βάση υλοποίησης διαφόρων εργασιών. Αυτό σημαίνει πως με εξαίρεση την αρχική του πορεία, πλέον θεωρείται ένα κρυπτονόμισμα το οποίο αν και δεν μένει σταθερό στην αξία του, η διακυμάνσεις του είναι σε θεωρητικά λογικά πλαίσια. Με δυσκολία δηλαδή θα πολλαπλασιαστεί ξανά 30 φορές σε λίγα χρόνια. Ακόμα, οι υποδομές και οι χρήσεις του έχουν γερές βάσεις, σε αντίθεση με κάποιο Meme Coin (κρυπτονόμισμα που δεν αποσκοπεί κάπου, δεν έχει χρήσεις και έχει δημιουργηθεί για λόγους ψυχαγωγίας), όπως το Dogecoin. Έτσι μπορούμε να αποκλείσουμε φαινόμενα που έχουν συμβεί στο παρελθόν και έχουν μεταβάλει δραματικά σε λίγα λεπτά την αξία ενός κρυπτονομίσματος όπως μια δημοσίευση ή ανάρτηση ενός διάσημου προσώπου. Όλα αυτά, συμβάλλουν σε καλύτερες και πιο εύστοχες προβλέψεις μέσω του αλγόριθμου που δημιουργήθηκε, αποτέλεσμα που δεν θα ήταν το ίδιο εάν επιλέγαμε την ίδια ανάλυση και πρόβλεψη σε άλλα κρυπτονομίσματα. Παρόλα αυτά, ο συγκεκριμένος κώδικας μπορεί εύστοχα να χρησιμοποιηθεί και σε άλλα κρυπτονομίσματα ανάλογης σοβαρότητας όπως το Ethereum.

Όπως μπορεί να επιβεβαιωθεί και από αντίστοιχες μελέτες, η μέθοδος πρόβλεψης με τα LSTM νευρωνικά δίκτυα παρουσιάζει μικρά σφάλματα στις προβλέψεις, με εξαίρεση ορισμένες τοπολογίες. Αξίζει να σημειωθεί πως σε μια ιδανική περίπτωση, καθώς ο κώδικας βγάζει κάθε φορά που υλοποιείται ελαφρώς διαφορετικά αποτελέσματα είτε με μικρότερα σφάλματα είτε με μεγαλύτερα, θα γινόταν μια αξιολόγηση και συγχώνευση των αποτελεσμάτων αφού ο κώδικας ολοκληρωθεί πολυάριθμες φορές. Κάτι τέτοιο όμως απαιτεί μεγάλη υπολογιστική ισχύ και σε ένα ιδανικό σενάριο ο κώδικας θα έτρεχε διαρκώς με συνεχή ανατροφοδότηση και αξιολόγηση των δεδομένων.

Συμπερασματικά, κρίνοντας από τους μετρητές απόδοσης του μοντέλου αλλά και την τελική αναπαράσταση την προβλεπόμενης τιμής σε σχέση με αυτή της πραγματικής, το εγχείρημα της πρόβλεψης τιμών κρυπτονομισμάτων με χρήση νευρωνικών δικτύων μπορεί να κριθεί ως επιτυχές. Ιδανικά κάποιος που θα ήθελε να αξιοποιήσει αυτό ή και ανάλογα μοντέλο για πρόβλεψη και επένδυση σε κρυπτονομίσματα, θα ήταν εύλογο να τα αξιοποιήσει ως μια γενική κατεύθυνση σε τάσεις ανόδου και καθόδου των τιμών, αφού το μοντέλο φαίνεται να είναι αρκετά ακριβές σε αυτό τον τομέα.

7. Παράρτημα

Στο παράρτημα θα ακολουθήσει η παράθεση ολόκληρου το κώδικα που γράφτηκε στην γλώσσα προγραμματισμού Python για αξιοποίηση και πιθανή εκμετάλλευση ή βελτιστοποίηση του μοντέλου.

```
1 import os
2 import pandas as pd
3 pd.set_option('display.max_columns', None)
4 import numpy as np
5 import math
6 import datetime as dt
7 import matplotlib.pyplot as plt
8
9 from sklearn.metrics import mean_squared_error,
  mean_absolute_error, explained_variance_score,
  r2_score
10 from sklearn.metrics import mean_poisson_deviance,
  mean_gamma_deviance, accuracy_score
11 from sklearn.preprocessing import MinMaxScaler
12
13 import tensorflow as tf
14 from tensorflow.keras.models import Sequential
15 from tensorflow.keras.layers import Dense, Dropout
16 from tensorflow.keras.layers import LSTM
17
18 import matplotlib.pyplot as plt
19 from itertools import cycle
20 import plotly.graph_objects as go
21 import plotly.express as px
22 from plotly.subplots import make_subplots
23
24 maindf = pd.read_csv(r"C:\Users\jimka\Desktop\opa\45\
  123.csv")
25 print('Total number of days present in the dataset: '
  , maindf.shape[0])
26 print('Total number of fields present in the dataset
  : ', maindf.shape[1])
27
28 maindf.shape
29 print(maindf.shape)
30
31 maindf.head()
32 print(maindf.head)
33 maindf.tail()
34 print(maindf.tail)
35
```

```

36 maindf.info()
37 print(maindf.info)
38 maindf.describe()
39 print(maindf.describe)
40
41 print('Null Values:', maindf.isnull().values.sum())
42 print('NA values:', maindf.isnull().values.any())
43
44 # Printing the start date and End date of the dataset
45
46 sd = maindf.iloc[0][0]
47 ed = maindf.iloc[-1][0]
48
49 print('Starting Date', sd)
50 print('Ending Date', ed)
51
52 maindf['Date'] = pd.to_datetime(maindf['Date'],
    format='%Y-%m-%d')
53
54 y_2014 = maindf.loc[(maindf['Date'] >= '2014-09-17')
55                    & (maindf['Date'] < '2014-12-31')]
56
57 y_2014.drop(y_2014[['Adj Close', 'Volume']], axis=1)
58 print(y_2014)
59
60 monthwise = y_2014.groupby(y_2014['Date'].dt.strftime
    ('%B'))[['Open', 'Close']].mean()
61 new_order = ['January', 'February', 'March', 'April'
    , 'May', 'June', 'July', 'August',
62              'September', 'October', 'November', '
    December']
63 monthwise = monthwise.reindex(new_order, axis=0)
64 monthwise
65 print(monthwise)
66
67 #2014 open and close price
68 fig = go.Figure()
69
70 fig.add_trace(go.Bar(
71     x=monthwise.index,
72     y=monthwise['Open'],

```

```

73     name='Bitcoin Open Price',
74     marker_color='crimson'
75 ))
76 fig.add_trace(go.Bar(
77     x=monthwise.index,
78     y=monthwise['Close'],
79     name='Bitcoin Close Price',
80     marker_color='lightsalmon'
81 ))
82
83 fig.update_layout(barmode='group', xaxis_tickangle=-
45,
84                     title='Monthwise comparison
between Bitcoin open and close price')
85 fig.show()
86
87 #2014 high low price
88 y_2014.groupby(y_2014['Date'].dt.strftime('%B'))['
Low'].min()
89 monthwise_high = y_2014.groupby(maindf['Date'].dt.
strftime('%B'))['High'].max()
90 monthwise_high = monthwise_high.reindex(new_order,
axis=0)
91
92 monthwise_low = y_2014.groupby(y_2014['Date'].dt.
strftime('%B'))['Low'].min()
93 monthwise_low = monthwise_low.reindex(new_order,
axis=0)
94
95 fig = go.Figure()
96 fig.add_trace(go.Bar(
97     x=monthwise_high.index,
98     y=monthwise_high,
99     name='Bitcoin high Price',
100    marker_color='rgb(0, 153, 204)'
101 ))
102 fig.add_trace(go.Bar(
103     x=monthwise_low.index,
104     y=monthwise_low,
105     name='Bitcoin low Price',
106     marker_color='rgb(255, 128, 0)'

```

```

107 ))
108
109 fig.update_layout(barmode='group',
110                   title=' Monthwise High and Low
    Bitcoin price')
111 fig.show()
112
113
114 #2014 stock analysis chart
115 names = cycle(['Bitcoin Open Price', 'Bitcoin Close
    Price', 'Bitcoin High Price', 'Bitcoin Low Price'])
116
117 fig = px.line(y_2014, x=y_2014.Date, y=[y_2014['Open
    '], y_2014['Close'],
118                                     y_2014['
    High'], y_2014['Low']],
119 labels={'Date': 'Date', 'value': 'Bitcoin value'})
120 fig.update_layout(title_text='Bitcoin analysis chart
    ', font_size=15, font_color='black',
121                   legend_title_text='Bitcoin
    Parameters')
122 fig.for_each_trace(lambda t: t.update(name=next(
    names)))
123 fig.update_xaxes(showgrid=False)
124 fig.update_yaxes(showgrid=False)
125
126 fig.show()
127
128 #analysis 2014-2023
129 maindf['Date'] = pd.to_datetime(maindf['Date'],
    format='%Y-%m-%d')
130
131 y_overall = maindf.loc[(maindf['Date'] >= '2014-09-
    17')
132                       & (maindf['Date'] <= '2024-01-
    18')]
133
134 y_overall.drop(y_overall[['Adj Close', 'Volume']],
    axis=1)
135
136 print(maindf)

```

```

137
138 monthwise= y_overall.groupby(y_overall['Date'].dt.
    strftime('%B'))[['Open', 'Close']].mean()
139 new_order = ['January', 'February', 'March', 'April'
    , 'May', 'June', 'July', 'August',
140             'September', 'October', 'November', '
    December']
141 monthwise = monthwise.reindex(new_order, axis=0)
142 monthwise
143
144 #all chart
145 names = cycle(['Bitcoin Open Price', 'Bitcoin Close
    Price', 'Bitcoin High Price', 'Bitcoin Low Price'])
146
147 fig = px.line(y_overall, x=y_overall.Date, y=[
    y_overall['Open'], y_overall['Close'],
148                                                     y_overall[
    'High'], y_overall['Low']],
149             labels={'Date': 'Date', 'value': '
    Bitcoin value'})
150 fig.update_layout(title_text='Bitcoin analysis chart
    ', font_size=15, font_color='black',
151                   legend_title_text='Bitcoin
    Parameters')
152 fig.for_each_trace(lambda t: t.update(name = next(
    names)))
153 fig.update_xaxes(showgrid=False)
154 fig.update_yaxes(showgrid=False)
155
156 fig.show()
157
158 #lstm model
159 #all Close Price
160 closedf = maindf[['Date', 'Close']]
161 print("Shape of close dataframe:", closedf.shape)
162 print(closedf)
163
164 fig = px.line(closedf, x=closedf.Date, y=closedf.
    Close, labels={'date': 'Date', 'close': 'Close Stock
    '})
165 fig.update_traces(marker_line_width=2, opacity=0.8,

```



```

165 marker_line_color='orange')
166 fig.update_layout(title_text='Whole period of
    timeframe of Bitcoin close price 2014-2024',
    plot_bgcolor='white',
167             font_size=15, font_color='black')
168 fig.update_xaxes(showgrid=False)
169 fig.update_yaxes(showgrid=False)
170 fig.show()
171
172 closedf = closedf[closedf['Date'] > '2023-01-18']
173 close_stock = closedf.copy()
174 print("Total data for prediction: ", closedf.shape[0
    ])
175
176 print(closedf)
177
178
179 fig = px.line(closedf, x=closedf.Date, y=closedf.
    Close, labels={'date': 'Date', 'close': 'Close Stock
    '})
180 fig.update_traces(marker_line_width=2, opacity=0.8,
    marker_line_color='orange')
181 fig.update_layout(title_text='Considered period to
    predict Bitcoin close price',
182             plot_bgcolor='white', font_size=15
    , font_color='black')
183 fig.update_xaxes(showgrid=False)
184 fig.update_yaxes(showgrid=False)
185 fig.show()
186
187 # deleting date column and normalizing using MinMax
    Scaler
188
189 del closedf['Date']
190 scaler = MinMaxScaler(feature_range=(0, 1))
191 closedf = scaler.fit_transform(np.array(closedf).
    reshape(-1, 1))
192 print(closedf.shape)
193
194 #training set 60% and 40% testing set
195

```

```

196 training_size=int(len(closedf)*0.60)
197 test_size=len(closedf)-training_size
198 train_data, test_data=closedf[0:training_size, :],
    closedf[training_size:len(closedf), :1]
199 print("train_data: ", train_data.shape)
200 print("test_data: ", test_data.shape)
201
202 # convert an array of values into a dataset matrix
203
204 def create_dataset(dataset, time_step=1):
205     dataX, dataY = [], []
206     for i in range(len(dataset)-time_step-1):
207         a = dataset[i:(i+time_step), 0]    ###i=0, 0,
    1,2,3-----99    100
208         dataX.append(a)
209         dataY.append(dataset[i + time_step, 0])
210     return np.array(dataX), np.array(dataY)
211
212 time_step = 15
213 X_train, y_train = create_dataset(train_data,
    time_step)
214 X_test, y_test = create_dataset(test_data, time_step
    )
215
216 print("X_train: ", X_train.shape)
217 print("y_train: ", y_train.shape)
218 print("X_test: ", X_test.shape)
219 print("y_test", y_test.shape)
220
221 # reshape input to be [samples, time steps, features
    ] which is required for LSTM
222 X_train =X_train.reshape(X_train.shape[0], X_train.
    shape[1], 1)
223 X_test = X_test.reshape(X_test.shape[0], X_test.
    shape[1], 1)
224
225 print("X_train: ", X_train.shape)
226 print("X_test: ", X_test.shape)
227
228 model=Sequential()
229

```

```

230 model.add(LSTM(10, input_shape=(None, 1), activation
    ="relu"))
231
232 model.add(Dense(1))
233
234 model.compile(loss="mean_squared_error", optimizer="
    adam")
235
236 history = model.fit(X_train, y_train,
    validation_data=(X_test, y_test), epochs=200,
    batch_size=32, verbose=1)
237
238 import matplotlib.pyplot as plt
239
240 loss = history.history['loss']
241 val_loss = history.history['val_loss']
242
243 epochs = range(len(loss))
244
245 plt.plot(epochs, loss, 'r', label='Training loss')
246 plt.plot(epochs, val_loss, 'b', label='Validation
    loss')
247 plt.title('Training and validation loss')
248 plt.legend(loc=0)
249 plt.figure()
250
251
252 plt.show()
253
254 #prediction and check performance metrics
255 train_predict=model.predict(X_train)
256 test_predict=model.predict(X_test)
257 train_predict.shape, test_predict.shape
258
259 # Transform back to original form
260
261 train_predict = scaler.inverse_transform(
    train_predict)
262 test_predict = scaler.inverse_transform(test_predict
    )
263 original_ytrain = scaler.inverse_transform(y_train.

```



```

263 reshape(-1, 1))
264 original_ytest = scaler.inverse_transform(y_test.
    reshape(-1, 1))
265
266 # Evaluation metrics RMSE and MAE
267 print("Train data RMSE: ", math.sqrt(
    mean_squared_error(original_ytrain, train_predict)))
268 print("Train data MSE: ", mean_squared_error(
    original_ytrain, train_predict))
269 print("Train data MAE: ", mean_absolute_error(
    original_ytrain, train_predict))
270 print(
    "-----")
271 print("Test data RMSE: ", math.sqrt(
    mean_squared_error(original_ytest, test_predict)))
272 print("Test data MSE: ", mean_squared_error(
    original_ytest, test_predict))
273 print("Test data MAE: ", mean_absolute_error(
    original_ytest, test_predict))
274
275 print("Train data explained variance regression
    score:",
276       explained_variance_score(original_ytrain,
    train_predict))
277 print("Test data explained variance regression score
    :",
278       explained_variance_score(original_ytest,
    test_predict))
279
280 print("Train data R2 score:", r2_score(
    original_ytrain, train_predict))
281 print("Test data R2 score:", r2_score(original_ytest
    , test_predict))
282
283 print("Train data MGD: ", mean_gamma_deviance(
    original_ytrain, train_predict))
284 print("Test data MGD: ", mean_gamma_deviance(
    original_ytest, test_predict))
285 print(
    "-----")

```

```

285 -----")
286 print("Train data MPD: ", mean_poisson_deviance(
    original_ytrain, train_predict))
287 print("Test data MPD: ", mean_poisson_deviance(
    original_ytest, test_predict))
288
289
290 look_back=time_step
291 trainPredictPlot = np.empty_like(closedf)
292 trainPredictPlot[:, :] = np.nan
293 trainPredictPlot[look_back:len(train_predict)+
    look_back, :] = train_predict
294 print("Train predicted data: ", trainPredictPlot.
    shape)
295
296 # shift test predictions for plotting
297 testPredictPlot = np.empty_like(closedf)
298 testPredictPlot[:, :] = np.nan
299 testPredictPlot[len(train_predict)+(look_back*2)+1:
    len(closedf)-1, :] = test_predict
300 print("Test predicted data: ", testPredictPlot.shape
    )
301 names = cycle(['Original close price', 'Train
    predicted close price', 'Test predicted close price'
    ])
302
303
304 plotdf = pd.DataFrame({'date': close_stock['Date'],
305                       'original_close': close_stock
    ['Close'],
306                       'train_predicted_close':
    trainPredictPlot.reshape(1, -1)[0].tolist(),
307                       'test_predicted_close':
    testPredictPlot.reshape(1, -1)[0].tolist()})
308
309 fig = px.line(plotdf, x=plotdf['date'], y=[plotdf['
    original_close'], plotdf['train_predicted_close'],
310                                           plotdf['
    test_predicted_close']],
311             labels={'value': 'Bitcoin price', '
    date': 'Date'})

```

```
312 fig.update_layout(title_text='Comparison between  
    original close price vs predicted close price',  
313                      plot_bgcolor='white', font_size=15  
    , font_color='black', legend_title_text='Close Price  
    ')  
314 fig.for_each_trace(lambda t: t.update(name=next(  
    names)))  
315  
316 fig.update_xaxes(showgrid=False)  
317 fig.update_yaxes(showgrid=False)  
318 fig.show()
```

8. Βιβλιογραφία

- [1] Γ. Ζαχόπουλος, Πρόβλεψη τροχιάς αεροσκάφους με την χρήση νευρωνικών δικτύων LSTM, Μεταπτυχιακή διατριβή, Πανεπιστήμιο Πειραιώς, 2018
- [2] Kaspersky Resource Center, What is cryptocurrency and how does it work? , www.kaspersky.com
- [3] Sahar, Arshi., Li, Zhang., Rebecca, Strachan. (2019). Prediction Using LSTM Networks. 1-8. Available from: 10.1109/IJCNN.2019.8852206
- [4] Thushan Ganegedara, Stock Market Predictions with LSTM in Python, 2020, www.datacamp.com
- [5] Alan Coyne, Super Simple Neural Network for Bitcoin Price Prediction in Python, 2022, www.pyhton.plainenglish.io
- [6] Jason Brownlee, Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras, 2022, machinelearningmastery.com
- [7] Ι. Καμέτας, Κρυπτονομίσματα, Διπλωματική εργασία, Πανεπιστήμιο Αιγαίου, 2018
- [8] Σ. Δούμας, Κρυπτονομίσματα και διαχείριση κινδύνου κυβερνοχώρου, Διπλωματική εργασία, Πανεπιστήμιο Μακεδονίας, Θεσσαλονίκη, 2019.
- [9] Simanta Shekhar Sarmah, Understanding Blockchain Technology, Computer Science and Engineering, USA, 2018
- [10] Ε. Στούμπος, Μεταφορά Γνώσης στη Βαθιά Μάθηση από Επιτροπές Νευρωνικών Δικτύων, Πτυχιακή εργασία, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, 2019
- [11] Συμεωνίδης, Π. (2023). Μηχανική Μάθηση με Νευρωνικά Δίκτυα [Κεφάλαιο]. Στο Συμεωνίδης, Π. 2023. Ευφυή Συστήματα Συστάσεων [Μεταπτυχιακό εγχειρίδιο]. Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις. <https://hdl.handle.net/11419/9578>
- [12] Μ.Ε Κώνστα, Κρυπτογραφικά νομίσματα: τεχνικές αρχές και τρόπος λειτουργίας: η περίπτωση του Bitcoin, Πτυχιακή Εργασία, 2016
- [13] Α.Ν. Σωτηρίου, An LSTM approach to short-term load forecasting, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, 2023
- [14] Ι. Ε. Λιβιέρης, Μη Γραμμικές Μέθοδοι Συζυγών Κλίσεων για Βελτιστοποίηση και Εκπαίδευση Νευρωνικών Δικτύων, Διδακτορική Διατριβή, Πανεπιστήμιο Πατρών, 2012
- [15] Η. Παρίσης, Πρόβλεψη Τιμών Χρονοσειρών με Χρήση Αλγορίθμων Μηχανικής Μάθησης, Διπλωματική Εργασία, Ελληνικό Ανοικτό Πανεπιστήμιο, 2019
- [16] Ι.Ε. Livieris, Ε. Pintelas, Ρ. Pintelas, "A CNN–LSTM model for gold price time-series forecasting", 2020

[17] Συμεωνίδης, Π., & Γούναρης, Α. (2015). Χρονοσειρές [Κεφάλαιο]. Στο Συμεωνίδης, Π., & Γούναρης, Α. 2015. Βάσεις, αποθήκες και εξόρυξη δεδομένων με τον SQL Server [Εργαστηριακός Οδηγός]. Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις