



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Εισάγετε τον Τίτλο της Εργασίας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημήτρης Κ. Κυριάκου

Επιβλέπων : Όνομα, Αρχικό Πατρώνυμου, Επίθετο
Ιδιότητα Επιβλέποντα

Αθήνα, Ιούλιος 2022



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Εισάγετε τον Τίτλο της Εργασίας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημήτρης Κ. Κυριάκου

Επιβλέπων : Όνομα, Αρχικό Πατρώνυμου, Επίθετο

Ιδιότητα Επιβλέποντα

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 31^η Μήνα Έτος.

.....
Ον/μο Μέλος Δ.Ε.Π
Ιδιότητα Μέλους Δ.Ε.Π

.....
Ον/μο Μέλος Δ.Ε.Π
Ιδιότητα Μέλους Δ.Ε.Π

.....
Ον/μο Μέλος Δ.Ε.Π
Ιδιότητα Μέλους Δ.Ε.Π

Αθήνα, Μήνας Έτος

.....
Δημήτρης Κ. Κυριάκου

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Δημήτρης Κυριάκου 2022.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Λέξεις-κλειδιά

Abstract

Keywords

Ευχαριστίες

ΕΙΣΑΓΩΓΗ	9
ΣΚΟΠΟΣ	9
ΔΟΜΗ ΕΡΓΑΣΙΑΣ	10
A. ΕΝΟΤΗΤΑ ΘΕΩΡΙΑΣ.....	12
A.1. ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ.....	13
A.1.1. Συνάρτηση Κατακερματισμού (Hash function).....	13
A.1.2. Ασύμμετρη Κρυπτογραφία (Asymmetric Cryptography).....	14
A.2. BLOCKCHAIN	16
A.2.1. Bitcoin.....	16
A.2.2. Αρχές Blockchain	17
A.2.2.I. Συναλλαγή.....	17
A.2.2.II. Δημιουργία Block	18
A.2.2.III. Πρωτόκολλο Συναίνεσης (Consensus Protocol).....	19
A.2.2.III.α. Proof of Work - PoW.....	20
A.2.2.III.β. Proof of Stake - PoS.....	21
A.2.2.III.γ. Proof of Learning – PoLe.....	22
A.2.3. Ethereum.....	24
A.2.3.I. Ψηφιακή Μηχανή Ethereum (Ethereum Virtual Machine – EVM).....	24
A.2.3.II. Έξυπνα συμβόλαια (Smart Contracts)	25
A.2.3.III. Λογαριασμοί (Accounts)	25
A.2.3.IV. Dapps	26
A.3. ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ.....	28
A.3.1. Επιβλεπόμενη Μάθηση (Supervised Learning).....	28
A.3.2. Επεξεργασία Φυσικής Γλώσσας (NLP – Natural Language Processing).....	29
A.3.3. BERT - Bidirectional Encoder Representations from Transformers	30
A.4. ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ ΚΑΙ BLOCKCHAIN.....	31
A.4.1. Υφιστάμενες δουλειές.....	31
A.4.1.I. Decentralized & Collaborative AI on Blockchain	31
A.4.1.II. Decentralized and Distributed Machine Learning Model Training with Actors	32
A.4.1.III. Proof of Learning (PoLe).....	33
B. ΕΝΟΤΗΤΑ ΠΡΟΤΕΙΝΟΜΕΝΗΣ ΛΥΣΗΣ	34
B.1. ΣΧΕΔΙΑΣΜΟΣ	35
B.1.1. Κεντρική Ιδέα	35
B.1.2. Μηχανισμός Κινήτρου (Incentive Mechanism).....	37
B.1.3. Μοντέλο Νευρωνικού Δικτύου	39

B.1.3.I.	Σύνολα Δεδομένων (Datasets)	40
B.1.3.I.α.	Twitter Tweets Sentiment Dataset.....	40
B.1.3.I.β.	IMDB Movie Ratings Sentiment Analysis	40
B.1.3.I.γ.	Emotions Sentiment Analysis	40
B.1.3.II.	Αξιολόγηση μοντέλων ΝΔ.....	41
B.2.	ΥΛΟΠΟΙΗΣΗ.....	42
B.2.1.	Εργαλεία.....	42
B.2.1.I.	Remix IDE	42
B.2.1.II.	Truffle	42
B.2.1.III.	Ganache.....	42
B.2.1.IV.	Metamask.....	43
B.2.1.V.	React.js	43
B.2.1.VI.	Next.js	43
B.2.1.VII.	Flask	43
B.2.1.VIII.	Google Colab.....	44
B.2.1.IX.	Kaggle	44
B.2.1.X.	TensorFlow	44
B.2.1.XI.	Transformers	45
B.2.1.XII.	Ngrok	45
B.2.1.XIII.	GitLab.....	45
B.2.1.XIV.	Visual Paradigm.....	45
B.2.2.	Δομή Υλοποίησης.....	46
B.2.2.I.	Backend 1 – Blockchain.....	46
B.2.2.II.	Backend 2 – Διακομιστής (server) εκπαίδευσης ΝΔ	53
B.2.2.III.	Frontend	56
B.2.2.III.α.	Αρχική Σελίδα.....	56
B.2.2.III.β.	Σελίδα Request.....	57
B.2.2.III.γ.	Σελίδα Submit	58
B.2.2.III.δ.	Σελίδα Review	58
B.2.2.III.ε.	Σελίδα Predict.....	59
B.2.2.III.στ.	Σελίδα Train	60
B.2.2.III.ζ.	Σελίδα Evaluation.....	60
B.2.3.	Εγκατάσταση	62
B.2.3.I.	Αρχική Εκπαίδευση	62
B.2.3.II.	Οργάνωση Ganache	63
B.2.3.III.	Οργάνωση Frontend.....	64
B.2.3.IV.	Οργάνωση Metamask	65

B.2.3.V.	Οργάνωση Διακομιστή Flask στο Google Colab.....	67
B.2.3.VI.	Οργάνωση Διακομιστή Flask τοπικά.....	68
B.2.3.VII.	Επανεκκίνηση DEMOS.....	68
B.3.	ΑΠΟΤΕΛΕΣΜΑΤΑ & ΣΥΜΠΕΡΑΣΜΑΤΑ.....	69
B.4.	ΠΕΡΙΟΡΙΣΜΟΙ & ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΒΕΛΤΙΩΣΗ.....	71
	ΒΙΒΛΙΟΓΡΑΦΙΑ	72

ΕΙΣΑΓΩΓΗ

Οι τεχνολογίες Μηχανικής Μάθησης και Blockchain αποτελούν αυτή τη στιγμή δύο από τις πιο επίκαιρες τεχνολογίες της επιστήμης των υπολογιστών. Παρόλο που οι δύο τεχνολογίες έχουν μελετηθεί και αναπτυχθεί σημαντικά (ειδικά η Μηχανική Μάθηση), ο συνδυασμός των δύο είναι κάτι που ξεκίνησε μόλις στα τέλη της προηγούμενης δεκαετίας. Ωστόσο, φαίνεται να είναι κάτι που αναπτύσσεται συνεχώς με νέες προτάσεις και εφαρμογές.

Από τη μία, η Μηχανική Μάθηση προϋποθέτει την εκτέλεση χρονοβόρων και υπολογιστικά απαιτητικών πράξεων και διαδικασιών. Από την άλλη, το Blockchain προσφέρει τη δυνατότητα αποκέντρωσης και διαμοιρασμού της εργασίας σε πολλές υπολογιστικές μονάδες[1]. Παράλληλα, η τεχνολογία του Blockchain παράγει τεράστιο όγκο δεδομένων ενώ η Μηχανική Μάθηση χρησιμοποιεί πολλά δεδομένα. Έτσι μπορούν να χρησιμοποιηθούν τα δεδομένα για να δημιουργηθούν σύνολα δεδομένων εκπαίδευσης. Επίσης, για τη Μηχανική Μάθηση γενικά απαιτούνται ποιοτικά δεδομένα χωρίς παραλείψεις, λάθη και επαναλήψεις. Από το Blockchain παράγονται (συνήθως) δεδομένα που έχουν να κάνουν με συναλλαγές και τα δεδομένα αυτά περνούν από πολλαπλούς ποιοτικούς ελέγχους[2].

Με αφορμή τον συνδυασμό των δύο τεχνολογιών σχεδιάστηκε και δημιουργήθηκε μια αποκεντρωμένη εφαρμογή. Η εφαρμογή ονομάστηκε **DEMOS (Distributedly Enhanced Machine learning Optimization System)** ή αλλιώς **Αποκεντρωμένα Ενισχυμένο Σύστημα Βελτιστοποίησης Μηχανικής Μάθησης**. Το όνομα DEMOS (Δήμος) επιλέχθηκε γιατί συμβολίζει τη σημασία που έχει για την τεχνολογία του Blockchain η δημιουργία μίας κοινότητας που αποτελείται από πολλούς κόμβους ίσων δικαιωμάτων και υποχρεώσεων και διοικείται βάσει της πλειοψηφίας και όχι από κάποιο κεντρικό σύστημα διαχείρισης.

ΣΚΟΠΟΣ

Η παρούσα διπλωματική εργασία εντάσσεται στο πλαίσιο συνδυασμού των τεχνολογιών της Μηχανικής Μάθησης και του Blockchain. Η αφορμή ήταν η διερεύνηση των ευκαιριών που μπορεί να προσφέρει αυτό το καινοτόμο πλαίσιο που βρίσκεται ακόμα σε πρώιμη φάση. Το πρόβλημα που καλείται να λύσει αυτή η εργασία είναι η αξιοποίηση της τεχνολογίας Blockchain ώστε να προσφέρει ουσιαστικά στη βελτίωση της επίδοσης της Μηχανικής Μάθησης.

Σκοπός ήταν η μελέτη των δύο τεχνολογιών και του συνδυασμού τους, με επιστέγασμα τη δημιουργία μιας αποκεντρωμένης εφαρμογής που θα αξιοποιεί συστήματα Blockchain για την εκπαίδευση και βελτίωση μοντέλων μηχανικής μάθησης.

Η τελική αποκεντρωμένη εφαρμογή που προτείνεται ονομάζεται DEMOS. Το DEMOS χρησιμοποιεί ένα δίκτυο Blockchain ώστε να παρέχει διαρκή βελτίωση σε μοντέλα Νευρωνικών Δικτύων με το συνεχή εμπλουτισμό του συνόλου δεδομένων εκπαίδευσης. Το μοντέλο Νευρωνικού Δικτύου που χρησιμοποιεί το DEMOS επιλύει προβλήματα που ανήκουν στην Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing - NLP) και συγκεκριμένα στην κατηγορία της ανάλυσης συναισθημάτων (Sentiment Analysis).

ΔΟΜΗ ΕΡΓΑΣΙΑΣ

A ΕΝΟΤΗΤΑ ΘΕΩΡΙΑΣ

Σε αυτή την ενότητα περιγράφονται κάποιες βασικές έννοιες η κατανόηση των οποίων ήταν απαραίτητη για το σχεδιασμό και την υλοποίηση της αποκεντρωμένης εφαρμογής.

- Κεφάλαιο A.1 ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

Σε αυτό το κεφάλαιο αναλύονται οι έννοιες της Συνάρτησης Κατακερματισμού και της Ασύμμετρης Κρυπτογραφίας που είναι βασικές για την εξήγηση της λειτουργίας ενός Blockchain.

- Κεφάλαιο A.2 BLOCKCHAIN

Σε αυτό το κεφάλαιο περιγράφονται τα δύο βασικά δίκτυα Blockchain που υπάρχουν, το Bitcoin και το Ethereum με ιδιαίτερη έμφαση στο δεύτερο που χρησιμοποιήθηκε στην προτεινόμενη λύση. Αναλύονται επίσης κάποιες βασικές αρχές λειτουργίας του Blockchain όπως η δημιουργία ενός block και το πρωτόκολλο συναίνεσης. Ακόμη, δίνεται ο ορισμός της αποκεντρωμένης εφαρμογής (Decentralized Application - dApp) που είναι και το αντικείμενο της προτεινόμενης λύσης.

- Κεφάλαιο A.3 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Σε αυτό το κεφάλαιο δίνεται κάποιο θεωρητικό υπόβαθρο για κάποιες έννοιες Μηχανικής Μάθησης οι οποίες χρησιμοποιήθηκαν στην τελική αποκεντρωμένη εφαρμογή.

- Κεφάλαιο A.4 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ ΚΑΙ BLOCKCHAIN

Σε αυτό το κεφάλαιο γίνεται αναφορά σε προτάσεις που έχουν γίνει στο τομέα που συνδυάζει τις τεχνολογίες της Μηχανικής Μάθησης και του Blockchain. Αυτές οι προτάσεις αποτέλεσαν και πηγή άντλησης ιδεών για το σχεδιασμό της προτεινόμενης λύσης.

B ΕΝΟΤΗΤΑ ΠΡΟΤΕΙΝΟΜΕΝΗΣ ΛΥΣΗΣ

Σε αυτή την ενότητα αναλύεται ο τρόπος με τον οποίο σχεδιάστηκε και υλοποιήθηκε η αποκεντρωμένη εφαρμογή. Περιγράφονται επίσης κάποια αποτελέσματα και τα συμπεράσματα που προέκυψαν από αυτά αλλά και κάποιοι περιορισμοί μαζί με προτάσεις για μελλοντική βελτίωση της εφαρμογής

- Κεφάλαιο B.1 ΣΧΕΔΙΑΣΜΟΣ

Σε αυτό το κεφάλαιο περιγράφεται η γενική ιδέα της εφαρμογής και ο τρόπος λειτουργίας της. Επίσης αναλύεται το μοντέλο Νευρωνικού Δικτύου και τα σύνολα δεδομένων που χρησιμοποιήθηκαν αλλά και η μέθοδος αξιολόγησης της επίδοσής.

- **Κεφάλαιο Β.2 ΥΛΟΠΟΙΗΣΗ**

Σε αυτό το κεφάλαιο αναλύονται τα εργαλεία που χρησιμοποιήθηκαν για τη δημιουργία της αποκεντρωμένης εφαρμογής καθώς επίσης και ο τρόπος με τον οποίο είναι δομημένη. Παράλληλα, υπάρχει ένας συνοπτικό οδηγός εγκατάστασης και χρήσης της εφαρμογής.

- **Κεφάλαιο Β.3 ΑΠΟΤΕΛΕΣΜΑΤΑ & ΣΥΜΠΕΡΑΣΜΑΤΑ**

Σε αυτό το κεφάλαιο παρουσιάζονται κάποια αποτελέσματα που προέκυψαν από δοκιμές που έγιναν στην εφαρμογή. Περιλαμβάνονται επίσης συμπεράσματα που εξάχθηκαν τόσο κατά την υλοποίηση και τη δοκιμή της εφαρμογής αλλά και κατά τη μελέτη του θεωρητικού μέρους της εργασίας

- **Κεφάλαιο Β.4 ΠΕΡΙΟΡΙΣΜΟΙ & ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΒΕΛΤΙΩΣΗ**

Σε αυτό το κεφάλαιο περιγράφονται περιορισμοί της εφαρμογής που εντοπίστηκαν σε διάφορες φάσεις της δημιουργίας της και οδήγησαν σε παρεκκλίσεις από τον αρχικό σχεδιασμό. Τέλος, προτείνονται κάποιες ιδέες που θα μπορούσαν μελλοντικά να βελτιώσουν την εφαρμογή.

A. ΕΝΟΤΗΤΑ ΘΕΩΡΙΑΣ

A.1. ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

A.1.1. Συνάρτηση Κατακερματισμού (Hash function)

Η συνάρτηση κατακερματισμού είναι μία συνάρτηση που δέχεται ως δεδομένο εισόδου ένα σύνολο χαρακτήρων αυθαίρετου μεγέθους M και παράγει ως έξοδο ένα σύνολο χαρακτήρων σταθερού μεγέθους $H(M)$. [3]

Μια έγκυρη συνάρτηση κατακερματισμού οφείλει να πληροί τις εξής ιδιότητες:

- Η είσοδος της συνάρτησης M μπορεί να είναι σύνολο χαρακτήρων οποιουδήποτε μεγέθους
- Η έξοδος της συνάρτησης $H(M)$ οφείλει να είναι σταθερού και προκαθορισμένου μεγέθους
- Ο υπολογισμός της εξόδου $H(M)$ δεδομένης της εισόδου M να είναι υπολογιστικά απλός
- Ο υπολογισμός της εισόδου M δεδομένης της εξόδου $H(M)$ να είναι υπολογιστικά αδύνατος

Επιπλέον ιδιότητες που οφείλει να πληροί μία συνάρτηση κατακερματισμού για να θεωρηθεί κρυπτογραφική (cryptographic hash function)[4]:

- **Ανθεκτικότητα σε προεικόνες** (preimage resistance): Δεδομένης της συνάρτησης $H()$ και για κάθε έξοδο y να είναι υπολογιστικά αδύνατη η εύρεση τιμής x τέτοια ώστε $H(x) = y$
- **Ανθεκτικότητα σε δεύτερες προεικόνες** (second preimage resistance) ή **ασθενής ανθεκτικότητα σε συγκρούσεις** (weak collision resistance): Δεδομένης της συνάρτησης $H()$ και για κάθε είσοδο x να είναι υπολογιστικά αδύνατη η εύρεση τιμής x' τέτοια ώστε $H(x) = H(x')$
- **Ανθεκτικότητα σε συγκρούσεις** (collision resistance) ή **ισχυρή ανθεκτικότητα σε συγκρούσεις** (strong collision resistance): Δεδομένης της συνάρτησης $H()$ να είναι υπολογιστικά αδύνατη η εύρεση ζεύγους x, x' τέτοιο ώστε $H(x) = H(x')$

Οι συναρτήσεις κατακερματισμού έχουν ποικίλες εφαρμογές. Ανάμεσα σε αυτές είναι και οι εξής[3]:

- Ασφαλής αποθήκευση συνθηματικών σε μια βάση δεδομένων με την αποθήκευση της εξόδου συνάρτησης κατακερματισμού $H(x)$ αντί του πραγματικού συνθηματικού x . Με αυτό τον τρόπο εξασφαλίζεται αφενός η επιβεβαίωση της ορθότητας του συνθηματικού που καταχωρεί ο χρήστης, αφετέρου δεν υπάρχει κίνδυνος υποκλοπής του συνθηματικού από κακόβουλες επιθέσεις στη βάση δεδομένων.
- Επιβεβαίωση ακεραιότητας αρχείου διότι η παραμικρή μεταβολή στην είσοδο x της συνάρτησης κατακερματισμού δίνει εντελώς διαφορετική έξοδο $H(x)$. Επομένως, αποθηκεύοντας την έξοδο της συνάρτησης κατακερματισμού για ένα αρχείο μπορεί να εξασφαλιστεί η ακεραιότητά του.

Το Ethereum χρησιμοποιεί την κρυπτογραφική συνάρτηση κατακερματισμού Keccak-256[5]. Αυτή η συνάρτηση παράγει έξοδο μεγέθους 256 Bits. Ο Keccak-256 προτάθηκε αρχικά στον διαγωνισμό SHA-3 Cryptographic Hash Function Competition του 2007 που διοργανώθηκε από το NIST (National Institute of Science and Technology) όπου και κέρδισε.

A.1.2. Ασύμμετρη Κρυπτογραφία (Asymmetric Cryptography)

Εν αντιθέσει με την Συμμετρική Κρυπτογραφία (Symmetric Cryptography), η Ασύμμετρη Κρυπτογραφία (Asymmetric Cryptography) ή Κρυπτογραφία δημοσίου κλειδιού (Public Key Cryptography) χρησιμοποιεί ένα ζεύγος κλειδιών για την κρυπτογράφηση και την αποκρυπτογράφηση. Συγκεκριμένα, το ένα κλειδί χρησιμοποιείται για την κρυπτογράφηση ενός μηνύματος με τον κατάλληλο αλγόριθμο κρυπτογράφησης ενώ το άλλο κλειδί χρησιμοποιείται για την αποκρυπτογράφηση με τον κατάλληλο αλγόριθμο αποκρυπτογράφησης[3].

Παράλληλα, το ένα κλειδί χαρακτηρίζεται ως **ιδιωτικό** και είναι γνωστό αποκλειστικά στον κάτοχο του ενώ το άλλο χαρακτηρίζεται ως **δημόσιο** και δημοσιοποιείται ώστε να χρησιμοποιηθεί από άλλους.

Η κρυπτογράφηση μπορεί να γίνει χρησιμοποιώντας οποιοδήποτε από τα δύο κλειδιά και σε κάθε περίπτωση εξασφαλίζεται διαφορετικό πλεονέκτημα[3]

- Κρυπτογράφηση με ιδιωτικό κλειδί: Ο αποστολέας A κρυπτογραφεί το μήνυμα με το ιδιωτικό του κλειδί PR_A και το αποστέλλει στον παραλήπτη B οποίος το αποκρυπτογραφεί με το δημόσιο κλειδί του αποστολέα PU_A . Με αυτό τον τρόπο εξασφαλίζεται η **πιστοποίηση ταυτότητας** (identity verification) αλλά και η **αδυναμία άρνησης ευθύνης** (non-repudiation) [6] αφού ο καθένας μπορεί να αποκρυπτογραφήσει το μήνυμα αλλά μπορεί να προέρχεται μόνο από τον A. Επίσης εξασφαλίζεται η **ακεραιότητα των δεδομένων** (data integrity) αφού κανένας εκτός του A δεν μπορεί να παραποιήσει το μήνυμα και να το κρυπτογραφήσει με το ιδιωτικό κλειδί PR_A .
- Κρυπτογράφηση με δημόσιο κλειδί: Ο αποστολέας B κρυπτογραφεί το μήνυμα με το δημόσιο κλειδί του παραλήπτη PU_A ο οποίος με τη σειρά του αποκρυπτογραφεί το μήνυμα χρησιμοποιώντας το ιδιωτικό του κλειδί PR_A . Με αυτό τον τρόπο εξασφαλίζεται η **εμπιστευτικότητα** (Confidentiality) αφού ο οποιοσδήποτε μπορεί να κρυπτογραφήσει ένα τέτοιο μήνυμα και να το στείλει στον A αλλά μόνο ο A μπορεί να το διαβάσει.

Συνδυάζοντας τα πιο πάνω ένας αποστολέας A μπορεί να κρυπτογραφήσει ένα μήνυμα με το ιδιωτικό του κλειδί PR_A και στη συνέχεια με το δημόσιο κλειδί PU_B του παραλήπτη B. Ο παραλήπτης B μπορεί να αποκρυπτογραφήσει το μήνυμα αρχικά με το ιδιωτικό του κλειδί PR_B και έπειτα με το δημόσιο κλειδί PU_A του αποστολέα. Έτσι εξασφαλίζεται ότι το μήνυμα έχει προέρθει από τον A, δεν έχει παραποιηθεί και ο μοναδικός που μπορεί να το διαβάσει είναι ο B.

Η Ασύμμετρη Κρυπτογραφία θεωρείται πλέον βασικός πυλώνας της σύγχρονης ασφάλειας δεδομένων (Information Security) και έχει πολυάριθμες εφαρμογές. Στην τεχνολογία του Blockchain χρησιμοποιείται, εκτός άλλων, για τη ψηφιακή υπογραφή των συναλλαγών[5]. Συγκεκριμένα, όταν ένας χρήστης A κρυπτογραφεί τα στοιχεία μιας συναλλαγής με το ιδιωτικό του κλειδί εξασφαλίζει ότι όλοι οι υπόλοιποι κόμβοι του δικτύου μπορούν να επιβεβαιώσουν ότι η συναλλαγή προέρχεται από τον A και ότι τα στοιχεία δεν έχουν παραποιηθεί από κάποιον τρίτο.

Το Ethereum, όπως και το Bitcoin, χρησιμοποιεί **Κρυπτογραφία Ελλειπτικών Καμπυλών** (ECC - Elliptic Curve Cryptography). Η ECC προτιμήθηκε διότι ένα κλειδί 256 bits προσφέρει ισοδύναμη ασφάλεια με ένα κλειδί 3072 bits του επίσης δημοφιλούς αλγορίθμου

ασύμμετρου κλειδιού RSA(Rivest–Shamir–Adleman)[7]. Για να είναι επιτυχής μια επίθεση στην ECC πρέπει να επιλυθεί το πρόβλημα διακριτού λογαρίθμου ελλειπτικής καμπύλης. Από την άλλη για τον RSA απαιτείται παραγοντοποίηση που είναι υπολογιστικά πιο εύκολο πρόβλημα[8].

A.2. BLOCKCHAIN

A.2.1. Bitcoin



Το Bitcoin είναι ένα αποκεντρωμένο άυλο ψηφιακό νόμισμα του οποίου η λειτουργία δεν διέπεται από κανένα σύστημα κεντρικής διαχείρισης. Προτάθηκε το 2008 από τον Satoshi Nakamoto (του οποίου η πραγματική ταυτότητα παραμένει άγνωστη) κατά τη διάρκεια της τότε οικονομικής κρίσης. Η δυσπιστία των ανθρώπων για τις τράπεζες και τα κεντρικά συστήματα διαχείρισης προσέφεραν στο Bitcoin πρόσφορο έδαφος.

Η αρχική ιδέα ήταν η δημιουργία ενός ψηφιακού νομίσματος το οποίο θα προσέφερε στους ιδιοκτήτες του την αμεσότητα που προσφέρουν οι συναλλαγές με μετρητά. Παράλληλα, θα επέτρεπε στους ιδιοκτήτες του να διατηρούν την κυριότητα των νομισμάτων τους χωρίς να χρειάζεται να εμπιστεύονται κάποιο κεντρικό σύστημα αποθήκευσης τους (δηλαδή κάτι αντίστοιχο με το να διατηρούν μετρητά στο σπίτι τους). Η λειτουργία του ψηφιακού συστήματος πληρωμών βασιζόταν σε μεθόδους κρυπτογραφίας και όχι στην εμπιστοσύνη σε ένα κεντρικό σύστημα[9].

Η ιδέα του αποκεντρωμένου άυλου ψηφιακού νομίσματος δεν ήταν καινοτόμα ωστόσο αυτή ήταν η πρώτη ολοκληρωμένη πρόταση που δεν απαιτούσε κανενός είδους κεντρική διαμεσολάβηση. Το βασικό πρόβλημα που παρουσίαζαν οι προηγούμενες προτάσεις ήταν η τρωτότητα σε επιθέσεις όπου ένας χρήστης καταλάμβανε τον έλεγχο του δικτύου δημιουργώντας πολλαπλούς λογαριασμούς ή χρησιμοποιώντας πολλές υπολογιστικές μονάδες (Sybil Attacks)[10].

Για να λειτουργήσει σωστά και με ασφάλεια ένα σύστημα αποκεντρωμένου άυλου ψηφιακού νομίσματος πρέπει να ληφθεί υπόψη το γεγονός ότι οι χρήστες δεν γνωρίζουν και δεν εμπιστεύονται ο ένας τον άλλο. Πρέπει να υπάρχει ένα **Πρωτόκολλο Συναίνεσης** (Consensus Protocol) σύμφωνα με το οποίο η κατάσταση του δικτύου να είναι μοναδική και κοινώς αποδεκτή. Η κατάσταση του δικτύου καθορίζει το ποσό νομισμάτων που έχει κάθε ιδιοκτήτης στην κατοχή του και εξαρτάται αυστηρά από τη χρονολογική σειρά όλων των συναλλαγών[10].

Για την αντιμετώπιση αυτών των προβλημάτων ο Satoshi Nakamoto συνδύασε την αποθήκευση των στοιχείων των συναλλαγών σε μία Blockchain δομή δεδομένων με το **μηχανισμό PoW-Proof of Work**[10]. Αυτή η προσέγγιση μπορεί μεν να μην έδωσε ολοκληρωτική λύση στο πρόβλημα των Sybil Attacks αλλά κατέστησε υπολογιστικά αδύνατη οποιαδήποτε τέτοια επίθεση στο δίκτυο του Bitcoin[11].

A.2.2. Αρχές Blockchain

Το Blockchain είναι ένα αμετάβλητο κατακερματισμένο καθολικό (immutable distributed ledger)[12]. Η έννοια «καθολικό» θα μπορούσε αφαιρετικά να παρομοιαστεί με μία βάση δεδομένων. Τα δεδομένα (κατά βάση στοιχεία συναλλαγών) αποθηκεύονται σε **τμήματα** (blocks) όπου το καθένα ενώνεται με το προηγούμενο δημιουργώντας μια αλυσίδα, το λεγόμενο **Blockchain**. Το πρώτο block ονομάζεται **block γένεσης** (genesis block).

Το δίκτυο του Blockchain αποτελείται από υπολογιστικές μονάδες που αποκαλούνται **κόμβοι** (nodes). Οι κόμβοι μπορούν να εκτελέσουν, να καταγράψουν και να επιβεβαιώσουν συναλλαγές στο Blockchain. Για να υπάρχει γενική συναίνεση για την κατάσταση του Blockchain όλοι οι κόμβοι διατηρούν ένα πλήρες αντίγραφο του με όλες τις συναλλαγές που έχουν γίνει ποτέ στο δίκτυο. Το αντίγραφο ανανεώνεται τακτικά με τις νέες συναλλαγές που καταγράφονται.

Κάθε block περνά από συνάρτηση κατακερματισμού η έξοδος της οποίας αποθηκεύεται στο επόμενο block. Η διαδικασία επαναλαμβάνεται για το επόμενο block και το αποτέλεσμα αποθηκεύεται στο μεθεπόμενο κ.ο.κ. Όπως προαναφέρθηκε στην παράγραφο A.1.1 Συνάρτηση Κατακερματισμού (Hash function) σελ.13, οποιαδήποτε μεταβολή στην είσοδο της συνάρτησης κατακερματισμού αλλάζει εντελώς την έξοδο. Εξ αιτίας αυτού εξασφαλίζεται ότι τα δεδομένα που καταχωρούνται στο Blockchain δεν μπορούν ούτε να διαγραφούν αλλά ούτε και να παραποιηθούν.

Αν κάποιος κακόβουλος χρήστης προσπαθήσει να αλλοιώσει το δικό του αντίγραφο του Blockchain, η διαφορά των εξόδων κατακερματισμού σε σχέση με όλα τα υπόλοιπα αντίγραφα που διατηρούν οι υπόλοιποι κόμβοι είναι εμφανής. Ο μόνος τρόπος να υπερισχύσει μια παραποιημένη αλυσίδα Blockchain αντί της πραγματικής, είναι ο κακόβουλος χρήστης να αποκτήσει πρόσβαση σε ποσοστό τουλάχιστον 51% της συνολικής υπολογιστικής ισχύος του δικτύου, κάτι που είναι πρακτικά αδύνατο.

A.2.2.I. Συναλλαγή

Μια συναλλαγή στο Blockchain έχει ως είσοδο και έξοδο ένα ή περισσότερα UTXOs (Unspent Transaction Outputs). Κάθε UTXO έχει ένα ιδιοκτήτη και ένα ποσό και μπορεί να παρομοιαστεί με μία επιταγή που έχει ποσό και παραλήπτη και δεν έχει εξαργυρωθεί ακόμα. Η κατάσταση του δικτύου είναι στην ουσία η συλλογή όλων των UTXOs και βάσει αυτών καθορίζεται και το υπόλοιπο του κάθε λογαριασμού σε κρυπτονομίσματα[10].

Έστω ότι η Alice επιθυμεί να στείλει στον Bob 12.5 BTC (Bitcoins). Αρχικά πρέπει να έχει στην κατοχή της ένα ή περισσότερα UTXOs το άθροισμα των οποίων να είναι ίσο ή μεγαλύτερο από 12.5 BTC. Ας υποθέσουμε ότι έχει τρία UTXOs προς το λογαριασμό της με ποσά 6, 5 και 3 BTC, άρα συνολικό άθροισμα 14 BTC. Η είσοδος της συναλλαγής της είναι λοιπόν αυτά τα τρία UTXOs. Η έξοδος της συναλλαγής είναι ένα UTXO με ποσό 12.5 BTC και παραλήπτη τον Bob και ένα UTXO με ποσό 1.5 BTC με παραλήπτη την ίδια την Alice (κάτι δηλαδή σαν ρέστα από τη συναλλαγή). Η συναλλαγή αυτή καταγράφεται στο Blockchain και επιβεβαιώνεται από τους υπόλοιπους κόμβους με αποτέλεσμα το υπόλοιπο

του λογαριασμού της Alice να μειώνεται κατά 12.5 BTC και το υπόλοιπο του λογαριασμού του Bob να αυξάνεται κατά 12.5 BTC[10].

Για να θεωρηθεί έγκυρη μία συναλλαγή πρέπει:

- Ο αποστολέας να έχει στην κατοχή του έγκυρα UTXOs το άθροισμα των οποίων να είναι ίσο ή μεγαλύτερο από το ποσό της συναλλαγής (Από τη στιγμή που όλοι οι κόμβοι έχουν αντίγραφο για όλες τις συναλλαγές που έγιναν ποτέ στο δίκτυο, μπορούν εύκολα να το επιβεβαιώσουν ή να το διαψεύσουν)
- Η ψηφιακή υπογραφή του αποστολέα της συναλλαγής (όπως αυτή περιγράφεται στην παράγραφο A.1.2 Ασύμμετρη Κρυπτογραφία (Asymmetric Cryptography), σελ.14) να ταυτίζεται με αυτή των UTXOs εισόδου
- Το άθροισμα των ποσών των UTXOs εισόδου να μην είναι μικρότερο από το άθροισμα των ποσών των UTXOs εξόδου

Πρέπει επίσης να σημειωθεί ότι το τελικό ποσό της συναλλαγής περιλαμβάνει και ένα μικρό **τέλος συναλλαγής** (transaction fee) που προσφέρεται ως ανταμοιβή στον κόμβο που θα καταγράψει τη συναλλαγή στο Blockchain.

Εφόσον η συναλλαγή θεωρηθεί έγκυρη, η νέα κατάσταση του δικτύου ενημερώνεται και έτσι περιλαμβάνει τα UTXOs εξόδου και δεν περιλαμβάνει πια τα UTXOs εισόδου[10].

A.2.2.II. Δημιουργία Block

Οι συναλλαγές που δεν έχουν καταχωρηθεί ακόμα στο Blockchain θεωρούνται **αιτήματα συναλλαγών** η ύπαρξη των οποίων αναμεταδίδεται σε όλο το δίκτυο. Ειδικό κόμβοι του δικτύου που ονομάζονται **Μεταλλωρύχοι** (Miners) συλλέγουν αιτήματα συναλλαγών και δημιουργούν πιθανά blocks αφού πρώτα επιβεβαιώσουν την εγκυρότητα αυτών των συναλλαγών (με τον τρόπο που αναφέρθηκε στην παράγραφο A.2.2.I Συναλλαγή σελ.17). Η σειρά με την οποία ο Μεταλλωρύχος τοποθετεί τις συναλλαγές στο πιθανό block είναι σημαντική αφού τα UTXOs εξόδου μίας συναλλαγής μπορεί να χρησιμοποιούνται ως UTXOs εισόδου για μια άλλη συναλλαγή εντός του block. Επίσης, είναι σημαντικό να υπάρχει αναφορά στο αμέσως προηγούμενο έγκυρο block ώστε να δημιουργείται η αλυσίδα του Blockchain αλλά και η **χρονοσφραγίδα** (timestamp) του πιθανού block[13].

Αφότου γίνει αυτό, ο μεταλλωρύχος προσπαθεί να λύσει το πρόβλημα Proof of Work. Εφόσον τα καταφέρει, δημοσιοποιεί το πιθανό block σε ολόκληρο το Blockchain ώστε να υποβληθεί σε μια διαδικασία επικύρωσης από τους υπόλοιπους κόμβους του δικτύου[13].

Η επικύρωση ενός block γίνεται με την εξής διαδικασία[10]:

- Επαληθεύεται ότι το block το οποίο αναφέρεται ως αμέσως προηγούμενο υπάρχει και είναι έγκυρο
- Επαληθεύεται ότι η χρονοσφραγίδα του block έπεται χρονικά της χρονοσφραγίδας του προηγούμενου block αλλά όχι περισσότερο από 2 ώρες για το Bitcoin και 15 λεπτά για το Ethereum
- Επαληθεύεται ότι το PoW του block είναι έγκυρο
- Επαληθεύεται ότι οι συναλλαγές είναι έγκυρες και συμβατές με την κατάσταση που δημιουργήθηκε μετά την καταχώρηση του προηγούμενου block

Το PoW όπως έχει δημιουργηθεί ώστε να εξασφαλίζει ότι δημιουργείται ένα νέο block κάθε περίπου 10 λεπτά για το Bitcoin και κάθε περίπου 14 δευτερόλεπτα για το Ethereum. Είναι πιθανό να δημιουργηθούν 2 ή περισσότερα έγκυρα block σχεδόν την ίδια χρονική στιγμή με αποτέλεσμα να δημιουργηθεί μια **διακλάδωση** (fork) όπου κάποιοι κόμβοι να ακολουθούν μία διακλάδωση και κάποιοι κόμβοι άλλη διακλάδωση. Η υλοποίηση του Blockchain απαγορεύει ρητά την ύπαρξη τέτοιων διακλαδώσεων επομένως στο τέλος υπερισχύει μόνο μία διακλάδωση. Γενικά ισχύει ο κανόνας ότι η πιο μεγάλη αλυσίδα είναι και η πιο έγκυρη αλυσίδα του Blockchain και τελικά όλοι οι κόμβοι υιοθετούν μία και μοναδική αλυσίδα[9].

Η ιδέα της δημιουργίας των Blocks εξασφαλίζει το απαιτούμενο **πρωτόκολλο συναίνεσης** (consensus protocol) αφού η επίσημη σειρά των συναλλαγών είναι και η σειρά με την οποία τοποθετούνται στα blocks και είναι επαληθευμένη από όλους τους κόμβους του δικτύου. Παράλληλα δεν μπορεί να μεταβληθεί αφού άπαξ και ένα block προστεθεί στην τελική αλυσίδα τότε δεν μπορεί να διαγραφεί ή να παραποιηθεί με κανένα τρόπο.

Η δημιουργία των blocks με την επίλυση του PoW είναι μια ενεργειακά και υπολογιστικά κοστοβόρα διαδικασία αλλά είναι και ζωτικής σημασίας για την ύπαρξη του Blockchain. Για αυτό το λόγο, ο μεταλλωρύχος έχει το δικαίωμα να τοποθετεί ως πρώτη συναλλαγή μια ειδική συναλλαγή για την οποία δεν υπάρχουν UTXOs εισόδου αλλά υπάρχουν UTXOs εισόδου προς τον εαυτό του. Συνεπώς αυξάνεται το υπόλοιπο του λογαριασμού του μεταλλωρύχου και τίθενται σε κυκλοφορία νέα κρυπτονομίσματα (αφού δεν υπάρχει κάποια κεντρική αρχή για να τα διανείμει). Ωστόσο, το κέρδος του μεταλλωρύχου προέρχεται και από τα επί μέρους **τέλη συναλλαγών** (transaction fees) που προκύπτουν από κάθε συναλλαγή που προσθέτει στο block που δημιουργεί[9].

A.2.2.III. Πρωτόκολλο Συναίνεσης (Consensus Protocol)

Όπως προαναφέρθηκε, για τη σωστή λειτουργία του Blockchain είναι απαραίτητο ένα **Πρωτόκολλο Συναίνεσης** (Consensus Protocol) το οποίο εξασφαλίζει ότι η κατάσταση του δικτύου είναι μοναδική και κοινώς αποδεκτή.

Ένα βασικό πρόβλημα το οποίο μπορεί να κληθεί να αντιμετωπίσει το δίκτυο του Blockchain είναι το πρόβλημα της **διπλής εξόφλησης** (double spending). Αυτό προκύπτει διότι ένας αποδέκτης μίας συναλλαγής δεν μπορεί να είναι βέβαιος ότι ο αποστολέας δεν έχει χρησιμοποιήσει τα ίδια κρυπτονομίσματα (ουσιαστικά τα ίδια UTXOs) σε κάποια άλλη συναλλαγή[9].

Έστω δηλαδή ότι ο Bob δημιουργεί ένα αίτημα συναλλαγής με το οποίο στέλνει στην Alice 10 BTC. Αμέσως μετά δημιουργεί ένα νέο αίτημα συναλλαγής με τα ίδια 10 BTC και παραλήπτη τον εαυτό του. Αν καταφέρει να πείσει το δίκτυο πως η συναλλαγή προς τον εαυτό του προηγείται τότε η συναλλαγή προς την Alice θεωρείται άκυρη (μπορεί να παρομοιαστεί με ακάλυπτη επιταγή)[9].

Λόγω της φύσης του Blockchain, η συναλλαγή προς την Alice θα αναμεταδοθεί πρώτη και θα επιβεβαιωθεί η εγκυρότητά της πριν να επιβεβαιωθεί η εγκυρότητα της κακόβουλης συναλλαγής του Bob. Ο μόνος τρόπος να ξεγελάσει το δίκτυο ο Bob είναι δημιουργώντας ένα δικό του έγκυρο Block το οποίο να επικυρώνει την κακόβουλη συναλλαγή αντί της συναλλαγής προς την Alice δημιουργώντας έτσι μία διακλάδωση (fork)[10].

Από εκείνη τη στιγμή ξεκινάει ένας «αγώνας δρόμου». Ο Bob προσπαθεί να δημιουργήσει blocks πιο γρήγορα από όλο το υπόλοιπο δίκτυο ώστε η δική του παραποιημένη αλυσίδα να γίνει πιο μεγάλη και να υπερισχύσει της πραγματικής την οποία δημιουργούν όλοι οι υπόλοιποι κόμβοι. Όμως η δημιουργία ενός έγκυρου block απαιτεί την επίλυση του PoW (στις περιπτώσεις που το πρωτόκολλο που χρησιμοποιεί PoW) , κάτι που είναι υπολογιστικά δύσκολο. Επομένως, ο μόνος τρόπος να επιτύχει ο Bob είναι να έχει στη διάθεση του περισσότερη υπολογιστική ισχύ από ότι ολόκληρο το υπόλοιπο δίκτυο[10].

Μερικά από τα πρωτόκολλα που χρησιμοποιούνται ή έχουν προταθεί είναι τα εξής:

A.2.2.III.a. Proof of Work - PoW

Το PoW είναι το πρωτόκολλο που προτάθηκε αρχικά από τον Satoshi Nakamoto και χρησιμοποιείται, προς το παρόν, για το Bitcoin, το Ethereum αλλά και πολλά άλλα κρυπτονομίσματα. Είναι στην ουσία ένα περίπλοκο υπολογιστικό πρόβλημα η λύση του οποίου δεν μπορεί να βρεθεί με συστηματικό τρόπο αλλά μόνο με τυχαίες δοκιμές[14].

Η επικεφαλίδα ενός block διαθέτει διάφορα πεδία όπως η χρονοσφραγίδα του, το hash (έξοδος συνάρτησης κατακερματισμού) του αμέσως προηγούμενου block, ο αριθμός των συναλλαγών που περιέχει κ.α.. Στην επικεφαλίδα κάθε block υπάρχει και ένα πεδίο **nonce** το οποίο παίρνει μια αριθμητική τιμή. Ο στόχος των μεταλλωρύχων είναι να βρουν μια τιμή για το nonce τέτοια ώστε η έξοδος της συνάρτησης κατακερματισμού, όταν δοθεί ως είσοδος ολόκληρο το block, να ξεκινάει με κάποιο αριθμό μηδενικών (δηλαδή να είναι της μορφής 0000000000eebcb6d...). Όπως προειπώθηκε στην παράγραφο A.1.1 Συνάρτηση Κατακερματισμού (Hash function), σελ.13, οποιαδήποτε μικρή μεταβολή στην είσοδο της συνάρτησης αλλάζει πλήρως την έξοδο με ακανόνιστο τρόπο. Συνεπώς, ο τρόπος επίλυσης του PoW είναι δοκιμάζοντας τυχαίες τιμές για το nonce και η δυσκολία επίλυσής του είναι εκθετική ως προς τον αριθμό των μηδενικών που απαιτούνται.

Εφόσον ένας μεταλλωρύχος δημιουργήσει ένα έγκυρο block όπως αυτό περιγράφηκε στην παράγραφο A.2.2.II Δημιουργία Block, σελ.18 και βρει την κατάλληλη τιμή για το πεδίο nonce τότε το δημοσιοποιεί στο δίκτυο. Οι υπόλοιποι κόμβοι επιβεβαιώνουν την εγκυρότητα των δεδομένων του block αλλά και την ορθότητα της λύσης του PoW και το προσθέτουν στην αλυσίδα. Τέλος, ο μεταλλωρύχος λαμβάνει ως ανταμοιβή τα νέα κρυπτονομίσματα που δημιουργούνται καθώς και τα τέλη συναλλαγών (transaction fees) από τις συναλλαγές που πρόσθεσε στο block

Το PoW θα μπορούσε να παρομοιαστεί με ένα θησαυροφυλάκιο με μία κλειδαριά με κωδικό. Οι μεταλλωρύχοι δοκιμάζουν τυχαίες τιμές στη κλειδαριά μέχρι κάποιος να καταφέρει να την ανοίξει. Αυτός που θα την ανοίξει θα πάρει και το θησαυρό.

Ασφάλεια:

Το δίκτυο παραμένει ασφαλές εφόσον τουλάχιστον το 51% της συνολικής υπολογιστικής ισχύος διαχειρίζεται από καλόβουλους χρήστες [14]

Πλεονεκτήματα[9], [15]

- Είναι απλό στη λογική αλλά και στην υλοποίηση
- Είναι υπολογιστικά εύκολο να επιβεβαιωθεί η ορθότητά μιας πιθανής λύσης

- Είναι διαδεδομένο και έχει επαληθευτεί εδώ και χρόνια ότι αποδίδει καλά
- Είναι ένα σύστημα πλειοψηφίας στο οποίο ισχύει το one-CPU-one-vote άρα δεν έχει τα ελαττώματα των συστημάτων one-IP-address-one-vote όπου μία υπολογιστική μονάδα θα μπορούσε να έχει περισσότερες από μία ψήφους
- Έχει μεταβλητό επίπεδο δυσκολίας ώστε να είναι ελεγχόμενα τα χρονικά διαστήματα μεταξύ της δημιουργίας των blocks ανεξάρτητα με την αύξηση της υπολογιστικής ισχύος των μεταλλωρύχων
- Μπορεί, θεωρητικά, ο οποιοσδήποτε να γίνει μεταλλωρύχος και να προσπαθήσει να δημιουργήσει blocks χωρίς να έχει στην κατοχή του κρυπτονομίσματα

Μειονεκτήματα[15]

- Χρησιμοποιεί τεράστιες ποσότητες ηλεκτρικής ενέργειας με τραγικές συνέπειες για το περιβάλλον
- Στην πράξη, απαιτείται εξειδικευμένος υπολογιστικός εξοπλισμός με πολύ μεγάλη υπολογιστική ισχύ αφού υπάρχει μεγάλος ανταγωνισμός μεταξύ των μεταλλωρύχων
- Υπάρχει ανησυχία ότι οι δεξαμενές συλλογής αιτημάτων συναλλαγών από τις οποίες αντλούν αιτήματα συναλλαγών οι μεταλλωρύχοι για τη δημιουργία των blocks, μπορούν να οδηγήσουν σε κεντροποίηση του Blockchain
- Η υπολογιστική ισχύς που χρησιμοποιείται για το PoW αναλώνεται σε ανούσιες πράξεις που δεν επιτελούν κάποιο πρακτικό σκοπό

A.2.2.III.β. Proof of Stake - PoS

Το PoS είναι ένα ανερχόμενο πρωτόκολλο το οποίο αναμένεται να αντικαταστήσει το PoW στην περίπτωση του Ethereum και χρησιμοποιείται ήδη με επιτυχία σε άλλα κρυπτονομίσματα. Η βασική ιδέα του PoS είναι ότι δεν υπάρχουν πια μεταλλωρύχοι (miners) αλλά τη θέση τους παίρνουν **επικυρωτές** (validators)[16].

Στην περίπτωση του PoS οι κόμβοι δεν ανταγωνίζονται μεταξύ τους. Η επιλογή του κόμβου που θα δημιουργήσει το επόμενο block γίνεται τυχαία ανάμεσα σε όλους τους επικυρωτές. Για να έχουν το δικαίωμα να επιλεγούν, οι επικυρωτές θέτουν ως εγγύηση ένα μέρος του κεφαλαίου τους (των κρυπτονομισμάτων τους). Όσο μεγαλύτερη εγγύηση έχει θέσει ένας επικυρωτής, τόσο μεγαλύτερη η πιθανότητα να επιλεγεί για τη δημιουργία του επόμενου block. Παρόλα αυτά, η αμοιβή του επικυρωτή προέρχεται σε αυτή την περίπτωση αποκλειστικά από τα τέλη συναλλαγών που περιλαμβάνονται στο block (transaction fees) και δεν δημιουργούνται νέα νομίσματα[17], [18].

Εξάλλου, ένας επικυρωτής είναι επιφορτισμένος με την ευθύνη να επικυρώνει τα blocks που έχουν προταθεί από άλλους επικυρωτές. Αν δεν είναι συνεπής σε αυτή την υποχρέωση, μπορεί να του αφαιρεθεί μέρος της εγγύησης που έχει καταβάλει. Για να θεωρηθεί μια κατάσταση του δικτύου ως έγκυρη πρέπει να συμφωνήσουν τα 2/3 των επικυρωτών[16].

Το PoS θα μπορούσε να παρομοιαστεί με μια κλήρωση με λαχνούς. Ο νικητής επιλέγεται μεν τυχαία αλλά όσους περισσότερους λαχνούς έχει αγοράσει κάποιος επικυρωτής, τόσο πιθανότερο είναι να κερδίσει την κλήρωση.

Ασφάλεια:

Το δίκτυο παραμένει ασφαλές εφόσον δεν υπάρχει κακόβουλος χρήστης που να κατέχει το 51% των κρυπτονομισμάτων που έχουν τεθεί συνολικά ως εγγύηση (στην περίπτωση του Ethereum αυτό είναι πρακτικά αδύνατο λόγω του μεγάλου αριθμού των νομισμάτων αλλά και της υψηλής τιμής τους). Παράλληλα, ένας επικυρωτής μπορεί να τιμωρηθεί με κυρώσεις επί της εγγύησης που έχει θέσει, αν διαπιστωθεί κακόβουλη συμπεριφορά ή γενικά δεν συμμορφώνεται με τους κανόνες του δικτύου[14].

Πλεονεκτήματα[14], [16]–[18]

- Είναι πολύ πιο φιλικό προς το περιβάλλον αφού δεν απαιτεί τεράστια ποσά ηλεκτρικής ενέργειας για τους υπολογισμούς
- Δεν απαιτεί εξειδικευμένο εξοπλισμό εξόρυξης όπως το PoW επομένως μπορεί να συμμετέχει οποιοσδήποτε διαθέτει αρκετό κεφάλαιο για να καλύψει την ελάχιστη εγγύηση
- Τα τέλη συναλλαγών για την ανταμοιβή των επικυρωτών μπορούν να είναι πιο χαμηλά αφού δεν έχει σημαντικό κόστος η συμμετοχή στην κλήρωση (σε αντίθεση με την εξόρυξη)
- Μειώνεται η απειλή της κεντριοποίησης αφού υπάρχει κίνητρο σε περισσότερους κόμβους να συμμετέχουν ως επικυρωτές (από ότι σαν μεταλλωρύχοι)
- Οι κυρώσεις που επιβάλλονται επί των εγγυήσεων για κακοήγη συμπεριφορά, μειώνουν το κίνητρο για κακόβουλες επιθέσεις
- Αναμένεται ότι προσφέρει καλύτερη κλιμακωσιμότητα

Μειονεκτήματα[16]

- Είναι πιο σύνθετο στη λογική αλλά και στην υλοποίηση
- Δεν έχει χρησιμοποιηθεί όσο το PoW στην πράξη οπότε δεν έχει αποδειχθεί ότι αποδίδει εξίσου καλά
- Απαιτεί ένα σημαντικό αρχικό κεφάλαιο ως ελάχιστη εγγύηση, στην περίπτωση του Ethereum 32ETH (ποσό πέραν των €60000 στις 23/05/2022)
- Ευνοούνται οι επικυρωτές που έχουν μεγαλύτερα κεφάλαια επομένως υπάρχει συσσώρευση του πλούτου

A.2.2.III.γ. Proof of Learning – PoLe

Το συγκεκριμένο PoLe πρωτόκολλο δεν έχει υλοποιηθεί και δεν χρησιμοποιείται σε κάποιο Blockchain αλλά, προς το παρόν, αποτελεί θεωρητική πρόταση[1]. Η ιδέα για το PoLe προέκυψε με γνώμονα δύο βασικά προβλήματα. Το πρώτο ήταν ότι το PoW καταναλώνει τεράστια ποσά ηλεκτρικής ενέργειας και χρησιμοποιεί μεγάλη υπολογιστική ισχύ που αναλώνεται σε ανούσιες πράξεις που δεν επιτελούν κάποιο πρακτικό σκοπό. Το δεύτερο ήταν ότι οι υπολογισμοί για τη Μηχανική Μάθηση και συγκεκριμένα την εκπαίδευση των Νευρωνικών Δικτύων-ΝΔ απαιτούν μεγάλη υπολογιστική ισχύ. Ο στόχος λοιπόν του PoLe είναι να αξιοποιήσει την υπολογιστική ισχύ του δικτύου Blockchain για το πρωτόκολλο συναίνεσης για τη βελτιστοποίηση Νευρωνικών Δικτύων.

Σύμφωνα με το PoLe, το train dataset δημοσιοποιείται στο δίκτυο του Blockchain και οι κόμβοι (που λειτουργούν ως μεταλλωρύχοι στο PoW) θα αξιοποιούν την υπολογιστική τους ισχύ προσπαθώντας να εκπαιδεύσουν όσο γίνεται καλύτερα ένα Νευρωνικό Δίκτυο.

Για το PoLe οι κόμβοι του δικτύου χωρίζονται σε δύο τύπους, τους **κόμβους δεδομένων** (data nodes) και τους **κόμβους συναίνεσης** (consensus nodes) ή **μεταλλωρύχους** (miners).

Οι πρώτοι δημοσιοποιούν στο δίκτυο τα χαρακτηριστικά (τη δομή) ενός επιθυμητού ΝΔ, ένα train dataset, μία ελάχιστη ανεκτή τιμή ακριβείας (accuracy) που πρέπει να επιτευχθεί για να θεωρηθεί επιτυχής η εκπαίδευση και ένα ποσό ανταμοιβής. Είναι επίσης υπεύθυνοι να αποδώσουν το ποσό ανταμοιβής στο νικητή κόμβο συναίνεσης.

Οι κόμβοι συναίνεσης ή μεταλλωρύχοι λαμβάνουν τα δεδομένα από τους κόμβους δεδομένων και προσπαθούν μέσα σε εύλογο χρονικό διάστημα να εκπαιδεύσουν το ζητούμενο ΝΔ με ακρίβεια μεγαλύτερη από την ελάχιστη ανεκτή και να δημοσιοποιήσουν τη λύση τους. Εφόσον ο κόμβος καταφέρει να εκπαιδεύσει κατάλληλα ένα ΝΔ πριν την εκπνοή του διαθέσιμου χρόνου, δημιουργεί ένα πιθανό block στο οποίο περιλαμβάνονται και τα στοιχεία για το ΝΔ που εκπαιδεύσε.

Μόλις η διορία για την εκπαίδευση εκπνεύσει, το πραγματικό test dataset, βάσει του οποίου θα κριθούν τα υποβληθέντα ΝΔ, δημοσιοποιείται. Νικητής είναι ο κόμβος συναίνεσης που θα επιτύχει καλύτερη τιμή ακριβείας για το πραγματικό test dataset και το block του προστίθεται στο Blockchain, δεδομένου ότι το block που κατέθεσε είναι έγκυρο. (βλ. παράγραφο Α.2.2.Π Δημιουργία Block, σελ. 18)

Για την αποφυγή επαναχρησιμοποίησης εκπαιδευμένων μοντέλων προτάθηκε από το [1] η προσθήκη ενός στρώματος (layer) στο ΝΔ το οποίο ονομάζεται Secure Mapping Layer. Αυτό κωδικοποιεί τα διανύσματα χαρακτηριστικών εισόδου του ΝΔ με τρόπο που εξαρτάται από την έξοδο συνάρτησης κατακερματισμού (hash) του προηγούμενου block. Έτσι δεν μπορεί κάποιος κόμβος να επαναχρησιμοποιήσει κάποια λύση για προηγούμενο ΝΔ.

Σημείωση: Εκτός από τα πιο πάνω πρωτόκολλα έχουν προταθεί και χρησιμοποιούνται πολλά άλλα με τα δικά τους πλεονεκτήματα και μειονεκτήματα. Ταυτόχρονα, υπάρχουν Blockchains που χρησιμοποιούν συνδυασμούς από τέτοια πρωτόκολλα.



Το Ethereum ξεκίνησε ως ιδέα το 2013 από τον Vitalik Buterin ο οποίος σε συνεργασία με τους Gavin Wood, Charles Hoskinson, Anthony Di Iorio και Joseph Lubin το έθεσε σε κυκλοφορία το 2015[19]. Πρωταρχικός στόχος του Ethereum ήταν να αποτελέσει μια αναβαθμισμένη έκδοση Blockchain που θα προσέφερε προηγμένα χαρακτηριστικά σε σχέση με το Bitcoin. Η καινοτομία του Ethereum ήταν ότι προσέφερε υποστήριξη σε μια Turing Complete γλώσσα προγραμματισμού. Με αυτό τον τρόπο οι ιδρυτές του Ethereum οραματίστηκαν ένα Blockchain προγραμματίσιμο και προσαρμόσιμο που θα ξέφευγε από τα στενά όρια του να αποτελεί αποκλειστικά ένα ψηφιακό νόμισμα. Έτσι, το Ethereum έμελλε να αποτελέσει την απαρχή μιας νέας εποχής στο διαδίκτυο, βασισμένης σε αποκεντρωμένες εφαρμογές, γνωστής ως Web 3.

Η Web 1 εποχή ήταν και η πρώτη εποχή του διαδικτύου και περιελάμβανε στατικές ιστοσελίδες οι οποίες παρέχονταν στους χρήστες από κάποιο διακομιστή (server). Γενικά η αλληλεπίδραση του χρήστη με το διαδίκτυο ήταν μονόδρομη[20]. Η μετάβαση στην εποχή του Web 2 έγινε το 2004. Οι ιστοσελίδες άρχισαν να γίνονται δυναμικές και διαδραστικές και η αλληλεπίδραση με τον χρήστη έγινε αμφίδρομη[21]. Κλασσικό παράδειγμα εφαρμογών της Web 2 εποχής είναι τα μέσα κοινωνικής δικτύωσης.

Το κοινό χαρακτηριστικό των πρώτων δύο εποχών ήταν η εξάρτηση από κάποιο κεντρικό σύστημα. Το Web 3 βασίζεται σε εφαρμογές που είναι αποκεντρωμένες. Στο Web 3 δεν υπάρχει η έννοια της εξουσιοδότησης (permissionless) και οι χρήστες είναι ελεύθεροι να συμμετέχουν σε αυτό κατά βούληση. Επίσης, δεν υπάρχει η ανάγκη εμπιστοσύνης σε κάποιο κεντρικό σύστημα (trustless) και οι συναλλαγές γίνονται απευθείας μεταξύ των χρηστών. Παράλληλα, οι εφαρμογές δεν έχουν ένα μοναδικό σημείο αποτυχίας στην υποδομή τους (single point of failure) όπως ένας κεντρικός διακομιστής (server). Τέλος, το Web 3 προσφέρει στους χρήστες την κυριότητα των ψηφιακών τους περιουσιακών στοιχείων[22], [23].

Για παράδειγμα, στο Web 2 όταν ένας χρήστης αγοράζει ένα λογισμικό, η εταιρία του λογισμικού μπορεί να διαγράψει το λογαριασμό του χρήστη και να του απαγορεύσει την πρόσβαση. Κάτι τέτοιο δεν θα είναι εφικτό στο Web 3[22]. Ακόμη, στο Web 3 είναι αδύνατη η λογοκρισία αφού τα δεδομένα των χρηστών δεν θα περνούν από κάποιο σύστημα κεντρικής αξιολόγησης πριν δημοσιευτούν[23].

A.2.3.I. Ψηφιακή Μηχανή Ethereum (Ethereum Virtual Machine – EVM)

Το EVM είναι μία και μοναδική μηχανή καταστάσεων και ορίζει την παρούσα κατάσταση του δικτύου. Όλοι οι κόμβοι του δικτύου του Ethereum συμφωνούν με αυτή την κατάσταση και διατηρούν ένα αντίγραφο του EVM[24]. Για κάθε block του Ethereum Blockchain υπάρχει μία μόνο κατάσταση και το EVM καθορίζει επίσης τους κανόνες της μετάβασης από μία έγκυρη κατάσταση σε μία άλλη[25].

Η ανάγκη ύπαρξης του EVM πηγάζει από το γεγονός ότι το Ethereum διαφοροποιείται από τα υπόλοιπα ψηφιακά κρυπτονομίσματα. Παρόλο που έχει και αυτό το δικό του νόμισμα, το

Ether - ETH, οι λειτουργίες που επιτρέπει να γίνουν στο Blockchain είναι πολύ πιο σύνθετες. Στο Bitcoin και σε αντίστοιχα Blockchain τα δεδομένα που πρέπει να αποθηκευτούν είναι μόνο οι λογαριασμοί και τα υπόλοιπά τους. Στην περίπτωση του Ethereum όμως υπάρχουν περισσότερα και πιο σύνθετα δεδομένα τα οποία αποθηκεύονται σε καταστάσεις και αυτό οφείλεται στα smart contracts[25].

A.2.3.II. Έξυπνα συμβόλαια (Smart Contracts)

Τα Smart contracts είναι ουσιαστικά προγράμματα τα οποία ανεβαίνουν στο δίκτυο του Ethereum και αποθηκεύονται στο EVM ώστε μπορούν να εκτελεστούν από τους κόμβους του δικτύου[26]. Ο καθένας μπορεί να δημιουργήσει ένα τέτοιο συμβόλαιο και να το ανεβάσει στο δίκτυο πληρώνοντας ένα αντίτιμο σε ETH[27].

Τα smart contracts γράφονται σε κάποια συμβατή γλώσσα προγραμματισμού και η πιο διαδεδομένη είναι η Solidity. Αφότου ετοιμαστούν, τα smart contracts μεταγλωττίζονται σε bytecode κατανοητό από το EVM ενώ ταυτόχρονα δημιουργείται ένα JSON αρχείο το οποίο ονομάζεται Application Binary Interface – ABI. Το ABI περιέχει πληροφορίες για τη δομή του συμβολαίου και τις συναρτήσεις του[28].

Η δομή των contracts μοιάζει με κλάσεις (classes) σε αντικειμενοστραφείς γλώσσες προγραμματισμού (object-oriented languages). Μπορούν να περιέχουν σταθερές, μεταβλητές, συναρτήσεις, σύνθετες δομές δεδομένων όπως structs κ.α.. Μπορούν επίσης να κληρώνονται άλλα contracts ή να χρησιμοποιούνται ως βιβλιοθήκες[29].

Παρά τη μεγάλη ευελιξία τους, τα smart contracts υπόκεινται σε κάποιους περιορισμούς. Ένας βασικός περιορισμός είναι ότι δεν μπορούν να αντλήσουν πληροφορίες για τον πραγματικό κόσμο (πχ την πρόβλεψη του καιρού). Αυτό ισχύει διότι δεν μπορούν να στείλουν αιτήματα HTTP (HTTP requests) αν και υπάρχουν τρόποι να παρακαμφθεί αυτός ο περιορισμός. Ακόμη, υπάρχει περιορισμός για το μέγεθος που μπορούν τα contracts να έχουν[27].

Μια άλλη οπτική για τα smart contracts είναι ότι αποτελούν μια μέθοδο αναγωγής μιας συμφωνίας σε προγραμματιστικό κώδικα. Με άλλα λόγια αποτελούν ένα ψηφιακό συμβόλαιο το οποίο εκτελείται αυτόματα όταν οι όροι και οι προϋποθέσεις του έχουν εκπληρωθεί. Έτσι τα smart contracts αφαιρούν την ανάγκη εμπιστοσύνης μεταξύ δύο κόμβων στο Blockchain[30]. Για παράδειγμα, αν δύο άτομα βάλουν στοίχημα για την έκβαση ενός ποδοσφαιρικού αγώνα, ο ηττημένος μπορεί τελικά αρνηθεί να πληρώσει το ποσό που υποσχέθηκε. Αν όμως αυτή η συμφωνία υλοποιηθεί με ένα smart contract εξασφαλίζεται ότι κάτι τέτοιο δεν μπορεί να συμβεί.

Τέλος, τα Smart Contracts αποτελούν ένα ειδικό τύπο λογαριασμού στο Ethereum

A.2.3.III. Λογαριασμοί (Accounts)

Ένας λογαριασμός στο Ethereum είναι μια οντότητα με κάποιο υπόλοιπο σε ETH και τη δυνατότητα να εκτελεί συναλλαγές. Οι πληροφορίες των οντοτήτων είναι αποθηκευμένες στο EVM[24]. Στο Ethereum υπάρχουν δύο τύποι λογαριασμού, οι **Ιδιωτικοί Λογαριασμοί** (Externally Owned Accounts – EOA) και τα **Έξυπνα Συμβόλαια** (Smart Contracts).

Ένας EOA είναι ένας λογαριασμός με κάποια διεύθυνση που ελέγχεται από κάποιο χρήστη μέσω ιδιωτικού κλειδιού (βλ παράγραφο A.1.2 Ασύμμετρη Κρυπτογραφία (Asymmetric Cryptography), σελ.14). Ένα άλλο κύριο γνώρισμα ενός EOA είναι ότι η δημιουργία του δεν κοστίζει κάτι. Μέσω του EOA ένας χρήστης μπορεί να λάβει ή να αποστείλει ETH σε άλλους λογαριασμούς ή να αλληλεπιδράσει με κάποιο Smart Contract[31].

Για τη δημιουργία ενός λογαριασμού αρχικά επιλέγεται ένα τυχαίο ιδιωτικό κλειδί (random private key) μήκους 64 χαρακτήρων του δεκαεξαδικού συστήματος (64 hex characters). Χρησιμοποιώντας τον αλγόριθμο Ελλειπτικής Καμπύλης για ψηφιακή υπογραφή (Elliptic Curve Digital Signature Algorithm) προκύπτει το δημόσιο κλειδί[31].

Η **Δημόσια Διεύθυνση** (Public Address) ενός λογαριασμού στο Ethereum προκύπτει προσθέτοντας “0x” μπροστά από τα 20 τελευταία Bytes της εξόδου της συνάρτησης κατακερματισμού Keccak-256 όταν σε αυτή δοθεί σαν είσοδος το δημόσιο κλειδί[31].

Για να πραγματοποιήσει αίτημα συναλλαγής ένας EOA λογαριασμός πρέπει να υπογράψει τη συναλλαγή με το ιδιωτικό του κλειδί (όπως περιεγράφηκε στην παράγραφο A.1.2 Ασύμμετρη Κρυπτογραφία (Asymmetric Cryptography), σελ. 14). Κατ’ αυτόν τον τρόπο, οι υπόλοιποι κόμβοι του δικτύου μπορούν να επιβεβαιώσουν η διεύθυνση του αποστολέα της συναλλαγής ταυτίζεται με αυτή του EOA λογαριασμού και υπάρχει η συγκατάθεσή του για να γίνει η συναλλαγή[31].

Ο δεύτερος τύπος λογαριασμών στο Ethereum είναι τα Smart Contracts. Τα smart contracts έχουν δικιά τους διεύθυνση, υπόλοιπο σε ETH και μπορούν να κάνουν συναλλαγές, να δημιουργήσουν νέα smart contracts ή να αλληλεπιδράσουν με ήδη υπάρχοντα. Η διαφορά τους από τους EOA είναι ότι δεν ελέγχονται από κάποιο χρήστη αλλά λειτουργούν ανάλογα με το πως έχουν προγραμματιστεί εκ των προτέρων. Παράλληλα η δημιουργία τους έχει κάποιο κόστος διότι καταναλώνουν χώρο του δικτύου[31].

Τα smart contracts έχουν και αυτά **Δημόσια Διεύθυνση** (Public Address) η οποία προκύπτει από τη διεύθυνση του λογαριασμού που τα δημιουργεί και τον αριθμό συναλλαγών που αυτός έχει πραγματοποιήσει (nonce). Επίσης, εφόσον δημοσιοποιηθούν στο δίκτυο δεν μπορούν να τροποποιηθούν ή να διαγραφούν[27].

A.2.3.IV. Dapps

Η ευελιξία που προσφέρουν τα smart contracts στο Ethereum επιτρέπουν τη δημιουργία αποκεντρωμένων εφαρμογών που ονομάζονται Decentralized Applications - Dapps. Ένα Dapp είναι ο συνδυασμός smart contracts που υπάρχουν σε ένα Blockchain μαζί με ένα γραφικό περιβάλλον διεπαφής χρήστη (graphical user interface-GUI)[32]. Σε ένα Dapp, τον παραδοσιακό ρόλο του διακομιστή (server) που χειρίζεται τα αιτήματα του χρήστη, αναλαμβάνει το ίδιο το Blockchain[10].

Η ιδέα μια αποκεντρωμένης εφαρμογής δεν είναι καινούργια αλλά υπήρχε εδώ και αρκετά χρόνια με τις εφαρμογές Torrent. Η καινοτομία των σύγχρονων Dapps είναι η σύνδεσή τους με ένα Blockchain[33].

Εκ πρώτης όψεως Dapp μοιάζει με μία συνηθισμένη Web εφαρμογή αφού το front-end και των δύο τύπων εφαρμογών μπορεί να δημιουργηθεί με τα ίδια εργαλεία. Η διαφοροποίηση γίνεται στο back-end της εφαρμογής. Οι συνηθισμένες Web εφαρμογές πραγματοποιούν

αιτήματα σε μια βάση δεδομένων μέσω ενός Application Programming Interface – API και ο κώδικας τρέχει σε κάποιο κεντρικό διακομιστή (server). Για την ταυτοποίηση του χρήστη κατά την είσοδο στην εφαρμογή απαιτούνται κωδικοί οι οποίοι αντιπαραβάλλονται με τους κωδικούς που αποθηκεύτηκαν στη βάση κατά τη δημιουργία των λογαριασμών. Κάτι τέτοιο προϋποθέτει την ανάγκη εμπιστοσύνης των χρηστών σε ένα κεντρικό σύστημα αποθήκευσης. Αυτή η μέθοδος παρουσιάζει τρωτότητα αφού αν κάποιος κακόβουλος χρήστης αποκτήσει πρόσβαση στα δεδομένα της βάσης, αποκτά πρόσβαση και στους κωδικούς (στα Hashes των κωδικών)[33].

Στην περίπτωση των Dapps το back-end υλοποιείται με ένα ή περισσότερα smart contracts που αλληλεπιδρούν με το Blockchain[33]. Ο κώδικας των Dapps είναι, συνήθως, open-source και η λειτουργία τους δεν ελέγχεται από μία κεντρική οντότητα. Επίσης, τα δεδομένα της εφαρμογής είναι κρυπτογραφημένα και αποθηκευμένα στο Blockchain και δεν μπορούν να διαγραφούν ή να παραποιηθούν[34].

Πλεονεκτήματα[35], [36]

- Δεν είναι ευάλωτα σε επιθέσεις άρνησης εξυπηρέτησης (Denial of Service-DoS Attacks) διότι δεν υπάρχει ένας κεντρικός διακομιστής (server). Άπαξ και δημοσιοποιηθεί στο Blockchain ένα Smart Contract είναι προσβάσιμο σε όλους τους χρήστες συνεχώς.
- Τα δεδομένα των Dapps δεν μπορούν να τροποποιηθούν ή να διαγραφούν με κανένα τρόπο.
- Δεν μπορεί να υπάρξει λογοκρισία, τα δεδομένα αποθηκεύονται στο Blockchain και δεν μπορούν να τροποποιηθούν.
- Διατήρηση ανωνυμίας των χρηστών.
- Δεν υπάρχει ανάγκη εμπιστοσύνης σε ένα κεντρικό σύστημα διαχείρισης της εφαρμογής.

Μειονεκτήματα[35], [36]

- Όπως αναφέρθηκε στην παράγραφο Α.2.3.Π Έξυπνα συμβόλαια (Smart Contracts), σελ.25, τα συμβόλαια αυτά εφόσον δημοσιευτούν στο Blockchain δεν μπορούν να τροποποιηθούν. Επομένως, είναι απαραίτητο να σχεδιαστούν πολύ προσεκτικά και να περάσουν από εντατικούς ελέγχους.
- Δεν είναι εύκολη η συντήρηση αυτών των εφαρμογών διότι δεν μπορούν να τροποποιηθούν τα smart contracts και είναι δύσκολο να γίνουν αναβαθμίσεις.
- Δεν προσφέρουν καλή κλιμακωσιμότητα. Ο συνολικός όγκος των δεδομένων του Blockchain συνεχώς αυξάνεται και αυτό έχει αρνητική επίδραση στην ταχύτητα των εφαρμογών.
- Στην προσπάθεια τους να είναι φιλικές προς τους χρήστες κάποιες εφαρμογές χρησιμοποιούν κεντρικούς διακομιστές χάνοντας έτσι το νόημα της αποκεντρωμένης εφαρμογής.
- Η δημιουργία των Dapps βρίσκεται ακόμα σε πρώιμο στάδιο. Δεν υπάρχει ακόμα αρκετή εμπειρία και δεν έχουν δοκιμαστεί σε μεγάλη κλίμακα για μεγάλο χρονικό διάστημα.

A.3. ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

A.3.1. Επιβλεπόμενη Μάθηση (Supervised Learning)

Η **Μηχανική Μάθηση** (Machine Learning) χωρίζεται σε τρεις βασικές κατηγορίες, την **Επιβλεπόμενη Μάθηση** (Supervised Learning), τη **Μη-Επιβλεπόμενη** (Unsupervised Learning) και την **Ενθαρρυντική ή Ενισχυτική Μάθηση** (Reinforcement Learning).

Η **Επιβλεπόμενη Μάθηση** επιτυγχάνεται μέσα από δεδομένα των οποίων το επιθυμητό αποτέλεσμα είναι γνωστό και η κατηγοριοποίησή τους ξεκάθαρη και προκαθορισμένη. Αντιθέτως, η **Μη-Επιβλεπόμενη Μάθηση** επιτυγχάνεται με δεδομένα που δεν διαθέτουν γνωστό αποτέλεσμα. Ο στόχος μπορεί να είναι για παράδειγμα, η ομαδοποίησή τους βάσει των χαρακτηριστικών τους[37]. Κατά την **Ενθαρρυντική Μάθηση** δεν υπάρχει συγκεκριμένο σύνολο δεδομένων. Αντί αυτού, ένας πράκτορας (agent) αλληλεπιδρά με ένα περιβάλλον και εκπαιδεύεται βάσει της ανατροφοδότησης (θετικής ή αρνητικής) που παίρνει για τις πράξεις του[38].

Πιο συγκεκριμένα, κατά την επιβλεπόμενη μάθηση χρησιμοποιείται ένα σύνολο δεδομένων τέτοιο ώστε για κάθε δείγμα να υπάρχει μία ετικέτα εξόδου (label). Οι πιθανές τιμές των ετικετών είναι συνήθως συγκεκριμένες και διακριτές (αλλά μπορούν να είναι και συνεχείς). Ένα κλασσικό παράδειγμα συνόλου δεδομένων επιβλεπόμενης μάθησης είναι το MNIST[39]. Το MNIST αποτελείται από μαυρόασπρες εικόνες μεγέθους 28x28 pixel που αναπαριστούν ψηφία από το 0 ως το 9. Ο στόχος της εκπαίδευσης ενός Νευρωνικού Δικτύου χρησιμοποιώντας το MNIST είναι να δέχεται νέες εικόνες ιδίου μεγέθους και να αναγνωρίζει το ψηφίο που αναπαριστούν.

A.3.2. Επεξεργασία Φυσικής Γλώσσας (NLP – Natural Language Processing)

Ο μεγάλος όγκος γραπτών ή ηχητικών κειμένων που υπάρχει και συνεχώς αυξάνεται ήταν η αφορμή για τη δημιουργία του κλάδου της Επεξεργασίας Φυσικής Γλώσσας ή NLP. Το NLP μελετά την αλληλεπίδραση μεταξύ μηχανών και ανθρώπων χρησιμοποιώντας φυσική γλώσσα[37]. Αποτελεί ένα συνδυασμό των κλάδων της Γλωσσολογίας και της Μηχανικής Μάθησης[40].

Ο στόχος του είναι να χρησιμοποιήσει την υπολογιστική ισχύ των μηχανών για να μελετά και να αναλύει κείμενα με τρόπο παρόμοιο με τον ανθρώπινο. Οι εφαρμογές του NLP ποικίλουν και χρησιμοποιείται μεταξύ άλλων για τη μετάφραση κειμένων, την αναγνώριση φωνής, την ανίχνευση συναισθημάτων αλλά και τη τεχνητή δημιουργία κειμένων σε φυσική γλώσσα[40].

Για την επεξεργασία της φυσικής γλώσσας, οι λέξεις ή τμήματά τους αναπαρίστανται ως **διανύσματα** (word vectors). Η απεικόνιση των λέξεων σε διανύσματα γίνεται μέσω μίας διαδικασίας που ονομάζεται **ενσωμάτωση** (embedding)

Οι αρχικές μέθοδοι απεικόνισης λέξεων σε διανύσματα είχαν τον περιορισμό ότι μία λέξη απεικονιζόταν πάντα με το ίδιο διάνυσμα. Με άλλα λόγια, η συνάρτηση απεικόνισης της λέξης χρησιμοποιούσε ως είσοδο μόνο την ίδια τη λέξη. Με αυτό τον τρόπο όμως χανόταν η έννοια της αμφισημίας των λέξεων. Για παράδειγμα, για τις προτάσεις «ουρά του σκύλου» και «ουρά στο ταμείο», η λέξη «ουρά» θα ήταν καλό να αναπαρασταθεί με διαφορετικό τρόπο διότι έχει διαφορετικό νόημα. Προέκυψε λοιπόν η ανάγκη για ένα μοντέλο απεικόνισης που να λάμβανε υπόψη και τα συμφραζόμενα της λέξης. Ένα μοντέλο απεικόνισης ευαίσθητο στα συμφραζόμενα της λέξης είναι το BERT.

A.3.3. BERT - Bidirectional Encoder Representations from Transformers

Το BERT[41] είναι ένα μοντέλο αναπαράστασης φυσικής γλώσσας για την προεκπαίδευση (pre-training) μοντέλων NLP. Το BERT ξεχωρίζει χάρη σε δύο βασικά χαρακτηριστικά, το πρώτο είναι ότι η απεικόνιση των λέξεων γίνεται με **ευαισθησία στα συμφραζόμενα** (context-sensitive) και το δεύτερο ότι είναι διαθέτει **σταθερή αρχιτεκτονική δομή** (task agnostic architecture)[37].

Όπως περιγράφηκε και στην παράγραφο A.3.2 Επεξεργασία Φυσικής Γλώσσας (NLP – Natural Language Processing), σελ.29, μια λέξη μπορεί να παρουσιάζει αμφισημία επομένως είναι σημαντικό κατά την απεικόνισή της σε διάνυσμα, να λαμβάνονται υπόψη και τα συμφραζόμενά της. Το BERT προχωρά ένα βήμα παρακάτω σε σχέση με άλλες context-sensitive απεικονίσεις διότι λαμβάνει υπόψη τις προηγούμενες αλλά και τις επόμενες λέξεις της πρότασης (bidirectional pre-training). Όπως είναι φυσικό, η αμφίδρομη χρησιμοποίηση των συμφραζομένων δίνει καλύτερη αναπαράσταση της έννοιας της λέξης άρα και καλύτερα αποτελέσματα γενικότερα[41].

Σχετικά με τη δομή ενός μοντέλου αναπαράστασης φυσικής γλώσσας υπάρχουν δύο προσεγγίσεις. Η πρώτη είναι η feature-based προσέγγιση όπου η δομή του μοντέλου διαφοροποιείται πλήρως ανάλογα με το πρόβλημα. Η δεύτερη είναι η fine-tuning προσέγγιση όπου η δομή του μοντέλου παραμένει σταθερή ανεξάρτητα από το πρόβλημα. Αυτό που αλλάζει είναι το τελευταίο στρώμα (layer) του μοντέλου και γίνονται βελτιστοποιήσεις αλλάζοντας την τιμή κάποιων παραμέτρων[41].

Το BERT χρησιμοποιεί τη fine-tuning προσέγγιση και η εκπαίδευσή του χωρίζεται στην αρχική φάση της **προεκπαίδευσης** (pre-training) και την τελική φάση της **προσαρμογής** (fine-tuning)[41].

Κατά τη φάση της προεκπαίδευσης γίνονται δύο εργασίες. Η πρώτη είναι η **Μοντελοποίηση Κρυμμένων Λέξεων** (Masked Language Modeling) όπου επιλέγονται τυχαία κάποιες λέξεις και αποκρύπτονται. Στη συνέχεια το BERT προσπαθεί να τις μαντέψει βάσει των υπόλοιπων λέξεων των προτάσεων στις οποίες ανήκουν. Η δεύτερη είναι η **Πρόβλεψη Επόμενης Πρότασης** (Next Sentence Prediction). Σε αυτή το BERT λαμβάνει ένα ζεύγος προτάσεων και προσπαθεί να μαντέψει αν η δεύτερη πρόταση που έλαβε είναι η αμέσως επόμενη της πρώτης στο αρχικό κείμενο. Έτσι το μοντέλο BERT εκπαιδεύεται λαμβάνοντας υπόψη τα συμφραζόμενα των λέξεων[37].

Κατά τη φάση της προσαρμογής, προσαρμόζεται το μοντέλο ώστε να δέχεται τα δεδομένα εισόδου ενός συγκεκριμένου προβλήματος και να παράγει αποτελέσματα επιθυμητής μορφής. Έπειτα προσαρμόζονται οι τιμές των παραμέτρων ώστε να βελτιστοποιηθεί το αποτέλεσμα. Ως αποτέλεσμα, τα τελικά μοντέλα BERT που χρησιμοποιούνται για διαφορετικές εφαρμογές έχουν πολλά κοινά μεταξύ τους[41].

A.4. ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ ΚΑΙ BLOCKCHAIN

A.4.1. Υφιστάμενες δουλειές

Όπως φάνηκε και στην εισαγωγή, υπάρχουν πολλά πλαίσια γύρω από τα οποία αυτές οι τεχνολογίες της Μηχανικής Μάθησης και του Blockchain θα μπορούσαν να συνδυαστούν και συνεχώς προκύπτουν και καινούργια. Η πιο δημοφιλής μέθοδος συνδυασμού φαίνεται να είναι η εκμετάλλευση του Blockchain για την εκτέλεση παράλληλων εργασιών Μηχανικής Μάθησης.

Γενικά, υπάρχουν δύο κύριες προσεγγίσεις για την παραλληλοποίηση των εργασιών που εκτελούνται στη Μηχανική Μάθηση. Η μία είναι η **παραλληλοποίηση σε επίπεδο δεδομένων** και η δεύτερη είναι **παραλληλοποίηση σε επίπεδο μοντέλου**. Η τεχνολογία του Blockchain μπορεί να αξιοποιηθεί και στις δύο προσεγγίσεις. Πιο κάτω αναλύονται κάποιες υλοποιήσεις και προτάσεις που έχουν γίνει προς αυτή την κατεύθυνση και χρησιμοποιούν μία ή και τις δύο προσεγγίσεις.

A.4.1.1. Decentralized & Collaborative AI on Blockchain

Οι Harris J. και Waggoner B. [42], [43] πρότειναν ένα πλαίσιο (framework) με το οποίο οι κόμβοι ενός Blockchain μπορούν να μαζεύουν δεδομένα εκπαίδευσης ενός Νευρωνικού Δικτύου. Παράλληλα το μοντέλο του Νευρωνικού Δικτύου είναι προσβάσιμο σε όλους τους κόμβους και αναβαθμίζεται συνεχώς με την προσθήκη των δεδομένων. Αυτή η πρόταση εκμεταλλεύεται τόσο την παραλληλοποίηση σε επίπεδο δεδομένων όσο και την παραλληλοποίηση σε επίπεδο μοντέλου.

Το πλαίσιο ασχολείται αποκλειστικά με μοντέλα Επιβλεπόμενης Μάθησης. Οι υπολογισμοί για την εκπαίδευση των Νευρωνικών Δικτύων αλλά και η αποθήκευσή τους γίνεται on-chain, δηλαδή εντός κάποιου Smart Contract στο Blockchain. Λόγω της φύσης του Blockchain, η αποθήκευση δεδομένων σε αυτό έχει σημαντικό κόστος. Γι' αυτό προτιμήθηκε η χρήση απλών μοντέλων Νευρωνικών Δικτύων (Naïve Bayes, Nearest Centroid και Single Layer Perceptron) διότι δεν απαιτούν μεγάλο χώρο μνήμης για την αποθήκευσή τους και οι υπολογισμοί για την ενημέρωσή τους είναι εύκολοι. Ακόμη, χρησιμοποιήθηκαν απλά σύνολα δεδομένων όπως μικρές προτάσεις για τον ίδιο λόγο.

Τα μοντέλα που χρησιμοποιούνται έχουν ήδη προεκπαιδευτεί σε κάποιο βαθμό και έχει υπολογιστεί κάποια αρχική επίδοση βάσει κάποιας μετρικής (πχ accuracy). Παράλληλα υπάρχει η δυνατότητα ανάκτησης κάποιας πρότερης κατάστασης του ΝΔ στην περίπτωση που κακόβουλοι χρήστες καταφέρουν να το αλλοιώσουν με κακόβουλα δεδομένα εκπαίδευσης.

Ένα βασικό πρόβλημα που κλήθηκαν να αντιμετωπίσουν οι εισηγητές του πλαισίου είναι η δημιουργία ενός **μηχανισμού κινήτρου** (incentive mechanism) που θα ενθαρρύνει τους

κόμβους να προσφέρουν ποιοτικά δεδομένα για την εκπαίδευση του Νευρωνικού Δικτύου. Ο μηχανισμός πρέπει επίσης να αποθαρρύνει κάθε απόπειρα να διοχέτευσης κακών δεδομένων εκπαίδευσης στο Δίκτυο και κάθε απόπειρα παραποίησης του. Προτάθηκαν συνολικά τρεις πιθανοί μηχανισμοί κινήτρου.

Ο πρώτος δεν δίνει οικονομικά κίνητρα στους χρήστες και βασίζεται αποκλειστικά στην ευγενική πρόθεση των χρηστών να βοηθήσουν και να προσφέρουν καλά δεδομένα εκπαίδευσης. Οι χρήστες σε αυτή την περίπτωση επιβραβεύονται με πόντους ή ηλεκτρονικές κονκάρδες (badges) που επιβεβαιώνουν την προσφορά τους στο σύστημα. Οι χρήστες δεν έχουν μεν οικονομικά κίνητρα να βοηθήσουν αλλά δεν υπάρχει ούτε και οικονομικό κίνητρο σε κακόβουλους χρήστες να βλάψουν το σύστημα.

Ο δεύτερος μηχανισμός προβλέπει ότι υπάρχει ένας εξωτερικός οργανισμός που προσφέρει κάποια σύνολα δεδομένων για αξιολόγηση (test datasets) βάσει των οποίων αξιολογείται η προσφορά των χρηστών. Οι χρήστες καταβάλουν αρχικά κάποια προκαταβολή και σε περίπτωση που τα δεδομένα που προσέφεραν βελτίωσαν την επίδοση του ΝΔ τότε λαμβάνουν ένα ποσό μεγαλύτερο από την προκαταβολή ενώ σε αντίθετη περίπτωση χάνουν την προκαταβολή.

Ο τρίτος μηχανισμός ορίζει ότι κάθε χρήστης που καταβάλει κάποια δεδομένα Επιβλεπόμενης Μάθησης (έστω X) με τις αντίστοιχες ετικέτες (labels) (έστω Y) πρέπει να καταβάλει και κάποια προκαταβολή. Μετά από πέρας ορισμένου χρόνου, το ΝΔ προβλέπει την ετικέτα των δεδομένων (έστω $ND(X)$). Αν αυτή δεν συμπίπτει με την ετικέτα που έδωσε ο χρήστης (δηλ. $ND(X) \neq Y$) τα δεδομένα θεωρούνται αναξιόπιστα και χάνει την προκαταβολή. Αν συμπίπτει τότε ο χρήστης λαμβάνει την προκαταβολή που έδωσε και αποκτά το δικαίωμα να αποσπάσει και προκαταβολές άλλων χρηστών των οποίων τα δεδομένα κρίθηκαν αναξιόπιστα.

A.4.1.II. Decentralized and Distributed Machine Learning Model Training with Actors

Ο Addair T. στο [44] κάνει μια σύγκριση διαφορετικών προσεγγίσεων κατανεμημένης Μηχανικής Μάθησης. Για τον Addair, ο παραλληλισμός σε επίπεδο δεδομένων γίνεται όταν οι κόμβοι εκπαιδεύουν ξεχωριστά ίδια μοντέλα με διαφορετικά δεδομένα. Από την άλλη, ο παραλληλισμός σε επίπεδο μοντέλου γίνεται με τους κόμβους να εκπαιδεύουν διαφορετικά τμήματα (πχ στρώματα) του ιδίου μοντέλου. Στις δύο προσεγγίσεις που ακολουθεί συνδυάζει και τα δύο επίπεδα παραλληλισμού.

Η πρώτη προσέγγιση είναι η κεντροποιημένη κατανεμημένη Μηχανική Μάθηση όπου υπάρχει ένας κεντρικός διακομιστής (server) που διατηρεί μία και μοναδική κατάσταση των παραμέτρων του ΝΔ. Οι κόμβοι επικοινωνούν με τον διακομιστή και προτείνουν αλλαγές ώστε να μεταβάλλουν αυτές τις γενικές παραμέτρους.

Η δεύτερη προσέγγιση είναι η αποκεντρωμένη κατανεμημένη Μηχανική Μάθηση όπου δεν υπάρχει ένας κεντρικός διακομιστής. Οι κόμβοι αναμεταδίδουν στο δίκτυο τις αλλαγές στις παραμέτρους και λαμβάνουν με τη σειρά τους ενημερώσεις ώστε να παραμένουν ενήμεροι για τις τιμές των παραμέτρων. Για την αποφυγή συμφόρησης του δικτύου χρησιμοποιείται μια παράμετρος κατωφλίου τ η οποία καθορίζει την ελάχιστη μεταβολή στην τιμή μιας παραμέτρου ώστε να θεωρείται αξιόλογη για να μεταδοθεί στο δίκτυο. Έτσι μεταδίδονται

μόνο σημαντικές μεταβολές στις τιμές παραμέτρων και μειώνεται ο συνολικός αριθμός ενημερώσεων.

Από τις μετρήσεις που έγιναν προέκυψε το συμπέρασμα ότι η κεντριοποιημένη προσέγγιση δίνει μικρότερο σφάλμα από την αποκεντρωμένη. Ωστόσο, για κατάλληλη τιμή κατωφλίου τ φαίνεται ότι μπορεί να επιτευχθεί αρκετά μικρό σφάλμα με σημαντικά καλύτερη επίδοση από πλευράς χρόνου. Συγκεκριμένα, αποδεικνύεται ότι η αύξηση της τιμής του τ αυξάνει γραμμικά το μέσο τετραγωνικό σφάλμα αλλά μειώνει εκθετικά τις ενημερώσεις για αλλαγές στις τιμές των παραμέτρων άρα και το συνολικό χρόνο εκπαίδευσης. Συμπερασματικά, αποδεικνύεται ότι υπάρχει μια αντιστάθμιση μεταξύ του χρόνου εκπαίδευσης και της ακρίβειας του μοντέλου

A.4.1.III. Proof of Learning (PoLe)

Το Proof of Learning είναι μια ιδέα υλοποίησης ενός αλγορίθμου συναίνεσης (consensus algorithm) που βασίζεται στη Μηχανική Μάθηση και συγκεκριμένα την εκπαίδευση Νευρωνικών Δικτύων από τους Lan Y., Liu Y. Και Li B. [1]. Είναι μία πρόταση που εκμεταλλεύεται αποκλειστικά την παραλληλοποίηση σε επίπεδο μοντέλου αφού τα δεδομένα είναι κοινά και προσφέρονται από κάποιον εξωτερικό οργανισμό. Ιδιαίτερη αναφορά στο PoLe έγινε στην παράγραφο A.2.2.III Πρωτόκολλο Συναίνεσης (Consensus Protocol), σελ.19.

**B. ΕΝΟΤΗΤΑ
ΠΡΟΤΕΙΝΟΜΕΝΗΣ ΛΥΣΗΣ**

B.1. ΣΧΕΔΙΑΣΜΟΣ

B.1.1. Κεντρική Ιδέα

Στόχος της παρούσας διπλωματικής εργασίας ήταν η μελέτη του συνδυασμού των τεχνολογιών Blockchain και Μηχανικής Μάθησης και η δημιουργία ενός dApp που θα τα συνδυάζει. Πιο συγκεκριμένα, προτείνεται η χρήση ενός dApp, ώστε βελτιώνεται η επίδοση μοντέλων ΝΔ με συλλογή δεδομένων μέσω του δικτύου Blockchain και συνεχή εκπαίδευση τους. Η εργασία χρησιμοποιεί σκέψεις και συμπεράσματα από προηγούμενες προτάσεις και υλοποιήσεις αλλά προσφέρει και καινούργιες ιδέες.

Το τελικό dApp ονομάστηκε **DEMOS (Distributedly Enhanced Machine learning Optimization System)** ή αλλιώς **Αποκεντρωμένα Ενισχυμένο Σύστημα Βελτιστοποίησης Μηχανικής Μάθησης**. Το μοντέλο ΝΔ που επιλέχθηκε για το DEMOS dApp βασίζεται στο μοντέλο αναπαράστασης φυσικής γλώσσας BERT [38] που χρησιμοποιείται για την προεκπαίδευση (pre-training) μοντέλων NLP. Πιο συγκεκριμένα, το μοντέλο χρησιμοποιείται για NLP προβλήματα που ανήκουν στην κατηγορία της ανάλυσης συναισθημάτων (Sentiment Analysis).

Η γενική ιδέα του dApp είναι η εξής:

1. Ένας κόμβος αιτείται τη συλλογή δεδομένων εκπαίδευσης για ένα μοντέλο ΝΔ.
2. Το σύνολο των κόμβων του δικτύου Blockchain προσφέρει νέα δεδομένα εκπαίδευσης.
3. Το μοντέλο εκπαιδεύεται με τα δεδομένα που συλλέχθηκαν και επαναξιολογείται η ακρίβειά του.

Η συλλογή δεδομένων και εκπαίδευση του μοντέλου ΝΔ είναι συνεχής έτσι το μοντέλο βελτιώνεται συνεχώς.

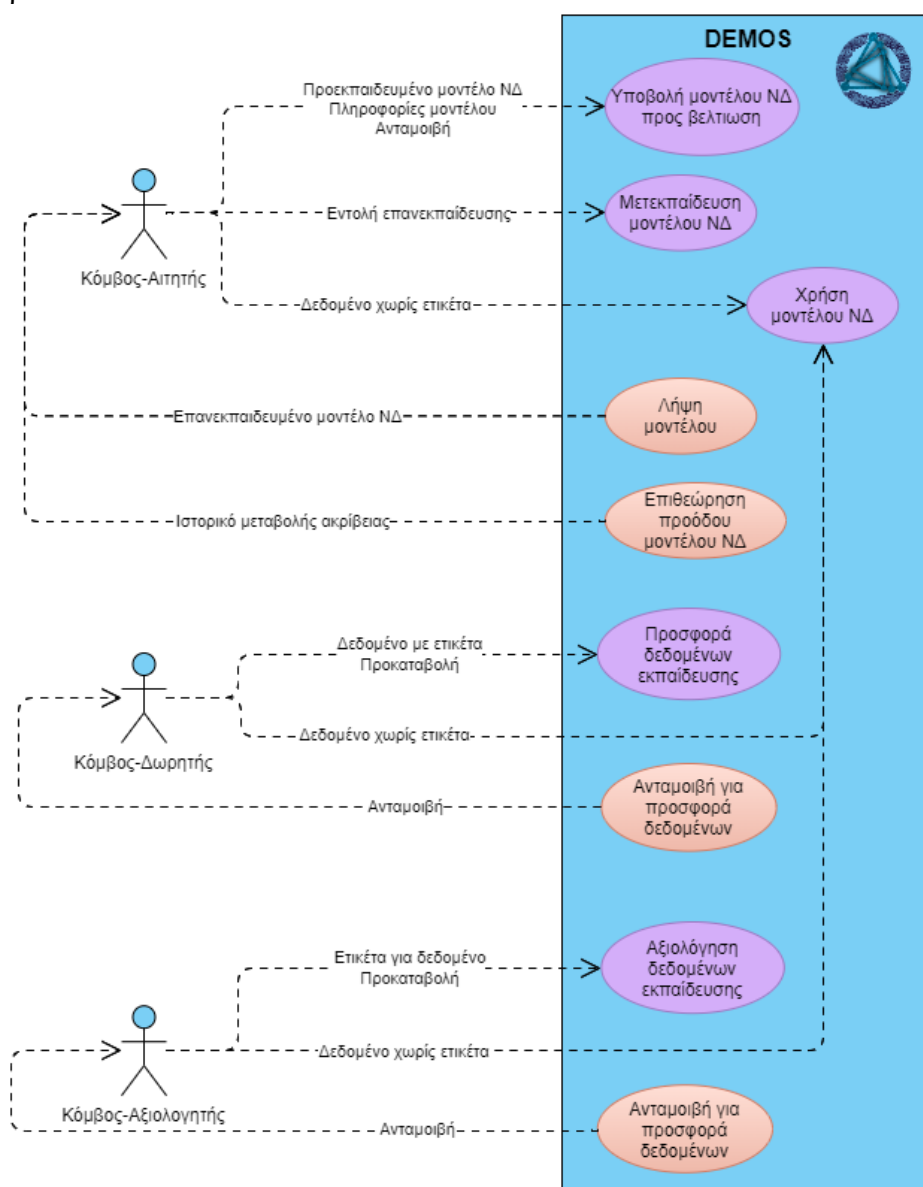
Υπάρχουν τρεις τύποι κόμβων (όπως φαίνεται στο Διάγραμμα 1) στο δίκτυο Blockchain και είναι οι εξής:

- **Κόμβοι-Αιτητές:** Αυτοί οι κόμβοι δημοσιεύουν στο dApp ένα προεκπαιδευμένο μοντέλο ΝΔ του οποίου θέλουν να βελτιστοποιήσουν την ακρίβεια. Περιγράφουν επίσης τον τύπο δεδομένων που χρησιμοποιεί το μοντέλο για να εκπαιδευτεί και τις πιθανές κατηγορίες ετικετών (πχ Τύπος δεδομένων: *προτάσεις στις οποίες να εκφράζεται κάποιο συναίσθημα*, Πιθανές ετικέτες: «αγάπη», «θυμός», «θλίψη», «φόβος»). Έπειτα καταβάλλουν ένα ποσό το οποίο θα μοιραστεί ως ανταμοιβή στους κόμβους που θα προσφέρουν και θα αξιολογήσουν δεδομένα.
- **Κόμβοι-Δωρητές:** Αυτοί οι κόμβοι προσφέρουν δεδομένα επιβλεπόμενης εκπαίδευσης με τις αντίστοιχες ετικέτες σύμφωνα με τις οδηγίες που έχουν δώσει οι Κόμβοι-Αιτητές. Καταβάλλουν επίσης ένα ποσό ως προκαταβολή. Σε περίπτωση που τα δεδομένα που προσέφεραν θεωρηθούν καλά τότε λαμβάνουν ένα ποσό μεγαλύτερο από την προκαταβολή που κατέβαλαν.

- **Κόμβοι-Αξιολογητές:** Τα δεδομένα που προσέφεραν οι Κόμβοι-Δωρητές δημοσιοποιούνται στο dApp αλλά αποκρύπτεται η ετικέτα που πρότειναν. Οι Κόμβοι-Αξιολογητές ψηφίζουν για κάθε δεδομένο ποια από τις πιθανές ετικέτες του μοντέλου τους ταιριάζει καλύτερα. Κάθε ψήφος απαιτεί κάποια προκαταβολή. Όταν μία ετικέτα για ένα δεδομένο θεωρηθεί επικυρωμένη (όπως περιγράφεται πιο κάτω στην παράγραφο B.1.2 Μηχανισμός Κινήτρου (Incentive Mechanism), σελ. 37) τότε όλοι οι Κόμβοι-Αξιολογητές που ψήφισαν αυτή την επιλογή λαμβάνουν ένα ποσό μεγαλύτερο από την προκαταβολή που κατέβαλαν. Όλοι οι υπόλοιποι Κόμβοι-Αξιολογητές χάνουν την προκαταβολή.

Ένας κόμβος μπορεί να είναι ταυτόχρονα και Αιτητής και Δωρητής και Αξιολογητής αλλά υπάρχουν κάποιοι περιορισμοί:

- Ένας κόμβος δεν μπορεί να αξιολογήσει ένα δεδομένο που προσέφερε ο ίδιος ως δωρητής
- Ένας κόμβος δεν μπορεί να αξιολογήσει ένα δεδομένο ψηφίζοντας πάνω από μία φορά



Διάγραμμα 1: Αλληλεπίδραση κόμβων-DEMOS

B.1.2. Μηχανισμός Κινήτρου (*Incentive Mechanism*)

Για τη σωστή λειτουργία του dApp ήταν απαραίτητος ο σχεδιασμός ενός μηχανισμού κινήτρου που θα ενθαρρύνει τους κόμβους να προσφέρουν πολλά και ποιοτικά δεδομένα για τα μοντέλα. Επίσης θα αποθαρρύνει τους κακόβουλους κόμβους από το να νοθεύσουν το σύστημα με κακής ποιότητας δεδομένα και να επηρεάσουν αρνητικά την επίδοσή του. Τελικά χρησιμοποιήθηκε ένας μηχανισμός με οικονομικά κίνητρα.

Τα δεδομένα για να είναι ποιοτικά πρέπει να περνούν από κάποιο ποιοτικό έλεγχο. Πρέπει δηλαδή είναι να υπάρχει κάποιος που να αξιολογεί την ποιότητα των δεδομένων. Όπως όμως έχει προαναφερθεί, η ουσία του Blockchain είναι πως οι κόμβοι που το συντελούν δημιουργούν ένα περιβάλλον στο οποίο δεν υπάρχει αξιοπιστία μεταξύ τους (trustless environment). Θεωρούμε λοιπόν δεδομένο ότι ο κόμβος που προσφέρει τα δεδομένα δεν μπορεί να θεωρηθεί αξιόπιστος.

Ο κόμβος που έχει αιτηθεί τη συλλογή δεδομένων για την εκπαίδευση του μοντέλου ΝΔ θα μπορούσε θεωρητικά να κρίνει τα νέα δεδομένα αφού δεν έχει λόγο να θέλει να βλάψει την επίδοση του μοντέλου του με μη ποιοτικά δεδομένα. Από την άλλη αυτό δεν προσφέρει καλή κλιμάκωση. Ο στόχος είναι η συλλογή όσο μεγαλύτερου όγκου δεδομένων γίνεται οπότε θα ήταν παράλογο να αξιολογεί ένας μονάχα κόμβος ένα-ένα τα δεδομένα και χάνεται το νόημα της αποκεντροποίησης.

Το ίδιο το μοντέλο θα μπορούσε να θεωρηθεί αμερόληπτο αλλά δεν είναι σε θέση να ξεχωρίσει με απόλυτο τρόπο τα ποιοτικά από τα μη ποιοτικά δεδομένα όπως προτείνει το [42] γιατί παρόλο που έχει κάποια προεκπαίδευση, παραμένει σε φάση εκπαίδευσης και ποτέ δεν έχει ακρίβεια 100%. Αυτό μπορεί να εξηγηθεί με το εξής παράδειγμα: Έστω ένα μοντέλο ΝΔ το οποίο δέχεται εικόνες από ζώα και τα κατηγοριοποιεί σε γάτες και σκύλους και αξιολογεί μόνο του τα δεδομένα πριν τα χρησιμοποιήσει για να εκπαιδευτεί. Έστω επίσης ότι ένας κόμβος προσφέρει ένα νέο σύνολο δεδομένων εκπαίδευσης που περιέχει εικόνες από κάποια ράτσα σκύλων που μοιάζει με γάτες. Το μοντέλο με βάση την προεκπαίδευση που έχει, κατηγοριοποιεί τις εικόνες λανθασμένα ως γάτες. Ως αποτέλεσμα, πρώτον ο κόμβος χάνει την προκαταβολή που κατέβαλε διότι τα δεδομένα του θεωρούνται μη ποιοτικά. Δεύτερον, το μοντέλο δεν μαθαίνει ποτέ να κατηγοριοποιεί σωστά αυτή τη ράτσα ως σκύλους και πιθανότατα θα χαρακτηρίσει και στο μέλλον αντίστοιχα δεδομένα ως μη ποιοτικά.

Για την αντιμετώπιση αντίστοιχου προβλήματος στο πρωτόκολλο συναίνεσης (τόσο στο PoW όσο και στο PoS) χρησιμοποιείται η προσέγγιση της κυριαρχίας της πλειοψηφίας όπου όσο το 51% των κόμβων είναι αξιόπιστο τότε το Blockchain παραμένει ασφαλές. Με βάση αυτή την ιδέα επιλέχθηκε ποιοτικός έλεγχος των δεδομένων που βασίζεται στην πλειοψηφία σε κάποια ψηφοφορία.

Πιο συγκεκριμένα:

1. Ο Κόμβος-Αιτητής ενός συγκεκριμένου μοντέλου έχει θέσει ένα αριθμό ψήφων M που χρειάζεται ένα δεδομένο για να θεωρηθεί επικυρωμένο.

2. Ένας Κόμβος-Δωρητής προσφέρει κάποιο δεδομένο (caption) με την αντίστοιχη ετικέτα E (label) (για παράδειγμα, δεδομένο: «η Ελένη φοβάται το σκοτάδι» και ετικέτα «φόβος»).
3. Το δεδομένο παρουσιάζεται στο dApp και οι Κόμβοι-Αξιολογητές ψηφίζουν ποια από τις πιθανές ετικέτες (για παράδειγμα, «αγάπη», «θυμός», «θλίψη», «φόβος») ταιριάζει καλύτερα στο δεδομένο.
4. Όταν κάποια πιθανή ετικέτα E' μαζέψει M ψήφους τότε το δεδομένο θεωρείται πλέον επικυρωμένο και μπορεί να χρησιμοποιηθεί για εκπαίδευση.
5. Αν $E = E'$ τότε θεωρείται ότι ο Κόμβος-Δωρητής έκανε καλή δωρεά και ανταμείβεται, αλλιώς θεωρείται ότι έκανε κακή δωρεά και χάνει το ποσό της προκαταβολής
6. Όλοι οι Κόμβοι-Αξιολογητές που ψήφισαν την ετικέτα E' ανταμείβονται ενώ όσοι ψήφισαν κάτι άλλο χάνουν το ποσό της προκαταβολής.

Σημείωση: Η μορφή ποιοτικού ελέγχου των δεδομένων είναι ανεξάρτητη του μοντέλου και το μοντέλο εκπαιδεύεται μόνο με δεδομένα που έχουν επικυρωθεί. Αυτό σημαίνει ότι το DEMOS dApp θα μπορούσε να εκπαιδεύσει και μοντέλα που δεν έχουν προεκπαιδευτεί καθόλου αλλά κατά τις δοκιμές του συστήματος δεν δοκιμάστηκαν τέτοια μοντέλα.

B.1.3. Μοντέλο Νευρωνικού Δικτύου

Το μοντέλο ΝΔ που επιλέχθηκε για το DEMOS dApp βασίζεται στο μοντέλο αναπαράστασης φυσικής γλώσσας BERT [41] που χρησιμοποιείται για την προεκπαίδευση (pre-training) μοντέλων NLP. Πιο συγκεκριμένα, το μοντέλο χρησιμοποιείται για NLP προβλήματα που ανήκουν στην κατηγορία της **ανάλυσης συναισθημάτων** (Sentiment Analysis) όπως προτείνεται στο [45].

Η αποθήκευση δεδομένων στο Blockchain έχει κάποιο κόστος το οποίο αυξάνεται ανάλογα με το μέγεθός τους. Έτσι επιλέχθηκε ένα μοντέλο NLP επιβλεπόμενης μάθησης γιατί τα δεδομένα του αποτελούνται από μικρές προτάσεις με κάποια σχετική ετικέτα.

Το μοντέλο ΝΔ που χρησιμοποιείται ονομάζεται bert-base-uncased [46] έχει κατασκευαστεί από την ομάδα ανάπτυξης λογισμικού Τεχνητής Νοημοσύνης Hugging Face [47]. Το μοντέλο περιλαμβάνεται στη βιβλιοθήκη Transformers και έχει προεκπαιδευτεί με δεδομένα από βιβλία και λήμματα της Wikipedia. Το bert-base-uncased δεν διαχωρίζει κεφαλαία και πεζά γράμματα και περιλαμβάνει λεξικό μεγέθους 30000 λέξεων. Η δομή του όπως φαίνεται στην Εικόνα 1 περιλαμβάνει ένα στρώμα με το BertMainLayer 109482240 παραμέτρων, ένα στρώμα Dropout και ένα πυκνό στρώμα ταξινομητή (Classifier Dense Layer) 2307 παραμέτρων για την έξοδο.

```
[ ] model.summary()
```

```
Model: "tf_bert_for_sequence_classification"
```

Layer (type)	Output Shape	Param #
bert (TFBertMainLayer)	multiple	109482240
dropout_37 (Dropout)	multiple	0
classifier (Dense)	multiple	2307

```
=====  
Total params: 109,484,547  
Trainable params: 109,484,547  
Non-trainable params: 0  
=====
```

Εικόνα 1: Δομή μοντέλου ΝΔ bert-base-uncased

Για κάθε διαφορετικό μοντέλο του DEMOS dApp, αρχικά χρησιμοποιείται ένα σύνολο δεδομένων το οποίο κάνει κάποια αρχική εκπαίδευση σε ένα αντίγραφο της πιο πάνω ακατέργαστης μορφή του προεκπαιδευμένου μοντέλου bert-base-uncased. Έπειτα το μοντέλο μπορεί να εισαχθεί στο dApp για μετεκπαίδευση. Πρέπει να σημειωθεί ότι για να γίνει η εκπαίδευση του μοντέλου τα δεδομένα πρέπει να περάσουν πρώτα από τον Tokenizer [48] που περιλαμβάνεται επίσης στη βιβλιοθήκη Transformers.

Σε κάθε ξεχωριστή περίπτωση το σύνολο δεδομένων χωρίζεται σε 3 τμήματα για training, validation και testing. Το μοντέλο αρχικά προεκπαιδεύεται χρησιμοποιώντας το τμήμα για training και το προεκπαιδευμένο μοντέλο που προκύπτει αποθηκεύεται στο Google Drive μαζί με τα τμήματα validation και testing.

Στη συνέχεια το μοντέλο καταχωρείται στο dApp μαζί με τις αντίστοιχες οδηγίες για τους Κόμβους-Δωρητές και Αξιολογητές. Εκεί οι κόμβοι του Blockchain μπορούν να προσθέσουν νέα δεδομένα του ίδιου τύπου για να εμπλουτίσουν το σύνολο δεδομένων και να βελτιώσουν την ακρίβεια του μοντέλου.

Κατά τη φάση της εκπαίδευσης, ανακτώνται τα τμήματα validation και training. Το τμήμα validation φορτώνεται και χρησιμοποιείται κατά την μετεκπαίδευση του μοντέλου ενώ το τμήμα testing χρησιμοποιείται για την επαναξιολόγηση του μοντέλου μετά από κάθε μετεκπαίδευση. Όταν ολοκληρωθεί η κάθε μετεκπαίδευση το ανανεωμένο μοντέλο αποθηκεύεται στο Google Drive αντικαθιστώντας το προηγούμενο.

Σημείωση: Δεν είναι απαραίτητο το ανανεωμένο μοντέλο να αντικαθιστά το προηγούμενο, θα μπορούσε να διατηρείται σαν μια νέα έκδοση αλλά για λόγω περιορισμένου χώρου στο Google Drive επιλέχθηκε να γίνεται αντικατάσταση.

B.1.3.I.Σύνολα Δεδομένων (Datasets)

Αν και το dApp δημιουργήθηκε με τρόπο που επιτρέπει την εκπαίδευση πολλών μοντέλων διαφορετικών συνόλων δεδομένων, οι δοκιμές έγιναν σε τρία συγκεκριμένα σύνολα δεδομένων που αντλήθηκαν από το Kaggle[49].

B.1.3.I.a. Twitter Tweets Sentiment Dataset

Το Twitter Tweets Sentiment σύνολο δεδομένων [50] περιλαμβάνει 27481 Tweets από χρήστες του Twitter και είναι κατηγοριοποιημένα σε 3 κατηγορίες (Positive, Negative και Neutral). Το σύνολο δεδομένων εκτός από τα ακατέργαστα Tweets με τις ετικέτες τους, περιλαμβάνει επίσης την επεξεργασμένη μορφή του κάθε Tweet διατηρώντας μόνο κάποιες σημαντικές λέξεις που χρησιμοποιούνται για την κατηγοριοποίηση. Ωστόσο αυτά τα δεδομένα παραλήφθηκαν εντελώς κατά την εκπαίδευση.

B.1.3.I.β. IMDB Movie Ratings Sentiment Analysis

Το IMDB Movie Ratings Sentiment Analysis σύνολο δεδομένων [51] περιλαμβάνει 39723 κριτικές ταινιών όπως αυτές καταχωρήθηκαν στο IDMB. Θεωρείται από τα πλέον κλασσικά παραδείγματα συνόλων δεδομένων για Sentiment Analysis. Σκοπός του μοντέλου είναι να δέχεται κριτικές ταινιών και να τις κατατάσσει σε αρνητικές ή θετικές (positive or negative) ανάλογα με το περιεχόμενό τους.

B.1.3.I.γ. Emotions Sentiment Analysis

Το Emotions Sentiment Analysis σύνολο δεδομένων [52] περιλαμβάνει 21405 μικρές προτάσεις των οποίων στο περιεχόμενο εκφράζεται κάποιο συναίσθημα. Οι ετικέτες

συναισθημάτων του συνόλου δεδομένων είναι λύπη, θυμός, αγάπη, έκπληξη, φόβος και χαρά (Sadness, Anger, Love, Surprise, Fear, Happiness). Το συγκεκριμένο σύνολο δεδομένων διαφέρει από τα προηγούμενα διότι έχει μεγαλύτερο αριθμό πιθανών ετικετών και δεν αρκείται στον απλό διαχωρισμό σε θετικά και αρνητικά.

B.1.3.II. Αξιολόγηση μοντέλων ΝΔ

Η αξιολόγηση της επίδοσης των μοντέλων ΝΔ γίνεται με τη χρήση της μετρικής **ακρίβειας** (Accuracy). Η συγκεκριμένη μετρική επιλέχθηκε διότι είναι ίσως η πιο διαδεδομένη μετρική αξιολόγησης μοντέλων ΝΔ γιατί δίνει μια καλή γενική εικόνα για την επίδοση του μοντέλου. Επίσης, τα σύνολα δεδομένων είναι ισορροπημένα (έχουν περίπου ίσα δείγματα για κάθε ετικέτα) κάτι που είναι σημαντικό ώστε η μετρική ακρίβειας να δίνει μια καλή εικόνα[53].

Η μετρική ακρίβειας ορίζεται ως το πηλίκο των ορθών προβλέψεων ως προς το σύνολο των προβλέψεων. Ο τύπος της είναι:

$$\text{Ακρίβεια} = \frac{\text{Αριθμός ορθών προβλέψεων}}{\text{Αριθμός συνολικών προβλέψεων}}$$

Η αξιολόγηση των μοντέλων γίνεται υπολογίζοντας την τιμή της ακρίβειας μετά από κάθε μετεκπαίδευση. Πιο συγκεκριμένα, πρώτα γίνεται η αρχική εκπαίδευση με ένα σύνολο δεδομένων όπως αυτά της παραγράφου Σύνολα Δεδομένων (Datasets), σελ.40. Μόλις ολοκληρωθεί γίνεται η πρώτη μέτρηση της ακρίβειας του μοντέλου. Έπειτα μέσω του DEMOS μαζεύονται, ανά διαστήματα, δεδομένα τα οποία χρησιμοποιούνται για την επανεκπαίδευση του μοντέλου. Αφότου ολοκληρωθεί κάθε κύκλος επανεκπαίδευσης, υπολογίζεται και αποθηκεύεται η νέα μέτρηση της ακρίβειας του μοντέλου.

Για να υπάρξει συνέπεια στη σύγκριση της ακρίβειας ενός μοντέλου ΝΔ μεταξύ των επανεκπαιδεύσεων πρέπει αυτή να υπολογίζεται κάθε φορά χρησιμοποιώντας το ίδιο σύνολο δεδομένων αξιολόγησης (test set). Για αυτό το λόγο κατά την αρχική εκπαίδευση κάθε μοντέλου, αποκόπτεται και αποθηκεύεται ένα τμήμα του συνόλου δεδομένων, το τμήμα test. Έπειτα, μετά από κάθε επανεκπαίδευση ο υπολογισμός της τιμής της ακρίβειας γίνεται βάσει αυτού του αρχικού τμήματος test.

B.2. ΥΛΟΠΟΙΗΣΗ

B.2.1. Εργαλεία

B.2.1.I. Remix IDE



Το Remix IDE [54] είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης smart contracts για Blockchains όπως το Ethereum. Υποστηρίζει τη δημιουργία smart contracts γραμμένα σε γλώσσα προγραμματισμού Solidity και χρησιμοποιείται για την ανάπτυξη, τη μεταγλώττιση αλλά και δοκιμή τους. Στα πλαίσια της διπλωματικής εργασίας χρησιμοποιήθηκε για την ανάπτυξη του smart contract `CaptionReview.sol`. Το σύμβολο

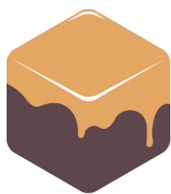
`CaptionReview.sol` είναι αυτό που κάνει εφικτή την επικοινωνία του συστήματος με το Blockchain και ορίζει τις συναρτήσεις για την αποθήκευση και αξιολόγηση δεδομένων και μοντέλων.

B.2.1.II. Truffle



TRUFFLE Το Truffle [55] είναι ένα πλαίσιο λογισμικού (framework) για την ανάπτυξη dApp εφαρμογών στο Ethereum χρησιμοποιώντας το EVM. Περιλαμβάνει χρήσιμες εντολές για τη δημιουργία ενός αρχικού περιγράμματος (template) με τους βασικούς φακέλους που χρειάζεται ο προγραμματιστής, εντολές για τη μεταγλώττιση των smart contracts, εντολές για την παράταξη (deployment) της εφαρμογής αλλά και για την εκτέλεση δοκιμών (testing). Στα πλαίσια της διπλωματικής εργασίας χρησιμοποιήθηκε για την ανάπτυξη του ενός εκ του Blockchain backend του DEMOS dApp το οποίο περιέχεται στο φάκελο `contracts`. Μεταξύ άλλων εκεί περιλαμβάνεται το σύμβολο `CaptionReview.sol`, η μεταγλώττισή του και το αρχείο δοκιμών.

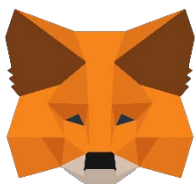
B.2.1.III. Ganache



Ganache

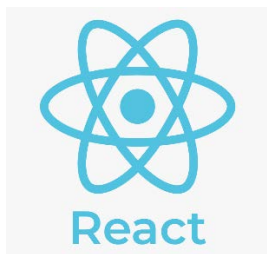
Το Ganache [56] είναι ένα εργαλείο για την δημιουργία τοπικών Ethereum Blockchain δικτύων ώστε να δοκιμαστεί ένα dApp πριν δημοσιευτεί στο δημόσιο δίκτυο του Ethereum. Προσφέρει τη δυνατότητα δημιουργίας εικονικών λογαριασμών Ethereum με τις δικές τους διευθύνσεις και υπόλοιπα λογαριασμών. Παράλληλα υποστηρίζει την παράταξη smart contracts σε κάποια διεύθυνση και οπτικοποιεί τα δεδομένα που περιέχουν. Στα πλαίσια της διπλωματικής εργασίας χρησιμοποιήθηκε για τη δημιουργία ενός τοπικού Ethereum Blockchain για λόγους δοκιμών αλλά και επίδειξης.

B.2.1.IV. Metamask



Το Metamask [57] είναι λογισμικό ηλεκτρονικού πορτοφολιού κρυπτονομισμάτων. Χρησιμοποιείται σε μορφή επέκτασης (extension) για προγράμματα περιήγησης (browsers) όπως το Google Chrome ή σε μορφή εφαρμογής. Προφέρει στους χρήστες την ευκαιρία να αλληλεπιδράσουν με Web 3 εφαρμογές χρησιμοποιώντας το Ethereum πορτοφόλι τους. Στα πλαίσια της διπλωματικής εργασίας χρησιμοποιήθηκε για να ενώνει μέσω του frontend τους εικονικούς λογαριασμούς που δημιουργεί το Ganache με το Blockchain ώστε να εκτελούν τις διάφορες συναλλαγές όπως προσφορά και αξιολόγηση δεδομένων. Σε αυτό φαίνονται διάφορα στοιχεία για τον κάθε λογαριασμό όπως η διεύθυνση και το υπόλοιπό του.

B.2.1.V. React.js



Η React.js [58] είναι μια frontend βιβλιοθήκη ανοικτού κώδικα (open-source library) για τη γλώσσα προγραμματισμού JavaScript. Χρησιμοποιείται για την ανάπτυξη περιβαλλόντων διεπαφής χρήστη (User Interfaces – UI). Στα πλαίσια της διπλωματικής εργασίας χρησιμοποιήθηκε εντός ενός Next.js project για τη δημιουργία του frontend του DEMOS, δηλαδή την ιστοσελίδα που χρησιμοποιεί ο χρήστης για να αλληλεπιδράσει με το σύστημα.

B.2.1.VI. Next.js



Η Next.js [59] είναι ένα React.js πλαίσιο λογισμικού ανοικτού κώδικα (open-source framework) για την ανάπτυξη εφαρμογών. Είναι κατασκευασμένο επί της React.js κάτι που επιτρέπει τη συγγραφή κώδικα που ακολουθεί την κανονική δομή της React.js αλλά προσφέρει και κάποιες επιπλέον διευκολύνσεις στον προγραμματιστή. Στα πλαίσια της διπλωματικής εργασίας χρησιμοποιήθηκε για την ανάπτυξη του frontend του DEMOS dApp. Ο σχετικός κώδικας βρίσκεται στο φάκελο client. Εκτός από το περιβάλλον διεπαφής που χρησιμοποιεί ο χρήστης (δλδ την ιστοσελίδα), χρησιμοποιήθηκε και για να ενώσει την εφαρμογή με το Smart Contract για την αλληλεπίδραση με το Blockchain. Χρησιμοποιήθηκε επίσης για την ένωση με τον Flask διακομιστή (Server) για την αλληλεπίδραση με τα μοντέλα ΝΔ αλλά και με το πορτοφόλι του χρήστη μέσω του Metamask.

B.2.1.VII. Flask



Το Flask [60] είναι πλαίσιο λογισμικού (framework) για τη γλώσσα προγραμματισμού Python. Χρησιμοποιείται για την ανάπτυξη web εφαρμογών. Στα πλαίσια της διπλωματικής εργασίας χρησιμοποιήθηκε για τη δημιουργία του δεύτερου backend του DEMOS dApp, τον Flask διακομιστή (Server). Ο διακομιστής εκτελεί τον κώδικα εκπαίδευσης, αξιολόγησης και αποθήκευσης των μοντέλων ΝΔ. Το συγκεκριμένο λογισμικό επιλέχθηκε

διότι υποστηρίζει τη γλώσσα προγραμματισμού Python η οποία χρησιμοποιήθηκε εξ αρχής για τον κώδικα που σχετίζεται με το ΝΔ. Είναι εξάλλου ίσως η πιο βασική γλώσσα προγραμματισμού για Μηχανική Μάθηση. Για αυτό το λόγο η ενσωμάτωση των συναρτήσεων εκπαίδευσης, αξιολόγησης και αποθήκευσης των μοντέλων στον κώδικα του διακομιστή ήταν αρκετά απλή. Συγκεκριμένα, ο κώδικας αυτών των συναρτήσεων συμπεριλήφθηκε απευθείας στον κώδικα των API endpoints του διακομιστή.

B.2.1.VIII. Google Colab



Το Google Colab [61] είναι ένα περιβάλλον (environment) που υποστηρίζει την ανάπτυξη και εκτέλεση Python notebooks (ipynb). Δεν απαιτεί κάποια εγκατάσταση και τρέχει αποκλειστικά σε διακομιστές της Google χωρίς να επιβαρύνει το μηχάνημα του προγραμματιστή.

Προσφέρει επίσης τη δυνατότητα εκτέλεσης των notebooks σε υπολογιστικούς πόρους όπως GPU. Αυτό προσφέρει σημαντική επιτάχυνση στο χρόνο εκτέλεσης, ειδικά σε προγράμματα Μηχανικής Μάθησης που εκτελούν απαιτητικούς υπολογισμούς. Στα πλαίσια της διπλωματικής εργασίας χρησιμοποιήθηκε αρχικά για τη δημιουργία Python notebooks τα οποία εκπαιδευαν, αξιολογούσαν και αποθήκευαν τα μοντέλα ΝΔ. Στην τελική μορφή του dApp χρησιμοποιείται για να τρέχει τον Flask διακομιστή (server) λόγω των πολύ απαιτητικών πράξεων που έπρεπε να εκτελεί (αφού το μηχάνημα ανάπτυξης του DEMOS dApp δεν ήταν σε θέση να τις εκτελέσει). Παράλληλα χρησιμοποιείται για την προεκπαίδευση των μοντέλων ΝΔ πριν αυτά ενταχθούν στο dApp για μετεκπαίδευση. Επίσης συνδέεται με το Google Drive όπου αποθηκεύονται και ανακτώνται τα μοντέλα ΝΔ.

B.2.1.IX. Kaggle



Το Kaggle [49] είναι μια διαδικτυακή κοινότητα Μηχανικής Μάθησης και ανάλυσης δεδομένων. Υποστηρίζει την ανάπτυξη και εκτέλεση Python notebooks (ipynb), αποτελεί αποθετήριο συνόλων δεδομένων ενώ παράλληλα διοργανώνει και διαγωνισμούς μηχανικής μάθησης. Το Kaggle προσφέρει επίσης υποστήριξη για απευθείας χρήση των συνόλων δεδομένων στο Google Colab. Στα πλαίσια της διπλωματικής εργασίας χρησιμοποιήθηκε για την άντληση συνόλων δεδομένων επιβλεπόμενης μάθησης για την αρχική εκπαίδευση των μοντέλων ΝΔ.

B.2.1.X. TensorFlow



Το TensorFlow [62] είναι μια βιβλιοθήκη λογισμικού ανοικτού κώδικα (open-source library) για Μηχανική Μάθηση. Υποστηρίζει διάφορες γλώσσες προγραμματισμού με ποιο χαρακτηριστικό παράδειγμα την Python. Χρησιμοποιείται για πληθώρα εφαρμογών αλλά εστιάζει κυρίως στα βαθιά Νευρωνικά Δίκτυα. Στα πλαίσια της διπλωματικής εργασίας χρησιμοποιήθηκε για να υποστηρίξει τα μοντέλα ΝΔ της βιβλιοθήκης Transformers. Συγκεκριμένα χρησιμοποιήθηκε στις συναρτήσεις για την εκπαίδευση, τη βελτιστοποίηση και την αξιολόγησή των μοντέλων αλλά και στην προεπεξεργασία των συνόλων δεδομένων εκπαίδευσης.

B.2.1.XI. Transformers



Το Transformers [63] είναι μια βιβλιοθήκη λογισμικού για Μηχανική Μάθηση της ομάδας ανάπτυξης λογισμικού Hugging Face που υποστηρίζει άλλες βιβλιοθήκες όπως το Tensorflow. Παρέχει μεταξύ άλλων έτοιμα προεκπαιδευμένα μοντέλα βαθιών Νευρωνικών Δικτύων ώστε να μη χρειάζεται ο προγραμματιστής να τα σχεδιάσει από την αρχή βαθμίδα προς βαθμίδα. Στα πλαίσια της διπλωματικής εργασίας χρησιμοποιήθηκε το tokenizer[48] και το μοντέλο bert-base-uncased[46]. Το Tokenizer είναι υπεύθυνο για την προετοιμασία και προεπεξεργασία των δεδομένων εισόδου του ΝΔ πριν χρησιμοποιηθούν για την εκπαίδευση. Το μοντέλο bert-base-uncased είναι ένα έτοιμο μοντέλο ΝΔ το οποίο βασίζεται στο μοντέλο αναπαράστασης φυσικής γλώσσας BERT[41] και είναι προεκπαιδευμένο για την Αγγλική γλώσσα.

B.2.1.XII. Ngrok



Το Ngrok [64] είναι μια cross-platform εφαρμογή που δημοσιοποιεί κάποια τοπική θύρα υπολογιστικού μηχανήματος στο διαδίκτυο και είναι ένας γρήγορος τρόπος να φιλοξενηθεί στο διαδίκτυο μία εφαρμογή. Στα πλαίσια της διπλωματικής εργασίας χρησιμοποιήθηκε για τη δημοσιοποίηση του Flask διακομιστή (server) στο διαδίκτυο ώστε να είναι ορατός από το front-end. Πιο αναλυτικά, ο Flask διακομιστής τρέχει τοπικά στη θύρα 5000 σε κάποιο μηχάνημα της Google. Αυτή η θύρα δεν είναι προσβάσιμη από άλλα μηχανήματα. Για τον λόγο αυτό χρησιμοποιείται το ngrok για να κάνει προσβάσιμη τη θύρα 5000 του μηχανήματος της Google στο διαδίκτυο. Έτσι, το frontend που βρίσκεται σε άλλο μηχάνημα μπορεί να αποστείλει τα απαραίτητα API requests για την επικοινωνία με τον Flask διακομιστή (server) και να αλληλεπιδράσει με τα μοντέλα ΝΔ.

B.2.1.XIII. GitLab



Το GitLab [65] είναι ένα λογισμικό ανοικτού κώδικα (open-source software) για την αποθήκευση και το διαμοιρασμό λογισμικού. Υποστηρίζει τη διαχείριση αποθετηρίων git (git repositories) τα οποία είναι ένα σύστημα ελέγχου εκδόσεων (version control) λογισμικού. Στα πλαίσια της διπλωματικής εργασίας χρησιμοποιήθηκε για την αποθήκευση της ίδιας της εργασίας σε διαδοχικές εκδόσεις. Ο τελικός κώδικας μαζί με όλα τα σχετικά έγγραφα και εγχειρίδια βρίσκονται αποθηκευμένα εκεί ώστε να έχει ο καθένας πρόσβαση σε αυτά.

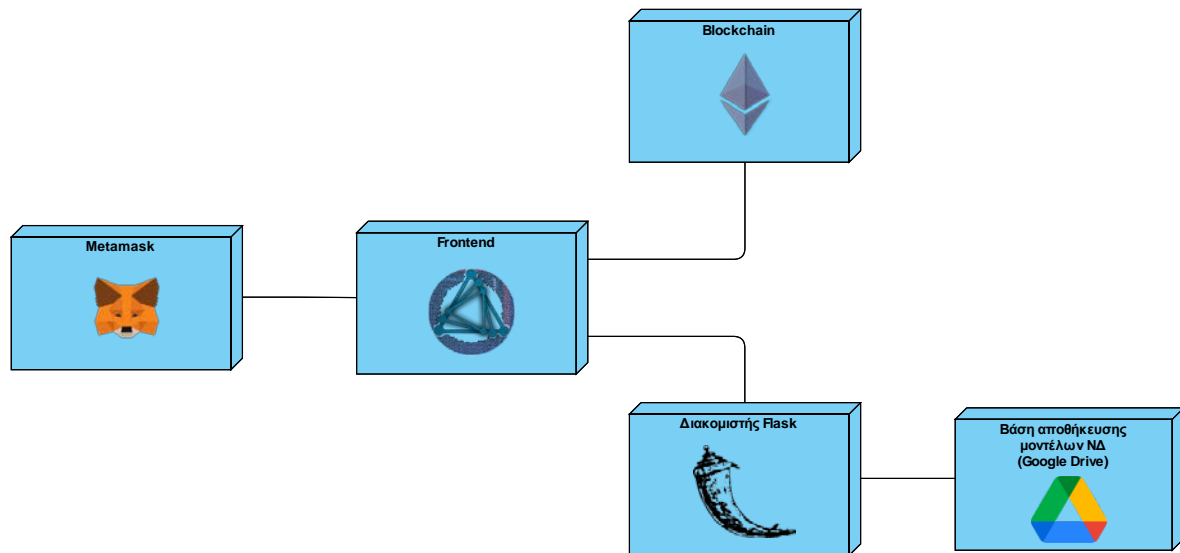
B.2.1.XIV. Visual Paradigm



Το Visual Paradigm [66] είναι ένα εργαλείο για τη δημιουργία διαγραμμάτων. Υποστηρίζει τη μοντελοποίηση με διάφορους τύπους διαγραμμάτων όπως τα UML διαγράμματα. Στα πλαίσια της διπλωματικής εργασίας χρησιμοποιήθηκε για τη δημιουργία διαγραμμάτων όπως αυτά που υπάρχουν στο παρόν έγγραφο.

B.2.2. Δομή Υλοποίησης

Το DEMOS dApp είναι δομημένο με δύο επί μέρους ανεξάρτητα backend τμήματα και ένα τμήμα frontend όπως φαίνεται στο Διάγραμμα 2. Τα δύο backend τμήματα αποτελούν τη ραχοκοκαλιά του dApp και είναι υπεύθυνα για όλες τις λειτουργίες του. Το frontend είναι ο συνδετικός κρίκος μεταξύ τους αλλά και με το χρήστη.



Διάγραμμα 2: Δομή Υλοποίησης

B.2.2.I.Backend 1 – Blockchain

Το 1^ο backend περιέχεται στο φάκελο contracts και δημιουργήθηκε κυρίως με τη χρήση του πλαισίου λογισμικού Truffle. Περιλαμβάνει το αρχείο truffle-config.js που περιέχει πληροφορίες για τις ρυθμίσεις του συγκεκριμένου truffle project. Περιλαμβάνει επίσης τα smart contracts του dApp γραμμένα σε Solidity, τη μεταγλώττισή τους σε bytecode κατανοητό από το EVM και τα ABIs που περιέχουν πληροφορίες για τη δομή των συμβολαίων και τις συναρτήσεις τους. Τα συμβόλαια που υπάρχουν είναι το Migrations.sol που περιέχει πληροφορίες για την παράταξη (migration) των συμβολαίων και το CaptionReview.sol που υποστηρίζει την αλληλεπίδραση του συστήματος με το Blockchain. Πέρα από αυτά, ο φάκελος contracts περιέχει και ένα αρχείο για τη δοκιμή των συναρτήσεων του CaptionReview.sol.

Το συμβόλαιο CaptionReview.sol χρησιμοποιείται για την αποθήκευση των πληροφοριών σχετικά με τα μοντέλα ΝΔ (βλ. Εικόνα 2) αλλά και για την αποθήκευση των δεδομένων εκπαίδευσης (βλ. Εικόνα 3). Περιλαμβάνει επίσης συναρτήσεις με τις οποίες ανακτώνται δεδομένα από το Blockchain αλλά και συναρτήσεις με τις οποίες οι κόμβοι αλληλεπιδρούν με το Blockchain. Τέτοιες συναρτήσεις είναι η προσθήκη νέων μοντέλων και δεδομένων, η αξιολόγηση δεδομένων (βλ. Εικόνα 4), η ανάκτηση πληροφοριών από το Blockchain και η ανταμοιβή των κόμβων (βλ. Εικόνα 5). Είναι επίσης υπεύθυνο να αποτρέπει παράνομες ενέργειες όπως η καταχώρηση δύο ψήφων από τον ίδιο κόμβο για το ίδιο δεδομένο ή η δωρεά δεδομένων από κάποιο κόμβο του οποίου το υπόλοιπο του λογαριασμού δεν

καλύπτει την προκαταβολή. Τέλος, σε αυτό ορίζονται οι τιμές για τα κόστη και τις ανταμοιβές των κόμβων.

```
struct ModelInfo{
    uint256 id;
    string name;
    string description;
    uint256 NumberOfVotes;
    uint256 NumberOfCaptions;
    string[] labels;
    address modelProviderAddr;
    uint256[] accuracy;
}
```

Εικόνα 2: Δομή αποθήκευσης πληροφοριών μοντέλων ΝΔ

```
struct Caption{
    string content;
    uint256 modelId;
    uint8 proposedLabel;
    uint8 verifiedLabel;
    uint8[] Votes;
    bool verified;
    address providerAddr;
    address[] voters;
    bool trained;
}
```

Εικόνα 3: Δομή αποθήκευσης δεδομένων εκπαίδευσης

```
function reviewCaption(uint _idx, uint8 _lbl, uint _modelId) external payable{
    require(msg.value >= reviewCost, "Not enough eth");
    require(captions[_idx].verified == false, "Cannot review a verified caption");
    require(msg.sender != captions[_idx].providerAddr, "Cannot review your own caption");
    require(_idx < captionCnt && _idx >= 0, "Invalid caption idx");
    require(_lbl >= 0 && _lbl < models[_modelId].labels.length, "Invalid label");
    require(notVoted(msg.sender, _idx), "Only one vote per account for every caption");
    captions[_idx].Votes[_lbl] += 1;
    captions[_idx].voters.push(msg.sender);
    if(captions[_idx].Votes[_lbl] == models[_modelId].NumberOfVotes){
        captions[_idx].verified = true;
        captions[_idx].verifiedLabel = _lbl;
        refunding(_idx);
    }
}
```

Εικόνα 4: Συνάρτηση Αξιολόγησης Δεδομένων

```
function refunding(uint _captionId) private {
    payable(captions[_captionId].providerAddr).transfer(captionRefund);
    for (uint i = 0; i < captions[_captionId].voters.length; i++){
        payable(captions[_captionId].voters[i]).transfer(reviewRefund);
    }
}
```

Εικόνα 5: Συνάρτηση απόδοσης ανταμοιβής

Από το φάκελο contracts γίνεται επίσης η δημοσίευση των συμβολαίων στο Blockchain ώστε να αποκτήσουν διεύθυνση και να μπορούν εκτελεστούν από τους κόμβους. Στην περίπτωση του DEMOS dApp η δημοσίευση γίνεται στο τοπικό Blockchain που δημιουργεί το Ganache.

Το Ganache προσφέρει εύκολη δημιουργία ενός τοπικού Blockchain και η αποθήκευση των δεδομένων σε blocks για τη δημιουργία του Blockchain γίνεται αυτόματα. Για την ανάπτυξη dApps όπως το DEMOS αυτό διευκολύνει σημαντικά τον προγραμματιστή διότι δεν χρειάζεται να δημιουργήσει το Blockchain από το 0 και να θέσει κόμβους-μεταλλωρύχους. Το Ganache παρουσιάζει επίσης σε γραφικό περιβάλλον απαραίτητες πληροφορίες για το blockchain αλλά και τα δεδομένα που είναι αποθηκευμένα σε αυτό.

Αρχικά το Ganache ρυθμίζεται χρησιμοποιώντας το αρχείο truffle-config.js και θέτονται κάποιες ρυθμίσεις για τους λογαριασμούς που θα δημιουργηθούν όπως αρχικό υπόλοιπο κτλ. Μετά τη δημιουργία του Blockchain και την παράταξη (migration) των συμβολαίων παρουσιάζονται οι επιλογές του Ganache οι οποίες συνοψίζονται ως εξής:

- Στην καρτέλα Accounts (βλ. Εικόνα 6) υπάρχει μια σύνοψη για τους λογαριασμούς, τα υπόλοιπα, τις διευθύνσεις τους κ.α.
- Στην καρτέλα Blocks (βλ. Εικόνα 7) φαίνονται τα blocks που έχουν δημιουργηθεί μέχρι εκείνη τη στιγμή και αποτελούν το Blockchain. Στην περίπτωση του DEMOS dApp δεν προσφέρουν κάποια ουσιαστική πληροφορία αλλά δίνουν μια εποπτεία και μια αντίληψη του τρόπου αποθήκευσης των συναλλαγών σε blocks.
- Στην καρτέλα Transactions (βλ. Εικόνα 8) φαίνονται οι συναλλαγές που έχουν καταχωρηθεί στο Blockchain. Όπως και η καρτέλα Blocks δεν προσφέρει ουσιαστική πληροφορία για το DEMOS αλλά από εκεί μπορεί να επιβεβαιωθεί ότι μια συναλλαγή έχει καταχωρηθεί στο Blockchain. Για παράδειγμα (βλ. Εικόνα 9), φαίνεται μία συναλλαγή από ένα λογαριασμό (συγκεκριμένα τον 5^ο) προς το συμβόλαιο CaptionReview για την αξιολόγηση ενός δεδομένου καλώντας τη συνάρτηση reviewCaption() με παραμέτρους: id δεδομένου = 12, ετικέτα = 1 (Negative) , id μοντέλου = 0 (Twitter).
- Στην καρτέλα Contracts (βλ. Εικόνα 10) φαίνονται τα δύο συμβόλαια του Blockchain, οι διευθύνσεις τους, η κατάστασή τους (deployed αφού έχουν παραταχθεί) κ.α. Για κάθε συμβόλαιο δίνονται αναλυτικές πληροφορίες για το υπόλοιπό του και τα δεδομένα που περιέχει (βλ. Εικόνα 11). Αυτές οι πληροφορίες είναι πολύ βοηθητικές για τόσο για το debugging όσο και για τη γενική εποπτεία των δεδομένων του Blockchain. Πιο κάτω φαίνονται και πληροφορίες για το μοντέλο Tweet Categorization (βλ. Εικόνα 12) αλλά και για ένα δεδομένο εκπαίδευσης (βλ. Εικόνα 13).

ACCOUNTS

BLOCKS

TRANSACTIONS

CONTRACTS

EVENTS

LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK
345

GAS PRICE
20000000000

GAS LIMIT
6721975

HARDFORK
MUIRGLACIER

NETWORK ID
5777

RPC SERVER
HTTP://127.0.0.1:7545

MINING STATUS
AUTOMINING

WORKSPACE
THESIS-GENERALPEERREVIEWV1

SWITCH

MNEMONIC

better already skate silver enact draw range over fruit skill dog coil

HD PATH

m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX	
0xc0d9B259df90a7c41d7647B01C32600C0F949504	982.60 ETH	151	0	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x6BBA0E63D4a2e5Cdf43cC9387d0563E7D3b0D836	987.90 ETH	60	1	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x4D7B51E9151aa0F2c21fb81790AA68d82Fa6C38F	1000.19 ETH	46	2	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x0754a2391BF7cfd653F79C5D03471A2Fc0a57237	1000.20 ETH	47	3	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x1A4445906F1756a417cE53710d7a4Ca3426dE1b5	1000.13 ETH	36	4	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x29Eb978dd4A28206C54cF455f17a0Efed86A2011	1000.01 ETH	3	5	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x790C85abdcAFA75f4B19fB49f33e6394c8667792	1000.01 ETH	2	6	
ADDRESS	BALANCE	TX COUNT	INDEX	
0xFF6d98b557b9392382aAe3ae333ab0E70182C5b3	1000.00 ETH	0	7	

Εικόνα 6: Ganache - Λίστα λογαριασμών

Ganache

ACCOUNTS

BLOCKS

TRANSACTIONS

CONTRACTS

EVENTS

LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK
345

GAS PRICE
20000000000

GAS LIMIT
6721975

HARDFORK
MUIRGLACIER

NETWORK ID
5777

RPC SERVER
HTTP://127.0.0.1:7545

MINING STATUS
AUTOMINING

WORKSPACE
THESIS-GENERALPEERREVIEWV1

SWITCH

BLOCK
345

MINED ON
2022-06-13 12:25:39

GAS USED
67043

1 TRANSACTION

BLOCK
344

MINED ON
2022-06-13 12:25:16

GAS USED
48822

1 TRANSACTION

BLOCK
343

MINED ON
2022-06-13 12:18:38

GAS USED
140934

1 TRANSACTION

BLOCK
342

MINED ON
2022-06-13 12:18:33

GAS USED
136734

1 TRANSACTION

BLOCK
341

MINED ON
2022-06-13 12:18:07

GAS USED
75765

1 TRANSACTION

BLOCK
340

MINED ON
2022-06-13 12:18:02

GAS USED
75765

1 TRANSACTION

BLOCK
339

MINED ON
2022-06-13 12:17:28

GAS USED
101230

1 TRANSACTION

BLOCK
338

MINED ON
2022-06-13 12:17:15

GAS USED
101230

1 TRANSACTION

BLOCK
337

MINED ON
2022-06-13 12:16:31

GAS USED
187930

1 TRANSACTION

BLOCK
336

MINED ON
2022-06-13 12:12:32

GAS USED
249805

1 TRANSACTION

BLOCK
335

MINED ON
2022-06-13 11:42:59

GAS USED
67043

1 TRANSACTION

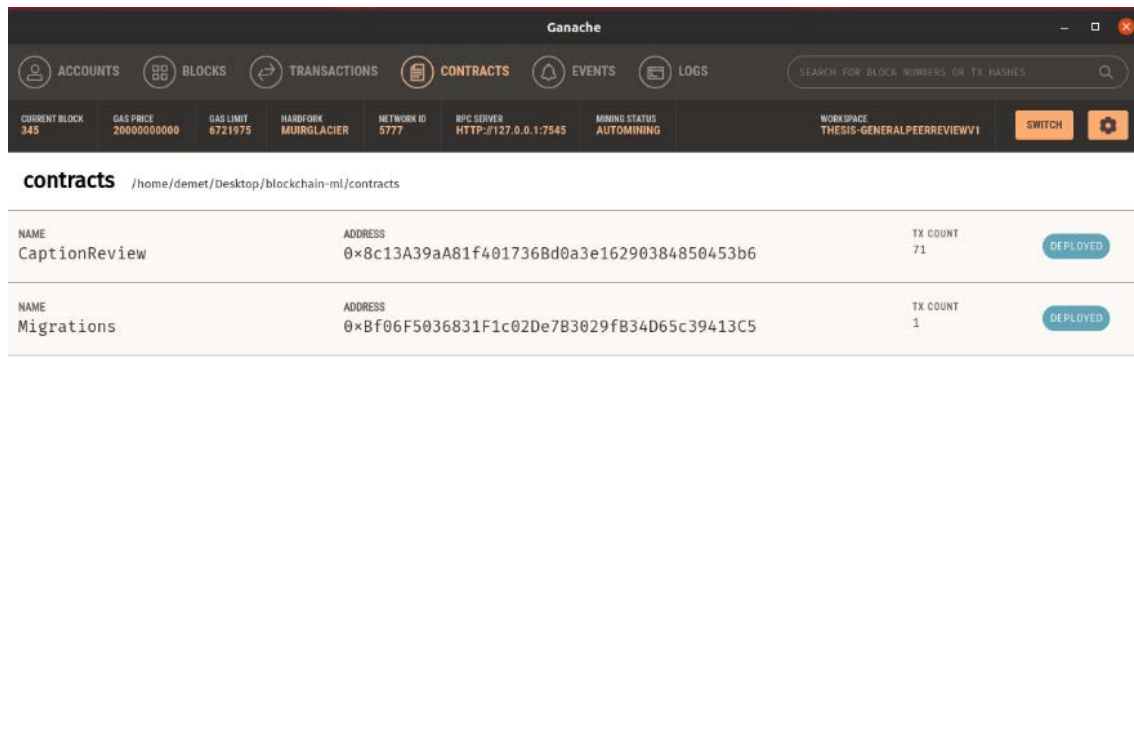
BLOCK
334

MINED ON
2022-06-13 11:42:54

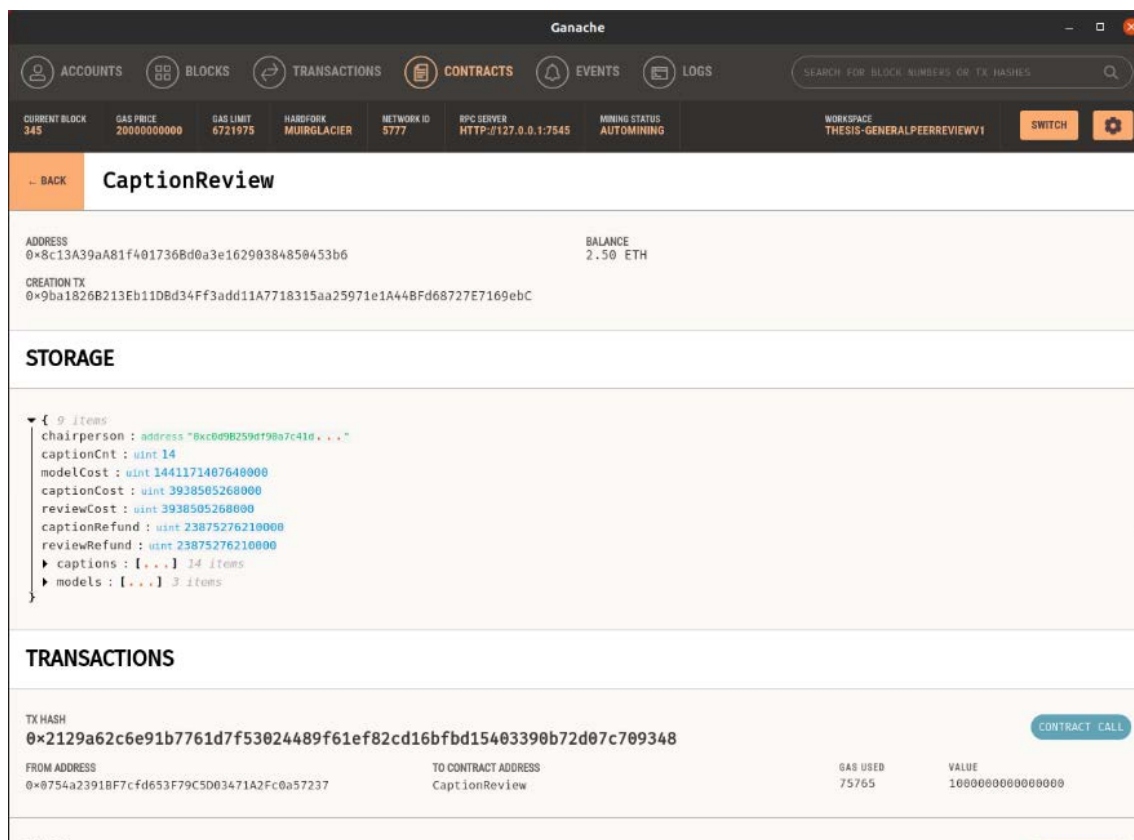
GAS USED
48834

1 TRANSACTION

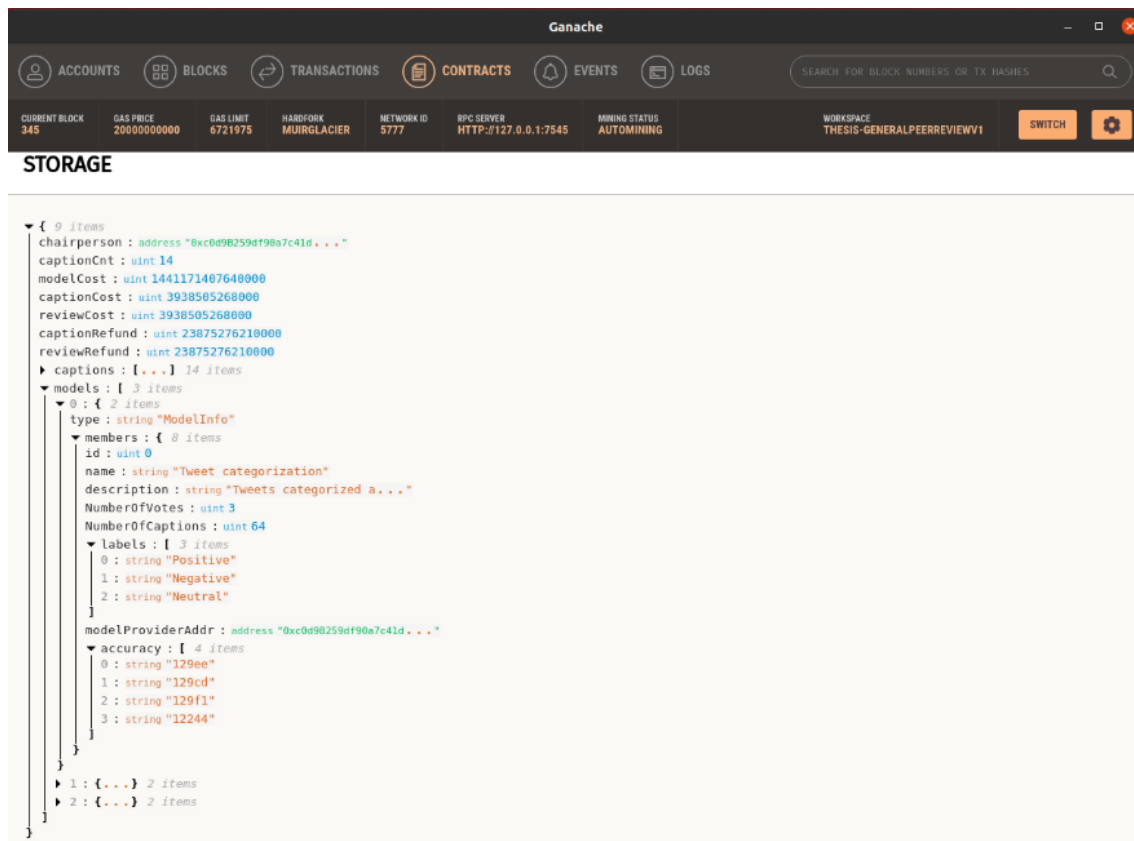
Εικόνα 7: Ganache - Λίστα δημιουργημένων Blocks



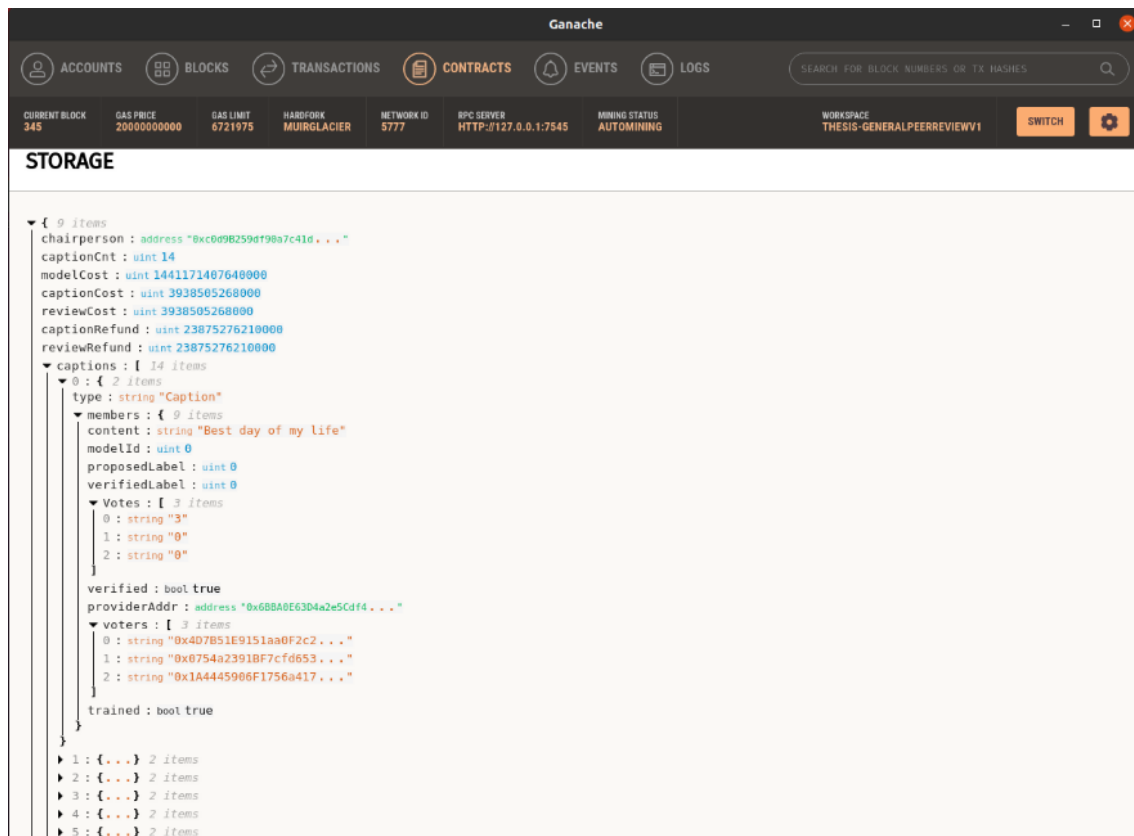
Εικόνα 10: Ganache - Λίστα Smart Contracts



Εικόνα 11: Ganache - Ανάλυση δεδομένων συμβολαίου CaptionReview



Εικόνα 12: Ganache - Ανάλυση αποθηκευμένων πληροφοριών μοντέλου ΝΔ








Εικόνα 13: Ganache – Ανάλυση πληροφοριών αποθηκευμένου δεδομένου εκπαίδευσης

B.2.2.II. Backend 2 – Διακομιστής (server) εκπαίδευσης ΝΔ

Το 2^ο backend δημιουργήθηκε χρησιμοποιώντας κυρίως το πλαίσιο λογισμικού Flask. Ο πλήρης κώδικας για την εκτέλεση του διακομιστή σε τοπικό μηχάνημα βρίσκεται στο φάκελο backend. Ωστόσο, στην πράξη ο διακομιστής τρέχει στο Google Colab και ο σχετικός κώδικας είναι το notebook αρχείο FlaskServerForBert.ipynb που βρίσκεται στο φάκελο GoogleColab-backend.

Στο φάκελο GoogleColab-backend υπάρχουν επίσης τα αρχεία notebook για την αρχική εκπαίδευση των 3 μοντέλων ΝΔ που χρησιμοποιούνται στο DEMOS dApp. Τα 3 αρχεία είναι πανομοιότυπα και διαφέρουν μόνο στο σημείο φορτώματος συνόλου δεδομένων και στις παραμέτρους βελτιστοποίησης. Σε περίπτωση που ένας χρήστης επιθυμεί να προσθέσει ένα νέο μοντέλο ΝΔ στο DEMOS θα χρειαστεί ένα αντίστοιχο αρχείο notebook που να φορτώνει το νέο σύνολο δεδομένων.

Το αρχείο FlaskServerForBert.ipynb ενώνεται με το Google Drive ώστε να ανακτά και να αποθηκεύει τα μοντέλα και τα τμήματα δεδομένων validation και testing, βλ. Εικόνα 14. Στην περίπτωση της τοπικής εκτέλεσης η αποθήκευση γίνεται τοπικά στο φάκελο NeuralNetwork το περιεχόμενο του οποίου όμως παρέμεινε κενό για λόγους εξοικονόμησης χώρου.

	BERTModel-Twitter	εγώ	7 Ιουν 2022
	BERTModel-IMDB	εγώ	7 Ιουν 2022
	BERTModel-Emotions	εγώ	8 Ιουν 2022
	Backups	εγώ	8 Ιουν 2022
	BERTModel-Twitter-validation.csv	εγώ	7 Ιουν 2022
	BERTModel-Twitter-test.csv	εγώ	7 Ιουν 2022
	BERTModel-IMDB-validation.csv	εγώ	7 Ιουν 2022
	BERTModel-IMDB-test.csv	εγώ	7 Ιουν 2022
	BERTModel-Emotions-validation.csv	εγώ	8 Ιουν 2022
	BERTModel-Emotions-test.csv	εγώ	8 Ιουν 2022

Εικόνα 14: Λίστα αποθηκευμένων αρχείων στο Google Drive

Ο διακομιστής παρέχει 3 API endpoints, /train, /evaluate και /test τα οποία αναλύονται ως εξής:

- /train endpoint: Δέχεται αιτήματα τύπου POST από το frontend τα οποία κουβαλούν μία παράμετρο model ID που παραπέμπει στο κατάλληλο μοντέλο ΝΔ και δεδομένα για να το εκπαιδεύσουν. Αφότου διατηρηθούν μόνο οι κατάλληλες πληροφορίες για την εκπαίδευση και αποθηκευτούν σε μορφή Pandas Dataframe [67] καλείται η συνάρτηση trainModel() που αρχικά περνά τα δεδομένα από τον tokenizer. Στη συνέχεια επανεκπαιδεύει το μοντέλο χρησιμοποιώντας το αντίστοιχο τμήμα validation και το αποθηκεύει στο Google Drive. Στο τέλος στέλνει μήνυμα πίσω στο frontend ότι η εκπαίδευση ολοκληρώθηκε με επιτυχία.

```
@app.route('/train', methods=['POST'])
def trainFun():
    print('Starting Training')

    modelId = int(request.args.get('model'))

    trainSetInput = json.loads(request.data.decode('utf-8'))

    trainDF = pd.DataFrame(trainSetInput)
    del trainDF['id']
    del trainDF['proposedLbl']
    del trainDF['goodData']
    trainDF.columns = ['DATA_COLUMN', 'LABEL_COLUMN']
    trainModel(loaderModel[modelId], tokenizer[modelId], validation[modelId], trainDF, modelNames[modelId])

    response = jsonify(status="ok")
    return response
```

- /evaluate endpoint: Δέχεται αιτήματα τύπου GET από το frontend τα οποία κουβαλούν μία παράμετρο model ID που παραπέμπει στο κατάλληλο μοντέλο ΝΔ. Η δουλειά της είναι να ανακτήσει το αποθηκευμένο μοντέλο ΝΔ με το αντίστοιχο ID και να εκτελεί τη συνάρτηση evaluateModel(). Η evaluateModel() μετρά την επίδοση του μοντέλου για τις μετρικές Accuracy και Loss χρησιμοποιώντας το αποθηκευμένο τμήμα δεδομένων testing. Στο τέλος επιστρέφει στο frontend τις τιμές για τις μετρικές Accuracy και Loss. Το αίτημα evaluate γίνεται από το frontend αμέσως μόλις λάβει ειδοποίηση ότι ολοκληρώθηκε με επιτυχία η εκπαίδευση.

```
@app.route('/evaluate')
def evaluateFun():
    print('Starting Evaluation')

    modelId = int(request.args.get('model'))

    eval = evaluateModel(loaderModel[modelId], tokenizer[modelId], test[modelId])
    [loss, acc] = eval

    response = jsonify(loss=loss, acc=acc)
    return response
```

- /test endpoint: Δέχεται αιτήματα τύπου POST από το frontend τα οποία κουβαλούν μία παράμετρο model ID που παραπέμπει στο κατάλληλο μοντέλο NΔ και ένα δεδομένο χωρίς ετικέτα. Έπειτα καλείται η συνάρτηση testModel που ανακτά το κατάλληλο μοντέλο και προβλέπει μία ετικέτα για το δεδομένο. Στο τέλος η πρόβλεψη επιστρέφεται στο frontend.

```
@app.route('/test', methods=['POST'])
def testFun():

    modelId = int(request.args.get('model'))

    dataIn = json.loads(request.data.decode('utf-8'))
    pred_sent = dataIn['caption']

    pred = testModel(loaderModel[modelId],tokenizer[modelId],pred_sent,labels[modelId])

    response = jsonify(prediction=pred)
    return response
```

Όταν ο διακομιστής εκκινήσει στο Google Colab τρέχει στη θύρα 5000 του τοπικού μηχανήματος της Google. Αυτή η θύρα δεν είναι προσβάσιμη από το frontend που εκτελείται από διαφορετικό μηχάνημα. Για αυτό το λόγο χρησιμοποιείται το Ngrok που εκθέτει τη θύρα σε ένα URL στο διαδίκτυο όπως φαίνεται πιο κάτω (βλ. Εικόνα 15).

Server Running 

```
* Serving Flask app "__main__" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Running on http://679d-104-199-161-238.ngrok.io
* Traffic stats available on http://127.0.0.1:4040
```

Εικόνα 15: Παράδειγμα στιγμιότυπου εκτέλεσης διακομιστή στο Google Colab

B.2.2.III. Frontend

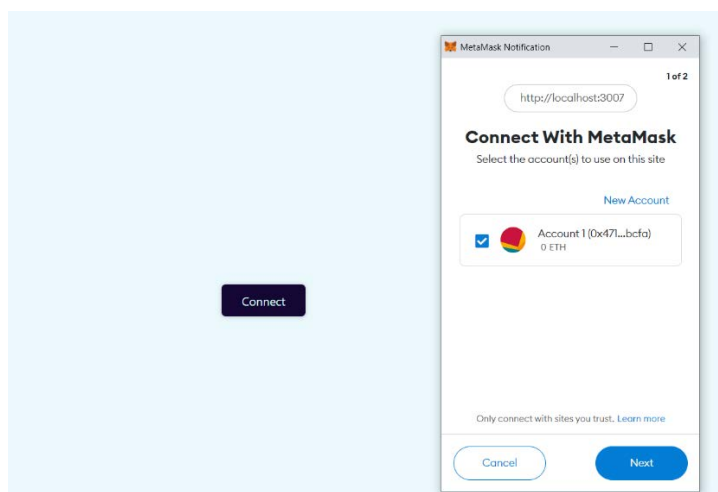
Το frontend δημιουργήθηκε χρησιμοποιώντας κυρίως το πλαίσιο λογισμικού Next.js. Είναι το μέσο που χρησιμοποιεί ο κόμβος-χρήστης για να αλληλεπιδράσει με το DEMOS.

Ουσιαστικά είναι μια ιστοσελίδα στην οποία ενώνεται ο κόμβος και μπορεί να εκτελέσει διάφορες λειτουργίες όπως αίτηση συλλογής δεδομένων για κάποιο μοντέλο ΝΔ, παροχή δεδομένων και εκπαίδευση μοντέλων ΝΔ. Το DEMOS διαθέτει 6 σελίδες πέρα από την αρχική. Ακολουθεί μια ανάλυση του frontend και των λειτουργιών του DEMOS dApp.

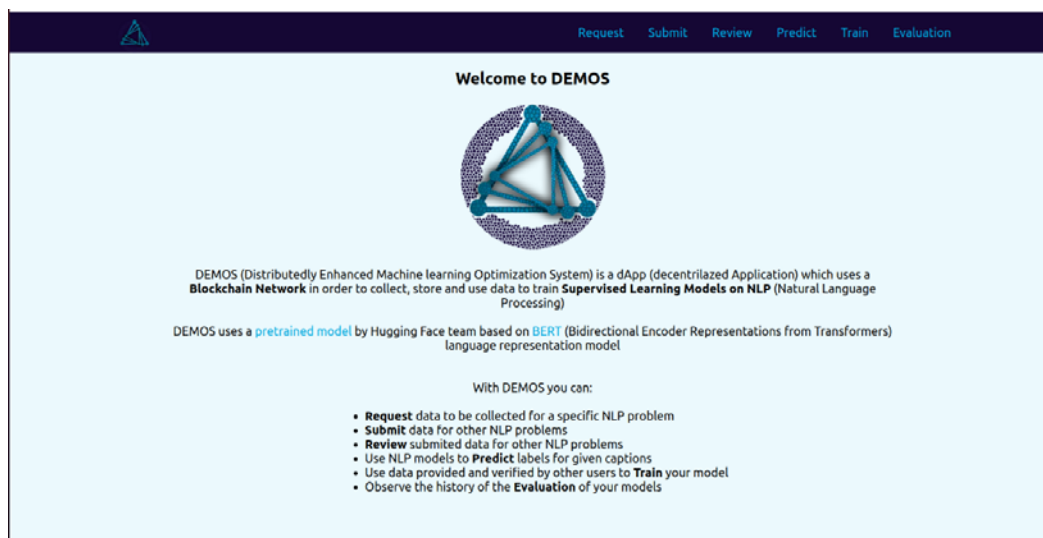
Σημείωση: Για περισσότερες πληροφορίες, στον αναλυτικό οδηγό εγκατάστασης (αρχείο README.md) που βρίσκεται στη σελίδα του DEMOS στο Gitlab [68] υπάρχει ένα κεφάλαιο, συγκεκριμένα το κεφάλαιο 7, που αναλύει με οδηγίες και παραδείγματα τον τρόπο χρήσης του.

B.2.2.III.a. Αρχική Σελίδα

Αρχικά ο χρήστης αρχικά συνδέεται στην εφαρμογή χρησιμοποιώντας ένα πορτοφόλι Metamask (βλ. Εικόνα 16). Μετά τη σύνδεση ο χρήστης οδηγείται στην αρχική σελίδα του DEMOS η οποία περιέχει κάποιες γενικές πληροφορίες για την εφαρμογή (βλ. Εικόνα 17)



Εικόνα 16: DEMOS - Σύνδεση στην εφαρμογή



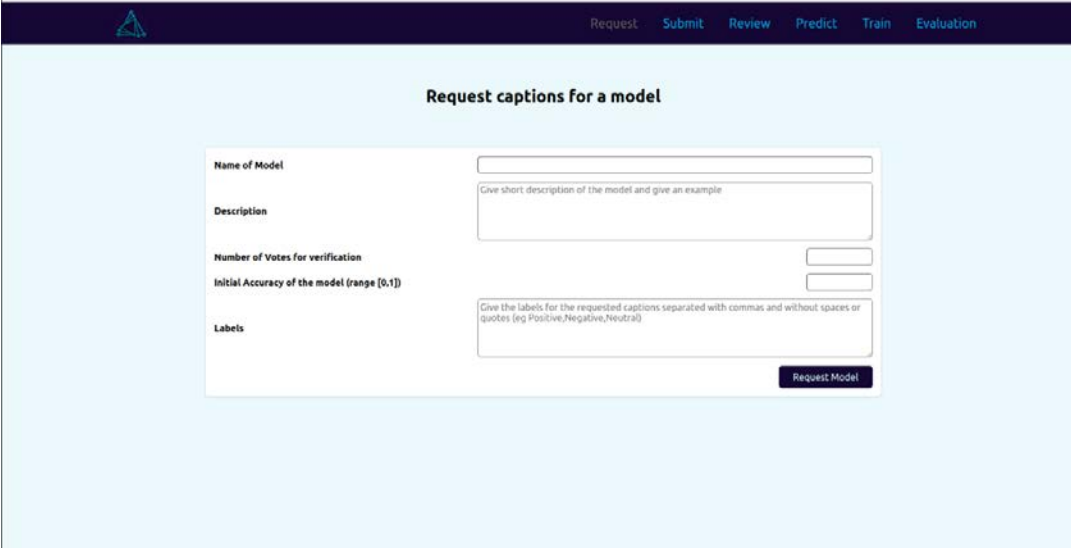
Εικόνα 17: DEMOS - Αρχική Σελίδα

B.2.2.III.β. Σελίδα Request

Από αυτή τη σελίδα (βλ. Εικόνα 18) ένας κόμβος-αιτητής μπορεί να αιτηθεί τη συλλογή δεδομένων επιβλεπόμενης μάθησης για τη μετεκπαίδευση ενός μοντέλου ΝΔ.

Ο κόμβος-αιτητής συμπληρώνει τη φόρμα και εξηγεί τον τύπο των δεδομένων που επιθυμεί να συλλεγούν. Δίνει επίσης τον αριθμό ψήφων M που χρειάζεται ένα δεδομένο για να θεωρηθεί επικυρωμένο (βλ. παράγραφο B.1.2 Μηχανισμός Κινήτρου (Incentive Mechanism), σελ.37), την αρχική ακρίβεια του μοντέλου και τις πιθανές ετικέτες των δεδομένων. Αυτές οι πληροφορίες καταχωρούνται στο Blockchain ως μία ModelInfo μορφή δεδομένων (βλ. Εικόνα 2) στον πίνακα των αποθηκευμένων μοντέλων models (βλ. Εικόνα 12). Πιο κάτω υπάρχει και ένα παράδειγμα (βλ. Εικόνα 19)

Σημείωση: Πρέπει επίσης να περαστεί το αντίστοιχο προεκπαιδευμένο μοντέλο στο backend. Λόγω μεγάλου μεγέθους των αρχείων και εξάρτησης από το Google Colab και Google Drive αυτό γίνεται χειροκίνητα αλλά σε μία υλοποίηση σε κανονικό διακομιστή το ανέβασμα του προεκπαιδευμένου μοντέλου θα γινόταν από αυτή τη σελίδα.

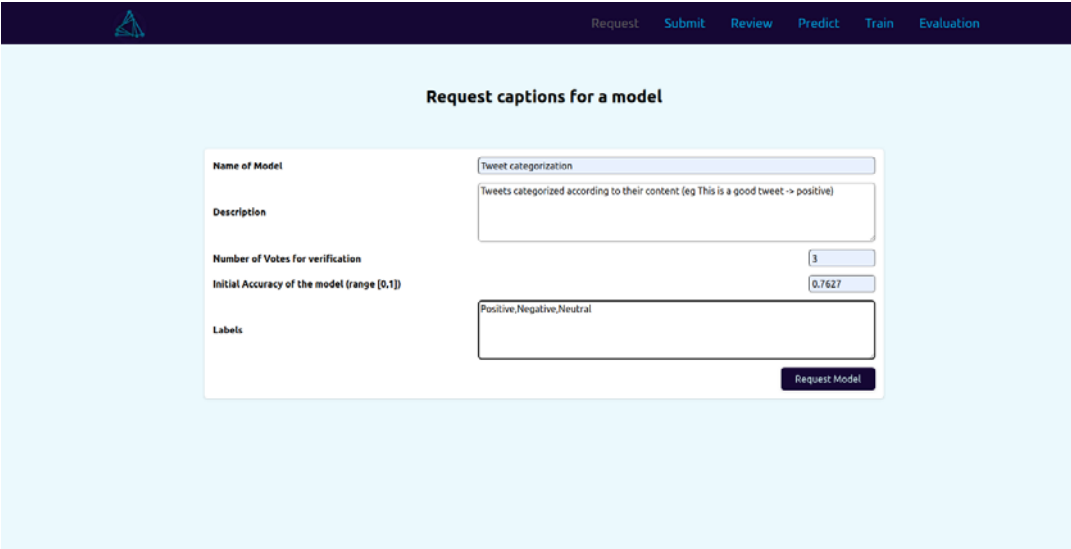


The screenshot shows a web interface with a dark blue header containing a logo and navigation links: Request, Submit, Review, Predict, Train, Evaluation. The main content area is light blue and titled "Request captions for a model". It contains a form with the following fields:

- Name of Model**: A text input field.
- Description**: A text input field with placeholder text "Give short description of the model and give an example".
- Number of Votes for verification**: A numeric input field.
- Initial Accuracy of the model (range [0,1])**: A numeric input field.
- Labels**: A text input field with placeholder text "Give the labels for the requested captions separated with commas and without spaces or quotes (eg Positive,Negative,Neutral)".

A "Request Model" button is located at the bottom right of the form.

Εικόνα 18: DEMOS - Σελίδα Request



This screenshot shows the same form as Figure 18, but with example data filled in:

- Name of Model**: Tweet categorization
- Description**: Tweets categorized according to their content (eg This is a good tweet -> positive)
- Number of Votes for verification**: 3
- Initial Accuracy of the model (range [0,1])**: 0.7627
- Labels**: Positive,Negative,Neutral

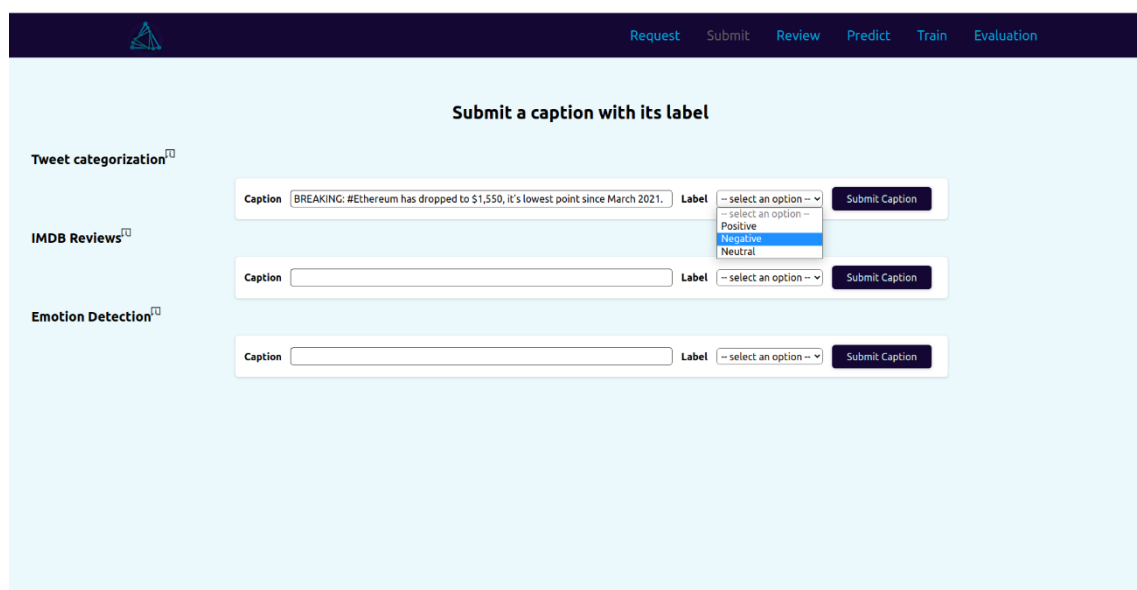
The "Request Model" button is still present at the bottom right.

Εικόνα 19: DEMOS – Σελίδα Request (παράδειγμα συμπλήρωσης φόρμας)

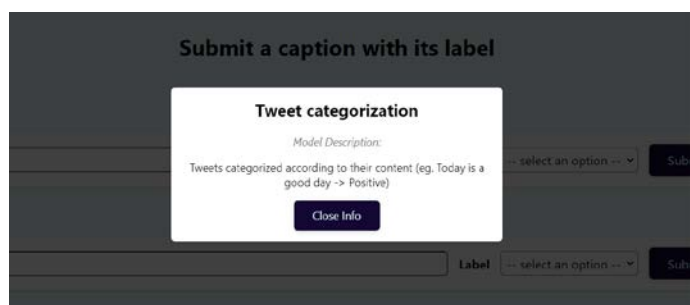
B.2.2.III.γ. Σελίδα Submit

Από αυτή τη σελίδα (βλ. Εικόνα 20) ένας κόμβος-δωρητής μπορεί να προσφέρει δεδομένα για κάποιο μοντέλο που έχει αιτηθεί κάποιος κόμβος-αιτητής.

Ο κόμβος-δωρητής βλέπει τις πληροφορίες των διαθέσιμων μοντέλων (βλ. Εικόνα 21) και παρέχει δεδομένα εκπαίδευσης μαζί με την ετικέτα που θεωρεί ότι ταιριάζει καλύτερα στο καθένα. Όπως ορίζει ο μηχανισμός κινήτρου, ένας κόμβος δωρητής μπορεί να προσφέρει όσα δεδομένα επιθυμεί και για όποια μοντέλα επιθυμεί. Υπάρχει μεν μια προκαταβολή για κάθε δωρεά αλλά εφόσον τα δεδομένα επικυρωθούν και είναι καλά τότε ο κόμβος θα ανταμειφθεί. Οι πληροφορίες για κάθε δεδομένο αποθηκεύονται στο Blockchain ως μία Caption δομή δεδομένων (βλ. Εικόνα 3) στον πίνακα αποθηκευμένων δεδομένων captions (βλ. Εικόνα 13).



Εικόνα 20: DEMOS - Σελίδα Submit



Εικόνα 21: DEMOS - Παράδειγμα ανάλυσης πληροφοριών μοντέλου NA

B.2.2.III.δ. Σελίδα Review

Από αυτή τη σελίδα (βλ. Εικόνα 22) ένας κόμβος-αξιολογητής μπορεί να αξιολογήσει τα δεδομένα που προσέφεραν οι κόμβοι-δωρητές.

Οι κόμβοι-αξιολογητές βλέπουν τα δεδομένα που έχουν καταχωρήσει οι κόμβοι-δωρητές αλλά όχι την ετικέτα που έχουν προτείνει. Μπορούν επίσης να δουν τις πληροφορίες για το κάθε μοντέλο και να επιλέξουν σύμφωνα με την κρίση τους ποια ετικέτα θεωρούν πιο

κατάλληλη για το δεδομένο. Όπως ορίζει ο μηχανισμός κινήτρου, ένας κόμβος μπορεί να αξιολογήσει όσα δεδομένα επιθυμεί, μία φορά το καθένα, αρκεί να μην είναι δεδομένα που έχει προσφέρει ο ίδιος ως κόμβος-δωρητής. Υπάρχει και σε αυτή την περίπτωση μια προκαταβολή για κάθε αξιολόγηση αλλά αν τα δεδομένα επικυρωθούν με την ίδια ετικέτα που πρότεινε ο κόμβος-αξιολογητής, τότε αυτός ανταμείβεται. Κάθε αξιολόγηση καταγράφεται στο Blockchain όπου ενημερώνεται ο πίνακας Votes της αντίστοιχης Caption δομής δεδομένων (βλ. Εικόνα 13). Στην Εικόνα 22 φαίνονται και δύο παραδείγματα καταχώρησης αξιολόγησης.

Εικόνα 22: DEMOS - Σελίδα Review

B.2.2.III.ε. Σελίδα Predict

Από αυτή τη σελίδα (βλ. Εικόνα 23) μπορεί ο καθένας να δοκιμάσει τα μοντέλα ΝΔ που έχουν καταχωρηθεί. Απλά επιλέγει ένα μοντέλο, δίνει ένα δεδομένο και το DEMOS εμφανίζει την ετικέτα που το μοντέλο προβλέπει πως ταιριάζει καλύτερα στο δεδομένο.

Στην ουσία μόλις ένας χρήστης πατήσει την επιλογή Predict Caption, αποστέλλεται ένα POST request στο /test API endpoint που περιέχει το δεδομένο που καταχώρησε (βλ. παράγραφο B.2.2.II Backend 2 – Διακομιστής (server) εκπαίδευσης ΝΔ, σελ.53). Μόλις ο Flask διακομιστής επεξεργαστεί το αίτημα αποστέλλει απάντηση με την ετικέτα που επέλεξε το μοντέλο ΝΔ. Στην Εικόνα 23 φαίνεται και το παράδειγμα εκτέλεσης για 3 μοντέλα ΝΔ.

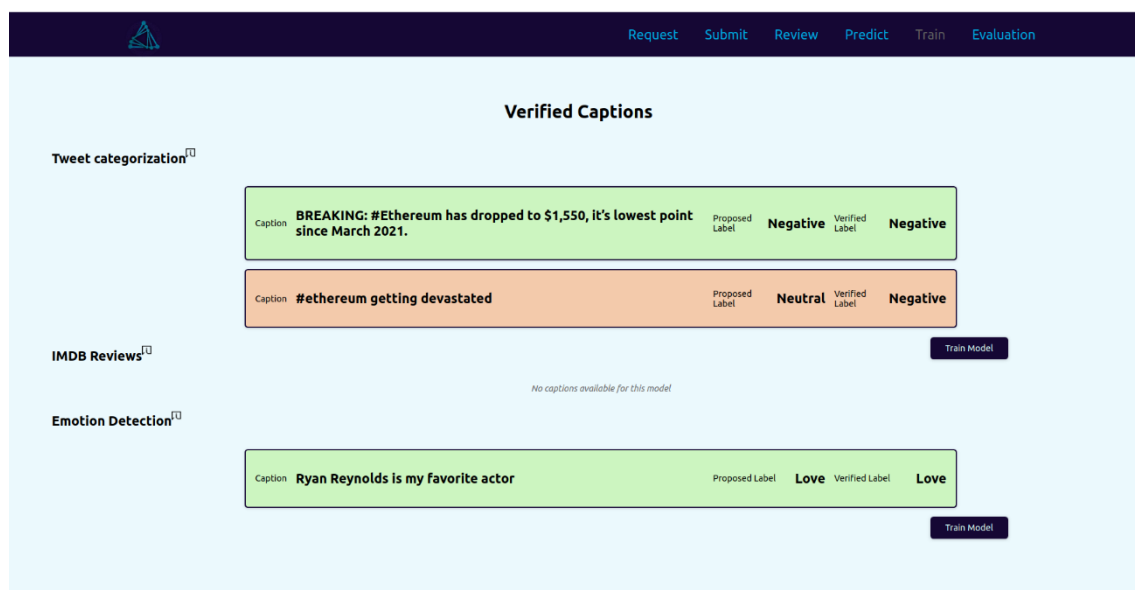
Εικόνα 23: DEMOS - Σελίδα Predict

B.2.2.III.στ. Σελίδα Train

Από τη σελίδα Train (βλ. Εικόνα 24) ένας κόμβος-αιτητής μπορεί να ζητήσει μετεκπαίδευση ενός από τα μοντέλα που έχει καταχωρήσει.

Ο κόμβος-αιτητής αρχικά βλέπει ανά μοντέλο που έχει αιτηθεί, όλα τα δεδομένα που έχουν επικυρωθεί και είναι έτοιμα για να χρησιμοποιηθούν για εκπαίδευση. Τα δεδομένα που βρίσκονται σε πράσινο πλαίσιο είναι δεδομένα όπου η αρχική ετικέτα που πρότεινε ο κόμβος-δωρητής συμφωνεί με την ετικέτα επικύρωσης που επέλεξαν οι κόμβοι-αξιολογητές και πορτοκαλί σε αντίθετη περίπτωση.

Μόλις ο κόμβος-αιτητής επιλέξει το Train Model, αποστέλλεται POST request στο /train API endpoint (βλ. παράγραφο B.2.2.II Backend 2 – Διακομιστής (server) εκπαίδευσης ΝΔ, σελ.53) που περιέχει τη λίστα δεδομένων εκπαίδευσης μαζί με τις επικυρωμένες ετικέτες (verified labels) και ξεκινά η μετεκπαίδευση του μοντέλου ΝΔ. Όταν ολοκληρωθεί η μετεκπαίδευση, ο Flask διακομιστής απαντά στο frontend ότι η εκπαίδευση ολοκληρώθηκε με επιτυχία. Τότε αποστέλλεται νέο GET request στο /evaluate API endpoint (βλ. παράγραφο B.2.2.II Backend 2 – Διακομιστής (server) εκπαίδευσης ΝΔ, σελ.53) για να ξεκινήσει η αξιολόγηση του μοντέλου ΝΔ χρησιμοποιώντας το τμήμα δεδομένων test. Μόλις ολοκληρωθεί και αυτή, ο Flask διακομιστής στέλνει απάντηση στο frontend που περιέχει την τιμή της ακρίβειας (Accuracy) και της απώλειας (Loss). Τέλος, καταχωρείται στο Blockchain η τιμή της ακρίβειας και συγκεκριμένα στον πίνακα Accuracy της αντίστοιχης ModelInfo μορφής δεδομένων (βλ. Εικόνα 12).



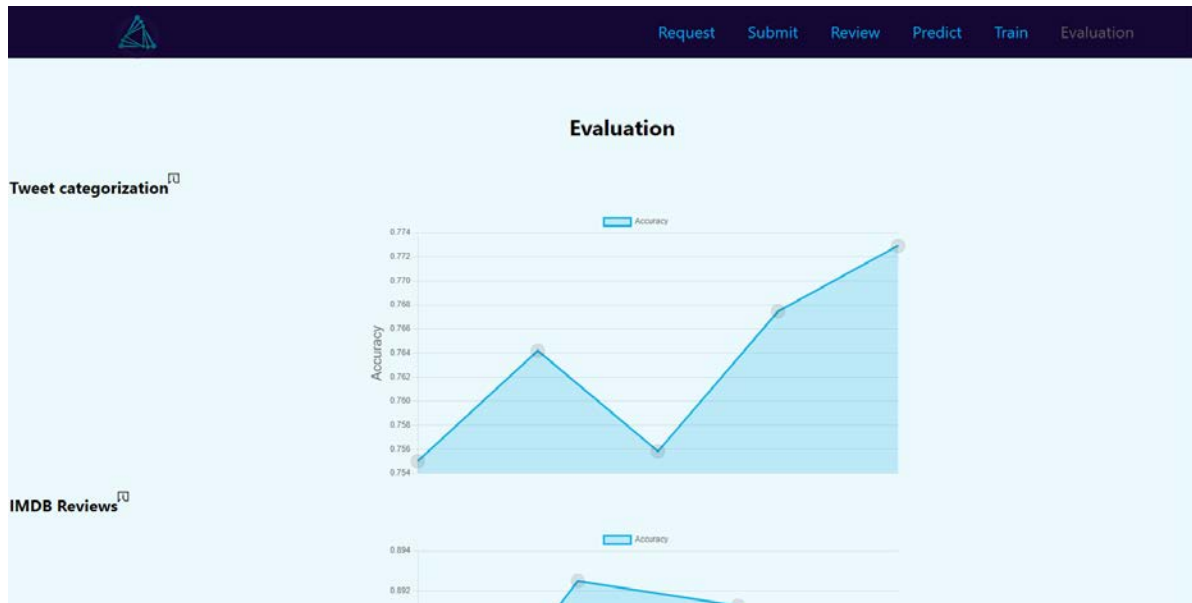
Εικόνα 24: DEMOS - Σελίδα Train

B.2.2.III.ζ. Σελίδα Evaluation

Από τη σελίδα Evaluation (βλ. Εικόνα 25) ένας κόμβος-αιτητής μπορεί να παρατηρήσει την πρόοδο των μοντέλων ΝΔ που έχει καταχωρήσει.

Κάθε φορά που ο κόμβος-αιτητής ζητά να γίνει εκπαίδευση του μοντέλου βάσει των δεδομένων που έχουν επικυρωθεί μέχρι εκείνη τη στιγμή, γίνεται η εκπαίδευση και

αποθηκεύεται η νέα τιμή της ακρίβειας στο Blockchain (βλ παράγραφο Β.2.2.III.στ Σελίδα Train, σελ.60). Παράλληλα τα δεδομένα που χρησιμοποιήθηκαν απορρίπτονται ώστε να μην ξαναχρησιμοποιηθούν. Στη σελίδα Evaluation ο κόμβος-αιτητής μπορεί να δει μία γραφική παράσταση που δείχνει τις μεταβολές της ακρίβειας του κάθε μοντέλου του.



Εικόνα 25: DEMOS - Σελίδα Evaluation

B.2.3. Εγκατάσταση

Η εγκατάσταση του DEMOS dApp απαιτεί κάποια διαδικασία που περιγράφεται αναλυτικά στον οδηγό εγκατάστασης (αρχείο README.md) που βρίσκεται στη σελίδα του DEMOS στο Gitlab [68]. Ακολουθεί μια σύνοψη της διαδικασίας.





B.2.3.I. Αρχική Εκπαίδευση

Το πρώτο βήμα που πρέπει να γίνει είναι η αρχική εκπαίδευση των μοντέλων. Γενικά το DEMOS μπορεί να υποστηρίξει διάφορα μοντέλα ΝΔ και σύνολα δεδομένων αλλά στα πλαίσια των δοκιμών χρησιμοποιήθηκε ένα μοντέλο που βασίζεται στην αναπαράσταση BERT (βλ. παράγραφο B.1.3 Μοντέλο Νευρωνικού Δικτύου, σελ.39) και 3 σύνολα δεδομένων από το Kaggle (βλ. παράγραφο B.1.3.I Σύνολα Δεδομένων (Datasets), σελ.40).

Τα 3 έτοιμα ipynb για αρχική εκπαίδευση που υπάρχουν στο φάκελο GoogleColab-backend (βλ. Εικόνα 26) φορτώνονται στο Google Drive και αφού τεθούν οι τιμές των παραμέτρων εκπαίδευσης, εκτελούνται. Αφότου ολοκληρωθεί η εκπαίδευση, γίνεται η αξιολόγησή του (βλ. Εικόνα 27) και αν ο χρήστης είναι ικανοποιημένος αποθηκεύει το προεκπαιδευμένο μοντέλο μαζί με τα τμήματα δεδομένων validation και test (βλ. Εικόνα 28).

Σημείωση: Η εκπαίδευση γίνεται μέσα από το Google Colab γιατί η χρήση GPU επιταχύνει την εκπαίδευση.

My Drive > Colab Notebooks > ml-blockchain ▾










Name ↑	Owner	Last modified
 Emotions-BERT-InitialTrainModel.ipynb	me	Jun 15, 2022
 FlaskServerForBert.ipynb	me	Jun 15, 2022
 IMDB-BERT-InitialTrainModel.ipynb	me	Jun 15, 2022
 Twitter-BERT-InitialTrainModel.ipynb	me	Jun 15, 2022

Εικόνα 26: Αρχική Εκπαίδευση - ipynb αρχεία

```
[ ] model.evaluate(test_data)
```

```
86/86 [=====] - 25s 289ms/step - loss: 0.9681 - accuracy: 0.7602  
[0.9680880904197693, 0.7601892352104187]
```

Εικόνα 27: Αρχική Εκπαίδευση - Αξιολόγηση εκπαίδευσης

Name ↑	Owner	Last modified
 BERTModel-Emotions	me	9:24 PM
 BERTModel-IMDB	me	8:46 PM
 BERTModel-Twitter	me	7:38 PM
 BERTModel-Emotions-test.csv	me	8:53 PM
 BERTModel-Emotions-validation.csv	me	8:53 PM
 BERTModel-IMDB-test.csv	me	7:45 PM
 BERTModel-IMDB-validation.csv	me	7:45 PM
 BERTModel-Twitter-test.csv	me	6:58 PM
 BERTModel-Twitter-validation.csv	me	6:58 PM

Εικόνα 28: Αρχική Εκπαίδευση - Αποθηκευμένα μοντέλα ΝΔ

B.2.3.II. Οργάνωση Ganache

Αφότου γίνει η αρχική εκπαίδευση πρέπει να δημιουργηθεί μέσω του Ganache το τοπικό Blockchain και να δημιουργηθούν οι λογαριασμοί των κόμβων που θα συμμετέχουν σε αυτό (βλ. Εικόνα 29).

ACCOUNTS					
<div> <div>CURRENT BLOCK</div> <div>GAS PRICE</div> <div>GAS LIMIT</div> <div>HARDFORK</div> <div>NETWORK ID</div> <div>RPC SERVER</div> <div>MINING STATUS</div> <div>WORKSPACE</div> </div>					
<div> <div>0</div> <div>20000000000</div> <div>6721975</div> <div>MURGLACIER</div> <div>5777</div> <div>HTTP://127.0.0.1:7545</div> <div>AUTOMINING</div> <div>BLOCKCHAIN-ML</div> </div>					
<div> <div> <div> <div>MMEMONIC</div> <div>any dolphin movie pear alert enrich ranch photo raccoon insect outdoor spoil</div> </div> <div> <div>HD PATH</div> <div>m/44'/60'/0'/0/account_index</div> </div> </div> </div>					
ADDRESS	BALANCE	TX COUNT	INDEX		
0x47137993DAba5B814951DBFF846f05115337BcfA	100.00 ETH	0	0		
ADDRESS	BALANCE	TX COUNT	INDEX		
0xDcc23b131bf98f461D8337a3730D190864D7AB70	100.00 ETH	0	1		
ADDRESS	BALANCE	TX COUNT	INDEX		
0x614946F9c8A88D0Cd23FC06aee9AA1434a5A0F3E	100.00 ETH	0	2		
ADDRESS	BALANCE	TX COUNT	INDEX		
0x4Bb6EB4dF9155778cd0eb2a62c0C069b13076fF	100.00 ETH	0	3		
ADDRESS	BALANCE	TX COUNT	INDEX		
0x5C4B0c26BfcbFEaD41dAEB5b18770224aC6871CF	100.00 ETH	0	4		
ADDRESS	BALANCE	TX COUNT	INDEX		
0xFbA534d3e518A5A57cde5fedc147e985cb76a49b	100.00 ETH	0	5		

Εικόνα 29: Οργάνωση Ganache - Λίστα λογαριασμών κόμβων

B.2.3.III. Οργάνωση Frontend

Μετά τη δημιουργία του τοπικού Blockchain πρέπει να παραταχθούν τα smart contracts στο Blockchain και να τεθεί σε λειτουργία το frontend του DEMOS. Τα smart contracts αποτελούν μέρος του truffle project που είναι ο φάκελος contracts και η παράταξή τους γίνεται με την εντολή: `truffle migrate --reset` (βλ. Εικόνα 30). Μέσω του Ganache μπορεί να επιβεβαιωθεί ότι η παράταξη έγινε σωστά (βλ. Εικόνα 31). Στη συνέχεια τίθεται σε λειτουργία το frontend το οποίο είναι το React.js project που είναι ο φάκελος client. Η ενεργοποίηση γίνεται στο port 7003 του localhost με την εντολή: `npm run dev`. Αν το frontend ενεργοποιηθεί σωστά θα φανεί το μήνυμα που φαίνεται στην Εικόνα 32 και θα εμφανιστεί το URL που οδηγεί στη σελίδα σύνδεσης (Login page) του DEMOS (βλ. Εικόνα 33)

```
PS C:\Users\Admin\Desktop\blockchain-ml-main\contracts> truffle migrate --reset

Compiling your contracts...
=====
> Compiling .\contracts\CaptionReview.sol
> Compiling .\contracts\Migrations.sol
> Artifacts written to C:\Users\Admin\Desktop\blockchain-ml-main\contracts\build\contracts
> Compiled successfully using:
   - solc: 0.8.12+commit.f00d7308.Emscripten.clang

Starting migrations...
=====
> Network name:      'development'
> Network id:        5777
> Block gas limit: 6721975 (0x6691b7)
```

Εικόνα 30: Οργάνωση Frontend - Παράταξη smart contracts

ACCOUNTS	BLOCKS	TRANSACTIONS	CONTRACTS	EVENTS	LOGS	SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 4	GAS PRICE 20000000000	GAS LIMIT 6721975	HARDFORK MUIRGLACIER	NETWORK ID 5777	RPC SERVER HTTP://127.0.0.1:7545	MINING STATUS AUTOMINING
WORKSPACE BLOCKCHAIN-ML					SWITCH	⚙
contracts C:\Users\Admin\Desktop\blockchain-ml-main\contracts						
NAME CaptionReview	ADDRESS 0x83bD96d5F63fF832B80F6646ED6372071e07d43	TX COUNT 0	DEPLOYED			
NAME Migrations	ADDRESS 0xbA0A7D0C7695bD5F160B3F3926B5218de904d8d9	TX COUNT 1	DEPLOYED			

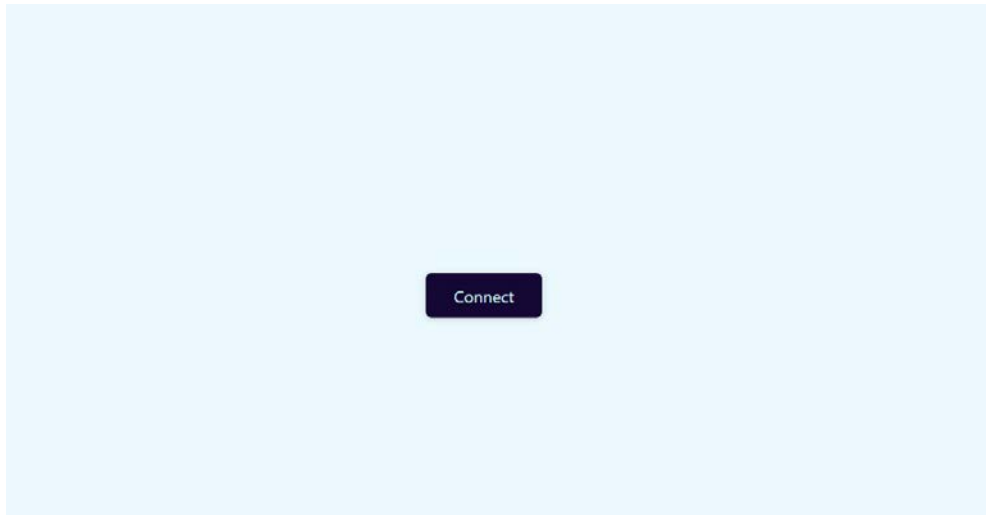
Εικόνα 31: Οργάνωση Frontend - Κατάσταση smart contracts μέσα από το Ganache

```
PS C:\Users\Admin\Desktop\blockchain-ml-main\client> npm run dev

> client@0.1.0 dev
> next dev -p 3007

ready - started server on 0.0.0.0:3007, url: http://localhost:3007
```

Εικόνα 32: Οργάνωση Frontend - Μήνυμα επιτυχούς ενεργοποίησης frontend



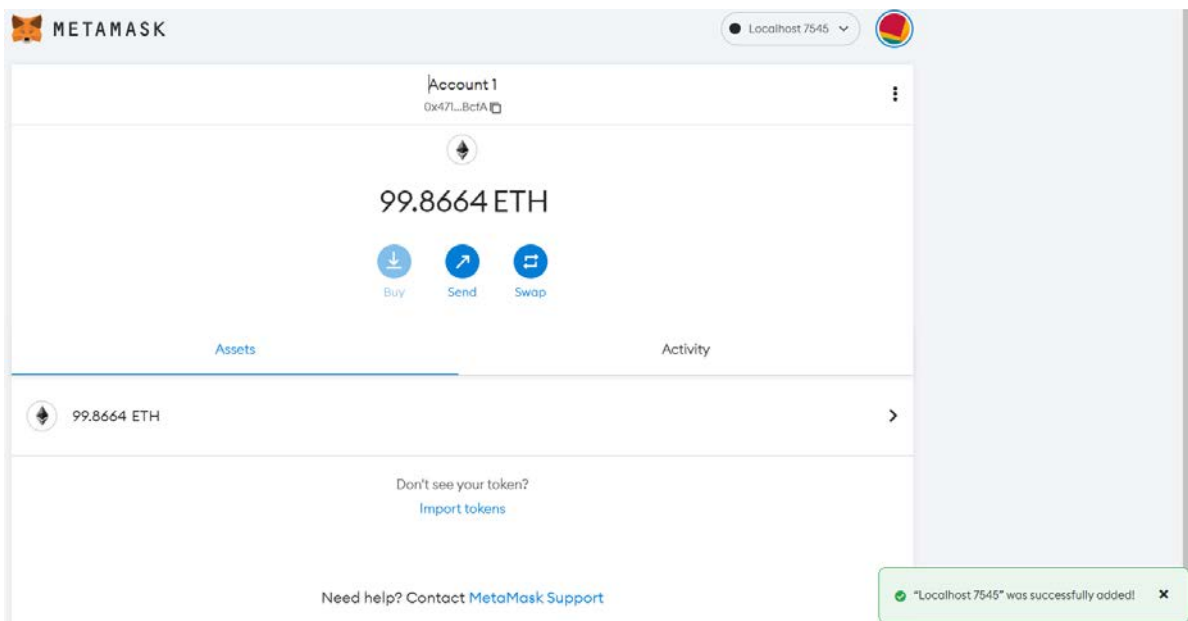
Εικόνα 33: Οργάνωση Frontend - Σελίδα σύνδεσης DEMOS

B.2.3.IV. Οργάνωση Metamask

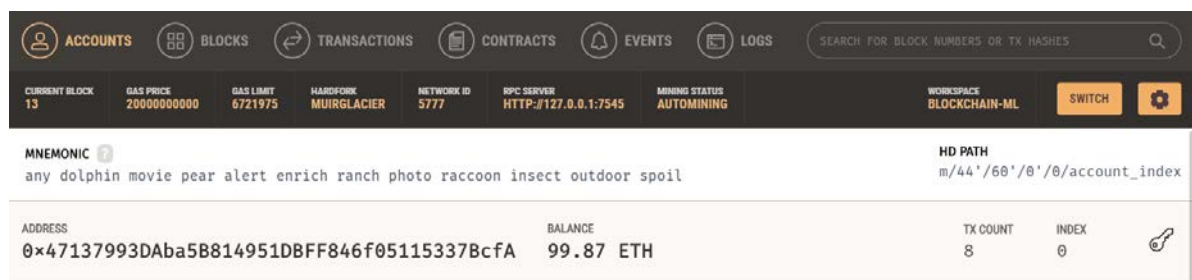
Ακολουθώντας, πρέπει να ενωθεί το Metamask με το Blockchain και με το frontend ώστε να μπορούν οι κόμβοι να συνδεθούν στο DEMOS και να εκτελέσουν τις διάφορες λειτουργίες. Χρησιμοποιώντας το κλειδί 12 λέξεων που παράγει το Ganache, ενώνεται το Metamask με το Blockchain (βλ. Εικόνα 34). Στην Εικόνα 35 φαίνεται ο λογαριασμός μέσα από το Metamask που συμπίπτει με το λογαριασμό που φαίνεται στην Εικόνα 36 μέσα από το Ganache. Στη συνέχεια, ενώνονται οι λογαριασμοί του Blockchain στο frontend μέσω του Metamask (βλ. Εικόνα 37). Μετά από αυτό μπορεί να γίνει σύνδεση στο DEMOS και να φανεί η αρχική του σελίδα (βλ Εικόνα 38)

A screenshot of the Metamask mobile app interface. At the top, there is a header with the Metamask logo and a '< Back' link. Below the header, the title 'Import a wallet with Secret Recovery Phrase' is displayed. A sub-note states: 'Only the first account on this wallet will auto load. After completing this process, to add additional accounts, click the drop down menu, then select Create Account.' The main section is titled 'Secret Recovery Phrase' and features a dropdown menu with the option 'I have a 12-word phrase'. Below this, a blue tip box says 'You can paste your entire secret recovery phrase into any field'. There are 12 input fields arranged in a 3x4 grid, each containing a masked character (dots) and a 'copy' icon. Below the grid, there are two password input fields: 'New password (8 characters min)' and 'Confirm password'. At the bottom, there is a checkbox labeled 'I have read and agree to the Terms of Use' which is checked, and a red 'Import' button.

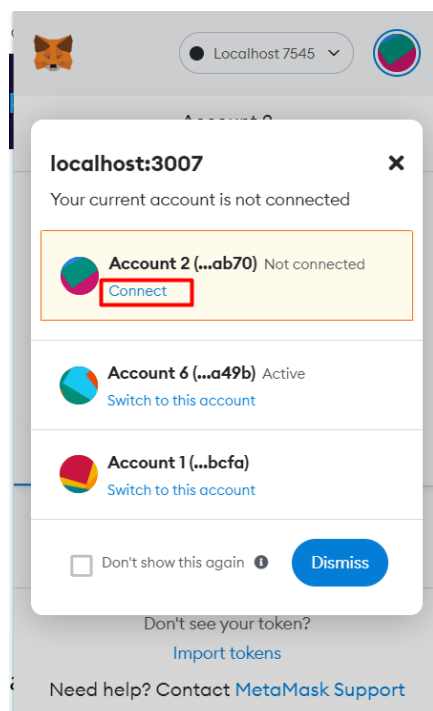
Εικόνα 34: Οργάνωση Metamask - σύνδεση Metamask – Blockchain



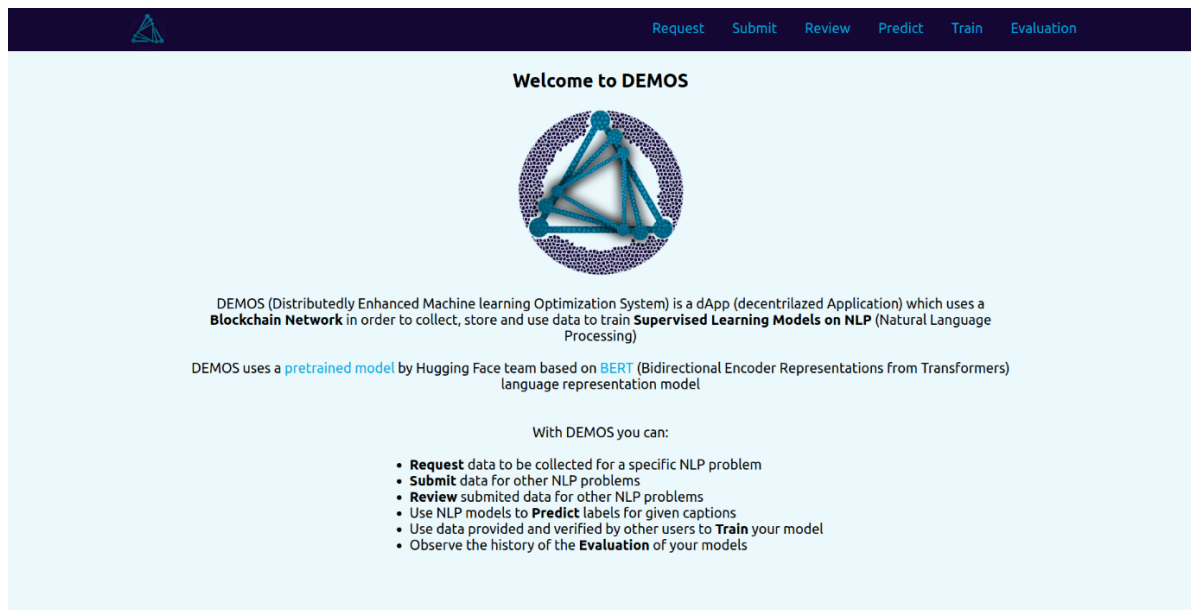
Εικόνα 35: Οργάνωση Metamask - Στοιχεία λογαριασμού μέσα από το Metamask



Εικόνα 36: Οργάνωση Metamask - Στοιχεία λογαριασμού μέσα από το Ganache



Εικόνα 37: Οργάνωση Metamask - Σύνδεση λογαριασμών Metamask με το frontend



Εικόνα 38: Οργάνωση Metamask - Αρχική σελίδα DEMOS

B.2.3.V. Οργάνωση Διακομιστή Flask στο Google Colab

Όταν το frontend έχει ενεργοποιηθεί έχει σειρά ο διακομιστής Flask που συνδέει το DEMOS με τα μοντέλα ΝΔ. Για να τρέξει ο διακομιστής Flask στο Google Colab εκτελείται το αρχείο FlaskServerForBert.ipynb. Αυτό ενεργοποιεί τον διακομιστή σε κάποια θύρα του μηχανήματος της Google. Για να είναι αυτή η θύρα προσβάσιμη από εξωτερικά μηχανήματα, όπως αυτό που τρέχει το frontend, χρησιμοποιείται το Ngrok που δίνει ένα URL. Μόλις ενεργοποιηθεί ο διακομιστής στο Google Colab και παραχθεί το URL του Ngrok (βλ. Εικόνα 39), αυτό αντιγράφεται στον κώδικα του frontend (βλ. Εικόνα 40) ώστε να κατευθύνονται σωστά τα API αιτήματα.

```
Server Running
* Serving Flask app "__main__" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Running on http://1a44-35-197-59-81.ngrok.io
* Traffic stats available on http://127.0.0.1:4040
```

Εικόνα 39: Οργάνωση Διακομιστή Flask στο Google Colab – Μήνυμα επιτυχούς ενεργοποίησης



Εικόνα 40: Οργάνωση Διακομιστή Flask στο Google Colab – Ορισμός διεύθυνσης αποστολής API αιτημάτων

B.2.3.VI. Οργάνωση Διακομιστή Flask τοπικά

Οι υπολογισμοί που απαιτούνται για την εκπαίδευση και αξιολόγηση των μοντέλων ΝΔ είναι αρκετά πολύπλοκοι. Αν το μηχάνημα εγκατάστασης του DEMOS έχει υλικό που μπορεί να τους υποστηρίξει, τότε η ενεργοποίηση του διακομιστή Flask μπορεί να γίνει τοπικά χωρίς το Google Colab και το Ngrok. Ο φάκελος backend περιέχει όλα όσα χρειάζονται για να γίνει τοπικά η ενεργοποίηση η οποία γίνεται στη θύρα 5000 με την εντολή flask run. Πρέπει επίσης να χρησιμοποιηθεί το σωστό μονοπάτι προς το localhost:5000 για να κατευθύνονται τα API αιτήματα στο τοπικό μηχάνημα (βλ. Εικόνα 41).



```
JS parameters.js M X
client > constants > JS parameters.js > ...
1  module.exports = {
2    ModelCost: "1",
3    CaptionCost: "0.001",
4    ReviewCost: "0.001",
5    // APIpath: "http://0211-35-245-112-159.ngrok.io",
6    APIpath: "http://127.0.0.1:5000",
7  };

```

Εικόνα 41: Οργάνωση Διακομιστή Flask τοπικά– Ορισμός διεύθυνσης αποστολής API αιτημάτων

B.2.3.VII. Επανεκκίνηση DEMOS

Αν τα προηγούμενα βήματα έχουν εκτελεστεί ορθά και έχει τρέξει σωστά το DEMOS dApp είναι αρκετά απλή η επανεκκίνησή του. Πρέπει να γίνουν τα εξής:

1. Εκκίνηση του Ganache και επιλογή του Blockchain που δημιουργήθηκε πιο πριν
2. Ενεργοποίηση του frontend με την εντολή: `npm run dev` στο φάκελο client
3. Άνοιγμα του DEMOS στον browser
4. Ένωση στο DEMOS μέσω του Metamask
5. Ενεργοποίηση διακομιστή Flask
 - ο Στο Google Colab τρέχοντας το FlaskServerForBert.ipynb και αντιγράφοντας το URL του Ngrok
 - ο Τοπικά με την εντολή: `run flask` στο φάκελο backend

B.3. ΑΠΟΤΕΛΕΣΜΑΤΑ & ΣΥΜΠΕΡΑΣΜΑΤΑ

ffffffffff



Συνεχής ανανέωση, αναβάθμιση του dataset

Αμετάβλητη βάση δεδομένων

Εξασφάλιση ορθότητας δεδομένων

Κατανομή εργασίας

θα μπορούσε να εκπαιδεύσει και μοντέλα που δεν έχουν προεκπαιδευτεί

B.4. ΠΕΡΙΟΡΙΣΜΟΙ & ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΒΕΛΤΙΩΣΗ

Model version

New path

Not local server

Model χειροκίνητο πέρασμα

Αλλαγή παραμέτρων από κόμβους

Μέθοδος αξιολόγησης (μεταβλητό accuracy)

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Y. Lan, Y. Liu, and B. Li, “Proof of Learning (PoLe): Empowering Machine Learning with Consensus Building on Blockchains,” Jul. 2020, [Online]. Available: <http://arxiv.org/abs/2007.15145>
- [2] “What Happens When You Combine Blockchain and Machine Learning - Intersog.” <https://intersog.com/blog/what-happens-when-you-combine-blockchain-and-machine-learning/> (accessed Jun. 09, 2022).
- [3] W. Stallings and L. Brown, *Ασφάλεια Υπολογιστών: Αρχές και Πρακτικές*, 3η έκδοση. Αθήνα: Κλειδάριθμος, 2016.
- [4] Μ. Αναγνώστου, “Σημειώσεις μαθήματος Ασφάλειας Δικτύων Υπολογιστών,” 2021.
- [5] A. Antonopoulos and G. Wood, “4. Cryptography - Mastering Ethereum [Book].” <https://www.oreilly.com/library/view/mastering-ethereum/9781491971932/ch04.html> (accessed May 20, 2022).
- [6] “What is Public-key Cryptography? :: What is Public-key Cryptography? :: GlobalSign GMO Internet, Inc.” <https://www.globalsign.com/en/ssl-information-center/what-is-public-key-cryptography> (accessed May 21, 2022).
- [7] “Elliptic Curve Cryptography: A Basic Introduction | Boot.dev.” <https://blog.boot.dev/cryptography/elliptic-curve-cryptography/> (accessed May 21, 2022).
- [8] “What is Asymmetric Cryptography? Definition from SearchSecurity.” <https://www.techtarget.com/searchsecurity/definition/asymmetric-cryptography> (accessed May 21, 2022).
- [9] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008. [Online]. Available: www.bitcoin.org
- [10] V. Buterin, “Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform.,” 2014.
- [11] “Sybil Attacks Explained | Binance Academy.” <https://academy.binance.com/en/articles/sybil-attacks-explained> (accessed May 22, 2022).
- [12] Bina Ramamurthy, “Blockchain Basics,” *Coursera*. Accessed: Jun. 22, 2022. [Online]. Available: <https://www.coursera.org/learn/blockchain-basics>
- [13] “Mining | ethereum.org.” <https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/mining/> (accessed May 22, 2022).
- [14] “Consensus mechanisms | ethereum.org.” <https://ethereum.org/en/developers/docs/consensus-mechanisms/> (accessed May 23, 2022).
- [15] “Proof-of-work (PoW) | ethereum.org.” <https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/> (accessed May 23, 2022).
- [16] “Proof-of-stake (PoS) | ethereum.org.” <https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/> (accessed May 23, 2022).
- [17] “Review of blockchain consensus mechanisms | by Gleb Kostarev | Waves Protocol | Medium.” <https://medium.com/wavesprotocol/review-of-blockchain-consensus-mechanisms-f575afae38f2> (accessed May 23, 2022).
- [18] “A (Short) Guide to Blockchain Consensus Protocols - CoinDesk.” <https://www.coindesk.com/markets/2017/03/04/a-short-guide-to-blockchain-consensus-protocols/> (accessed May 23, 2022).

- [19] “Ethereum - Wikipedia.” <https://en.wikipedia.org/wiki/Ethereum> (accessed May 25, 2022).
- [20] “Web 1.0 and Web 2.0 | BYU McKay School of Education.” <https://education.byu.edu/alumni/teach-admin/tech-tips/web-2.0> (accessed May 25, 2022).
- [21] “Web 1.0 vs Web 2.0: Full Comparison - History Computer.” <https://history-computer.com/web-1-0-vs-web-2-0-full-comparison/> (accessed May 25, 2022).
- [22] “What is Web3 and why is it important? | ethereum.org.” <https://ethereum.org/en/web3/> (accessed May 25, 2022).
- [23] “Web2 vs Web3 | ethereum.org.” <https://ethereum.org/en/developers/docs/web2-vs-web3/> (accessed May 25, 2022).
- [24] “Intro to Ethereum | ethereum.org.” <https://ethereum.org/en/developers/docs/intro-to-ethereum/> (accessed May 21, 2022).
- [25] “Ethereum Virtual Machine (EVM) | ethereum.org.” <https://ethereum.org/en/developers/docs/evm/> (accessed May 26, 2022).
- [26] Bina Ramamurthy, “Smart Contracts,” *Coursera*. Accessed: Jun. 22, 2022. [Online]. Available: <https://www.coursera.org/learn/smarter-contracts>
- [27] “Introduction to smart contracts | ethereum.org.” <https://ethereum.org/en/developers/docs/smart-contracts/> (accessed May 26, 2022).
- [28] “Compiling smart contracts | ethereum.org.” <https://ethereum.org/en/developers/docs/smart-contracts/compiling/> (accessed May 26, 2022).
- [29] “Structure of a Contract — Solidity 0.8.15 documentation.” <https://docs.soliditylang.org/en/develop/structure-of-a-contract.html> (accessed May 26, 2022).
- [30] “Smart contracts | ethereum.org.” <https://ethereum.org/en/smart-contracts/> (accessed May 26, 2022).
- [31] “Ethereum accounts | ethereum.org.” <https://ethereum.org/en/developers/docs/accounts/> (accessed May 21, 2022).
- [32] Bina Ramamurthy, “Decentralized Applications (Dapps),” *Coursera*.
- [33] “What is a dApp? Decentralized Application on the Blockchain.” <https://blockchainhub.net/decentralized-applications-dapps/> (accessed May 27, 2022).
- [34] “Decentralized Application (DApp) | Binance Academy.” <https://academy.binance.com/en/glossary/decentralized-application> (accessed May 27, 2022).
- [35] “Introduction to dapps | ethereum.org.” <https://ethereum.org/en/developers/docs/dapps/> (accessed May 27, 2022).
- [36] “Decentralized Applications (dApps) Definition.” <https://www.investopedia.com/terms/d/decentralized-applications-dapps.asp> (accessed May 27, 2022).
- [37] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, “Dive into Deep Learning.” arXiv, 2021. doi: 10.48550/ARXIV.2106.11342.
- [38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [39] “MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges.” <http://yann.lecun.com/exdb/mnist/> (accessed Jun. 08, 2022).
- [40] “What is Natural Language Processing? | IBM.” <https://www.ibm.com/cloud/learn/natural-language-processing> (accessed Jun. 08, 2022).

- [41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *CoRR*, vol. abs/1810.04805, 2018, [Online]. Available: <http://arxiv.org/abs/1810.04805>
- [42] J. D. Harris and B. Waggoner, "Decentralized and collaborative AI on blockchain," in *Proceedings - 2019 2nd IEEE International Conference on Blockchain, Blockchain 2019*, Jul. 2019, pp. 368–375. doi: 10.1109/Blockchain.2019.00057.
- [43] J. D. Harris, "Analysis of Models for Decentralized and Collaborative AI on Blockchain," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020, vol. 12404 LNCS, pp. 142–153. doi: 10.1007/978-3-030-59638-5_10.
- [44] T. Addair, "Decentralized and Distributed Machine Learning Model Training with Actors," 2017. [Online]. Available: <https://blog.skymind.ai/distributed-deep-learning-part-1-an>
- [45] "Sentiment Analysis in 10 Minutes with BERT and TensorFlow | by Orhan G. Yalçın | Medium | Towards Data Science." <https://towardsdatascience.com/sentiment-analysis-in-10-minutes-with-bert-and-hugging-face-294e8a04b671> (accessed Jun. 22, 2022).
- [46] "bert-base-uncased · Hugging Face." <https://huggingface.co/bert-base-uncased> (accessed Jun. 12, 2022).
- [47] "Hugging Face – The AI community building the future." <https://huggingface.co/> (accessed Jun. 13, 2022).
- [48] "Tokenizer." https://huggingface.co/docs/transformers/main_classes/tokenizer (accessed Jun. 12, 2022).
- [49] "Kaggle: Your Home for Data Science." <https://www.kaggle.com/> (accessed Jun. 18, 2022).
- [50] "Twitter Tweets Sentiment Dataset | Kaggle." <https://www.kaggle.com/datasets/yasserh/twitter-tweets-sentiment-dataset> (accessed Jun. 12, 2022).
- [51] "IMDB Movie Ratings Sentiment Analysis | Kaggle." <https://www.kaggle.com/datasets/yasserh/imdb-movie-ratings-sentiment-analysis?select=movie.csv> (accessed Jun. 12, 2022).
- [52] "Emotions in text | Kaggle." <https://www.kaggle.com/datasets/ishantjuyal/emotions-in-text> (accessed Jun. 12, 2022).
- [53] "Classification: Accuracy | Machine Learning Crash Course | Google Developers." <https://developers.google.com/machine-learning/crash-course/classification/accuracy> (accessed Jun. 21, 2022).
- [54] "Remix - Ethereum IDE." <https://remix.ethereum.org> (accessed Jun. 11, 2022).
- [55] "Truffle - Truffle Suite." <https://trufflesuite.com/truffle/> (accessed Jun. 11, 2022).
- [56] "Ganache - Truffle Suite." <https://trufflesuite.com/ganache/> (accessed Jun. 11, 2022).
- [57] "The crypto wallet for Defi, Web3 Dapps and NFTs | MetaMask." <https://metamask.io/> (accessed Jun. 11, 2022).
- [58] "React – A JavaScript library for building user interfaces." <https://reactjs.org/> (accessed Jun. 11, 2022).
- [59] "Next.js by Vercel - The React Framework." <https://nextjs.org/> (accessed Jun. 11, 2022).
- [60] "Welcome to Flask — Flask Documentation (2.1.x)." <https://flask.palletsprojects.com/en/2.1.x/> (accessed Jun. 12, 2022).
- [61] "Welcome To Colaboratory - Colaboratory." <https://colab.research.google.com/> (accessed Jun. 12, 2022).
- [62] "TensorFlow." <https://www.tensorflow.org/> (accessed Jun. 12, 2022).

- [63] “🤗 Transformers.” <https://huggingface.co/docs/transformers/index> (accessed Jun. 12, 2022).
- [64] “ngrok - Online in One Line.” <https://ngrok.com/> (accessed Jun. 12, 2022).
- [65] “Projects · Dashboard · GitLab.” <https://gitlab.com/> (accessed Jun. 19, 2022).
- [66] “Ideal Modeling & Diagramming Tool for Agile Team Collaboration.” <https://www.visual-paradigm.com/> (accessed Jun. 21, 2022).
- [67] “pandas.DataFrame — pandas 1.4.2 documentation.” <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html> (accessed Jun. 13, 2022).
- [68] “NETMODE / blockchain-ml · GitLab.” <https://gitlab.com/netmode/blockchain-ml/> (accessed Jun. 18, 2022).