



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΟΡΥΚΤΩΝ ΠΟΡΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Χρήση νευρωνικών δικτύων ή/και νευροασαφών συστημάτων για την εκτίμηση αποθεμάτων ποιότητας κοιτασμάτων και εύρεση βέλτιστων ορίων εκσκαφής

Νάσαϊ Ρισάλντ

Εξεταστική Επιτροπή:

Καθηγητής Γαλετάκης Μιχαήλ (Επιβλέπων)
Επίκουρος Καθηγητής Ξηρουδάκης Γεώργιος
Δρ. Σαράτσης Γεώργιος

Χανιά
Νοέμβριος, 2023

Οι απόψεις που εκφράζονται στην παρούσα εργασία αποτελούν την προσωπική τοποθέτηση του συγγραφέα και δεν αντικατοπτρίζουν τις θέσεις της τριμελούς εξεταστικής επιτροπής. Παρατίθεται πλήρης βιβλιογραφική λίστα για όλες τις πηγές που έχουν χρησιμοποιηθεί είτε αυτούσιες είτε παραφρασμένες μέσα στο κείμενο.

Ευχαριστίες

Η παρούσα διπλωματική εργασία πραγματοποιήθηκε στη Σχολή Μηχανικών Ορυκτών Πόρων του Πολυτεχνείου Κρήτης, με επιβλέποντα καθηγητή τον κ. Γαλετάκη Μιχαήλ.

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή της διπλωματικής μου κ. Γαλετάκη Μιχαήλ για την πολύτιμη βοήθειά του, την καθοδήγηση που μου προσέφερε και για τις γνώσεις που μου παρείχε τόσο κατά την εκπόνηση της εργασίας, όσο και κατά τη διάρκεια των σπουδών μου. Επίσης την συνεργάτιδα του εργαστηρίου κα Ανθούλα Βασιλείου Μηχ. Ορυκτών Πόρων MSc, για την βοήθεια της κατά τη συγγραφή της εργασίας. Να ευχαριστήσω επίσης τα μέλη της εξεταστικής επιτροπής Επικ. Καθηγητή κ. Ξηρουδάκη και Δρ. Γεώργιο Σαράτση για τα σχόλια τους και τη συμμετοχή τους.

Τέλος, οφείλω να ευχαριστήσω την οικογένειά μου, για τη συμπαράσταση και την υπομονή τους.

Περίληψη

Ο υπολογισμός της περιεκτικότητας και της εκτίμησης των αποθεμάτων ενός κοιτάσματος καθώς και ο καθορισμός των βέλτιστων ορίων εκσκαφής είναι μια εξαιρετικά σημαντική διαδικασία, καθώς αποτελεί βασικό πυλώνα της μελέτης και του σχεδιασμού μιας εκμετάλλευσης. Οι πιο γνωστές μέθοδοι υπολογισμού της περιεκτικότητας είναι η γεωστατιστική μέθοδος Kriging, οι μέθοδοι των αντιστρόφων αποστάσεων και οι γεωμετρικές μέθοδοι. Ο δε καθορισμός των βέλτιστων ορίων εκσκαφής είναι σημαντικός, διότι η υπερεκτίμηση μπορεί να αυξήσει τη σχέση αποκάλυψης, ενώ η υποεκτίμηση μπορεί να οδηγήσει σε εγκατάλειψη τμήματος του κοιτάσματος με οικονομικό ενδιαφέρον. Οι πιο γνωστές μέθοδοι προσδιορισμού των ορίων εκσκαφής βασίζονται στον γραμμικό και δυναμικό προγραμματισμό, στη μέθοδο του κινητού κώνου και στη θεωρία των γραφημάτων. Η πρόοδος των τελευταίων ετών πάνω στον τομέα της τεχνητής νοημοσύνης προσφέρει τη δυνατότητα εφαρμογής τους σε τέτοια μη γραμμικά προβλήματα χωρίς τις απλοποιήσεις και παραδοχές που συνήθως απαιτούν οι συμβατικές μέθοδοι.

Στην εργασία αυτή μελετήθηκε η δυνατότητα εκτίμησης της περιεκτικότητας και των αποθεμάτων σε ένα κοιτάσμα χαλκού, με την ανάπτυξη του ψηφιακού μοντέλου κοιτάσματος σε μορφή στοιχειωδών όγκων (blocks), με τη χρήση νευρωνικών δικτύων καθώς και ο καθορισμός στη συνέχεια των βέλτιστων ορίων εκσκαφής με την ενσωμάτωση οικονομικών στοιχείων και την εφαρμογή του αλγόριθμου “μέγιστης ροής” (Pseudoflow).

Αναπτύχθηκαν τρεις διαφορετικές παραλλαγές νευρωνικών δικτύων τα οποία όλα ήταν δίκτυα ευθείας προώθησης με οπισθοδιάδοση του σφάλματος. Στο πρώτο χρησιμοποιήθηκαν για την εκπαίδευση όλα τα σύνθετα δείγματα των γεωτρήσεων και με ένα κρυφό επίπεδο, στο δεύτερο με ένα επίπεδο και τροποποίηση των δεδομένων της εκπαίδευσης για βελτίωση της αποτελεσματικότητας του καθόσον στο πρώτο γινόταν μια σημαντική υποεκτίμηση των υψηλών περιεκτικοτήτων χαλκού και το τελευταίο με δύο κρυφά επίπεδα και τροποποίηση των δεδομένων της εκπαίδευσης. Το νευρωνικό δίκτυο αυτό πέτυχε ικανοποιητικά αποτελέσματα και με μικρότερο χρόνο εκπαίδευσής σε σχέση με το δεύτερο.

Τα αποτελέσματα του νευρωνικού δικτύου συγκρίθηκαν με αυτά της μεθόδου Kriging και το τετράγωνο των αντιστρόφων αποστάσεων (IDS) και έδειξαν ότι τα νευρωνικά δίκτυα μπορούν να αποδώσουν καλά τις χωρικές μεταβολές του χαλκού και ότι μπορούν να εντοπίσουν περιοχές με υψηλή περιεκτικότητα χαλκού που δεν ήταν δυνατόν να γίνει με τις μεθόδους Kriging και IDS.

Στη συνέχεια στα αποτελέσματα του ψηφιακού μοντέλου του κοιτάσματος που προέκυψε από το νευρωνικό δίκτυο και τις μεθόδους Kriging και IDS ενσωματώθηκαν οικονομικά στοιχεία και υπολογίστηκε η οικονομική αξία των blocks. Τέλος εφαρμόστηκε ο αλγόριθμος Pseudoflow για την εύρεση των βέλτιστων ορίων εκσκαφής. Τα αποτελέσματα έδειξαν ότι παρά το γεγονός ότι ο αριθμός των μπλοκ εντός της βέλτιστης εκσκαφής, δεν διαφέρει σημαντικά για τις τρεις μεθόδους, οι διαφορές ανάμεσα στην οικονομική αξία είναι σημαντικές, καθώς και η μέση οικονομική αξία ανά μπλοκ. Η διαφορά αυτή των οικονομικών τιμών που προκύπτει για τα μπλοκ της βέλτιστης εκσκαφής σχετίζεται άμεσα με τις εκτιμήσεις της περιεκτικότητας κάθε μεθόδου.

Λέξεις Κλειδιά: Εκτίμηση αποθεμάτων, ποιότητας, νευρωνικά δίκτυα, βέλτιστα όρια εκσκαφής

Abstract

The grade and reserve estimation of a deposit and determining the ultimate pit limit is an extremely important process, as it is a key pillar of the design and planning of the open pit mine. The best-known methods are the geostatistical method Kriging, the inverse distance methods, and geometrical methods. Finding the ultimate pit limit is important because overestimation may result in increased stripping ratio, while underestimation can lead to the rejection of a part of the deposit of economic interest. The best-known methods of finding the ultimate pit limit are based on linear and dynamic programming, the floating cone and graph theory. Recent advances in the field of artificial intelligence offer the possibility of applying them to such nonlinear problems without simplifications and assumptions that conventional methods usually require.

In this paper, it was studied the possibility of estimating the grade and resources of a copper deposit by developing the digital deposit model in the shape of blocks using neural networks and then determining the ultimate pit limits including economic elements and applying the “maximum flow” algorithm (Pseudoflow).

Three different variants of neural networks were developed which all were feedforward networks with back propagation. In the one all the composite drilling samples were used for training with one hidden layer, the second one with one hidden layer and resampling of the training data to improve its efficiency since the first one had a significant underestimation of the high copper grade and finally one with two hidden layers and resampling of the training data. This neural network achieved satisfactory results and with shorter training time than the second one.

The neural network results were compared with those of the Kriging and the inverse distance squared (IDS) methods and that they can identify areas of high copper grade that were not possible with Kriging and IDS methods. Economic data were then included in the results of the digital deposit model produced by the neural network, Kriging and IDS methods and the economic values of the blocks were calculated. Finally, the Pseudoflow algorithm was applied to find the ultimate pit limit. The results showed that although the differences in the number of blocks within the limits are not significantly different for the three methods, the differences between the economic value are significant as well as the average economic value per block. This difference in economic value of the blocks in the ultimate pit limit is directly related to the grade estimation of each method.

Keywords: Ore reserve and grade estimation, neural networks, ultimate pit limits

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη	iv
Abstract	v
Κατάλογος σχημάτων	vii
Κατάλογος πινάκων	ix
1 Εισαγωγή.....	1
2 Υφιστάμενη μεθοδολογία εκτίμησης αποθεμάτων	3
2.1 Υφιστάμενες μέθοδοι υπολογισμού αποθεμάτων και ποιότητας.....	3
2.1.1 Γεωμετρικές μέθοδοι υπολογισμού αποθεμάτων.....	3
2.1.2 Ανάπτυξη ψηφιακών μοντέλων του κοιτάσματος	3
2.1.3 Μέθοδος αντιστρόφων αποστάσεων (Inverse Distance)	4
2.1.4 Μέθοδος Kriging.....	4
2.2 Μέθοδοι καθορισμού των βέλτιστων ορίων εκσκαφής υπαίθριας εκμετάλλευσης.....	6
2.2.1 Μέθοδος κινητού κώνου	6
2.2.2 Αλγόριθμος Lerchs-Grossman.....	6
3 Τεχνητά νευρωνικά δίκτυα	7
3.1 Χαρακτηριστικά και δομή τεχνητών νευρωνικών δικτύων	7
3.2 Μέθοδοι εκπαίδευσης τεχνητών νευρωνικών δικτύων	8
4 Επεξεργασία-αξιολόγηση δεδομένων	11
4.1 Περιγραφή του κοιτάσματος χαλκού.....	11
4.2 Πρωτογενή στοιχεία γεωτρήσεων και δημιουργία σύνθετων δειγμάτων	12
4.3 Στατιστική ανάλυση και υπολογισμός ημιβαριογράμματος για τα σύνθετα δείγματα	13
5 Ανάπτυξη και εφαρμογή νευρωνικών δικτύων.....	17
5.1 Ανάπτυξη νευρωνικού δικτύου.....	17
5.2 Εφαρμογή νευρωνικού δικτύου για την κατασκευή του ψηφιακού μοντέλου του κοιτάσματος και τον υπολογισμό των αποθεμάτων και της ποιότητας	19
5.3 Εφαρμογή του νευρωνικού δικτύου για την ανάπτυξη του ψηφιακού μοντέλου και σύγκριση με τη μέθοδο IDS και Kriging	22
6 Αλγόριθμος ψευδοροής για την εύρεση των βέλτιστων ορίων.....	26
6.1 Από το Lerchs-Grossmann στον Pseudoflow: μια σύντομη αναδρομή	26
6.2 Αλγόριθμος ψευδοροής.....	26

6.2.1	Έννοιες από τη θεωρία των γράφων και τη βελτιστοποίηση των ορίων εκσκαφής μια υπαίθριας εκμετάλλευσης	27
6.2.2	Το πρόβλημα του εντοπισμού των βέλτιστων ορίων μιας εκσκαφής με τον αλγόριθμο ψευδοροής (Pseudoflow algorithm).....	28
6.2.3	Υπολογισμός του μέγιστου κλειστού υποσυνόλου με τον αλγόριθμο ψευδοροής	29
6.3	Εφαρμογή του αλγορίθμου Pseudoflow στο κοίτασμα Andina.....	33
7	Συμπεράσματα και προτάσεις.....	36
	Βιβλιογραφία	37
	Διεθνής βιβλιογραφία	37
	Ελληνική βιβλιογραφία.....	38
	Παράρτημα I: Κώδικας σε γλώσσα προγραμματισμού python, για την στατιστική και χωρική ανάλυση των γεωτρητικών δεδομένων	40
	Παράρτημα II: Κώδικας σε γλώσσα προγραμματισμού python, για τα αναπτυχθέντα νευρωνικά δίκτυα και την δημιουργία χαρτών.....	43
	Παράρτημα III: Κώδικας σε γλώσσα προγραμματισμού python, για τον αλγόριθμο υπολογισμού των βέλτιστων ορίων εκσκαφής.....	47

Κατάλογος σχημάτων

Σχήμα 3.1	Δομή δικτύου ευθείας προώθησης (feedforward).....	8
Σχήμα 3.2	Δομή δικτύου αναδρομικού τύπου (feedback).....	8
Σχήμα 3.3	Απεικόνιση της τεχνικής έγκαιρης διακοπής της εκπαίδευσης.....	10
Σχήμα 4.1	Χάρτης γεωτρήσεων κοιτάσματος (διαστάσεις κατά τους άξονες X και Y σε ft).....	12
Σχήμα 4.2	Δημιουργία σύνθετου δείγματος βαθμίδας με βάση τους πυρήνες των γεωτρήσεων	13
Σχήμα 4.3	Ιστόγραμμα της περιεκτικότητας σε % Cu για τα σύνθετα δείγματα.....	14
Σχήμα 4.4	Ιστόγραμμα αποτελεσμάτων λογαρίθμου της περιεκτικότητας σε % Cu των σύνθετων δειγμάτων.....	15
Σχήμα 4.5	Ημιβαριόγραμμα λογαριθμισμένων περιεκτικοτήτων	16
Σχήμα 5.1	Δομή επιλεχθέντα νευρωνικού δικτύου.....	18
Σχήμα 5.2	Ιστόγραμμα κατανομής της περιεκτικότητας σε Cu % των αρχικών δεδομένων εκπαίδευσης και των τροποποιημένων με βάση την τεχνική της αναδειγματοληψίας	19
Σχήμα 5.3	Διάγραμμα μέσου τετραγωνικού σφάλματος σε σχέση με τους κύκλους εκπαίδευσης (epochs).	

.....	20
Σχήμα 5.4 Διάγραμμα τιμών μοντέλου και αληθινών τιμών για τα δεδομένα εκπαίδευσης (training).....	20
Σχήμα 5.5 Διάγραμμα τιμών μοντέλου και αληθινών τιμών για τα δεδομένα ελέγχου (testing).....	21
Σχήμα 5.6 Κατανομή του σφάλματος εκτίμησης της περιεκτικότητας σε % Cu από το νευρωνικό δίκτυο.....	22
Σχήμα 5.7 Τρισδιάστατη απεικόνιση του ψηφιακού μοντέλου.....	22
Σχήμα 5.8 Διάγραμμα συσχέτισης των περιεκτικοτήτων σε % Cu των νευρωνικών δικτύων σε σχέση με την μέθοδο Kriging.....	24
Σχήμα 5.9 Διάγραμμα συσχέτισης των περιεκτικοτήτων σε % Cu των νευρωνικών δικτύων σε σχέση με την μέθοδο IDS.....	24
Σχήμα 6.1 (α) Απλό 2D μοντέλο 10 μπλοκ (β) Εξαρτήσεις μεταξύ των μπλοκ λόγω της απαιτούμενης κλίσης εκμετάλλευσης και της διαδοχής των μπλοκ	26
Σχήμα 6.2. Κατευθυνόμενο γράφημα που προέκυψε από το 2D μοντέλο των 10 μπλοκ του σχήματος 5.1α	27
Σχήμα 6.3. Περιπτώσεις 1 και 2: Κλειστά υποσύνολα (closed set) του κατευθυνόμενου γραφήματος. Περίπτωση 3: Μη κλειστό υποσύνολο (non-closed set). Περίπτωση 4: Μέγιστο κλειστό υποσύνολο	28
Σχήμα 6.4. 1) Σύνδεση κόμβων με κόμβο source και κόμβο sink και απόδοση βαρών στα τόξα. 2) Απόδοση τυχαίων τιμών λαμβάνοντας υπόψη τις μέγιστες επιτρεπόμενες και τα ισοζύγια ροής.....	29
Σχήμα 6.5 Εύρεση βέλτιστων ορίων με το γράφημα μέγιστης ροής.....	30
Σχήμα 6.6 Μεταβολή του χρόνου εύρεσης των βέλτιστων ορίων ορυχείου σε σχέση με τον αριθμό των μπλοκ για τις μεθόδους LG και Pseudoflow [Geovia, Whittle – Pseudoflow Method for Pit Optimization white paper].....	31
Σχήμα 6.7 Καμπύλη αποθεμάτων-ποιότητας με βάση τα νευρωνικά δίκτυα, τη μέθοδο Kriging, και τη μέθοδο IDS.....	32
Σχήμα 6.8 Χάρτες ισοπεριεκτικότητας Cu % που προέκυψαν από τις εκτιμήσεις των τεχνητών νευρωνικών δικτύων, Kriging .και IDS.....	33

Κατάλογος πινάκων

Πίνακας 4.1 Στατιστικά μεγέθη για την περιεκτικότητα σε % Cu.....	14
Πίνακας 4.2 Στατιστικά μεγέθη για τον λογάριθμο της περιεκτικότητας σε % Cu.....	15
Πίνακας 5.1 Μέσο τετραγωνικό σφάλμα εκπαίδευσης (training) και ελέγχου (testing) και οι αντίστοιχοι συντελεστές R^2 για τα τρία νευρωνικά δίκτυα.....	17
Πίνακας 5.2: Συντεταγμένες των μπλοκ (τυπικό απόσπασμα από το μοντέλο).....	22
Πίνακας 6.1 Αποτελέσματα του αλγορίθμου Pseudoflow για την εύρεση των βέλτιστων ορίων εκσκαφής.....	32
Πίνακας 6.2 Εκτίμηση αποθεμάτων ανά κατηγορία περιεκτικότητας Cu %.....	33

1 Εισαγωγή

Η μεταλλευτική βιομηχανία είναι κρίσιμης σημασίας για την παγκόσμια οικονομία και η αποτελεσματική διαχείριση των αποθεμάτων ορυκτών είναι απαραίτητη για τη μεγιστοποίηση της κερδοφορίας και την ελαχιστοποίηση των περιβαλλοντικών επιπτώσεων. Παραδοσιακές μέθοδοι όπως το Kriging και το αντίστροφο τετράγωνο απόστασης (IDS) έχουν χρησιμοποιηθεί εδώ και χρόνια για την εκτίμηση των αποθεμάτων. Ωστόσο, αυτές οι μέθοδοι μπορεί να είναι χρονοβόρες, να παράγουν ανακριβή αποτελέσματα, λόγω των παραδοχών που απαιτούν επιπλέον απαιτούν εξειδικευμένη γνώση για την ορθή εφαρμογή τους.

Τα τελευταία χρόνια, τα τεχνητά νευρωνικά δίκτυα (TNN) έχουν εμφανιστεί ως μια πολλά υποσχόμενη εναλλακτική λύση για την εκτίμηση των αποθεμάτων κοιτασμάτων. Τα TNN έχουν τη δυνατότητα να παρέχουν μια πιο αποτελεσματική, γρήγορη και οικονομικά αποδοτική λύση.

Οι εφαρμογές των τεχνητών νευρωνικών δικτύων στον μεταλλευτικό τομέα, για την εκτίμηση αποθεμάτων και ποιότητας κοιτασμάτων είναι σχετικά περιορισμένες.

Οι Xiping και Yingxin (1993) χρησιμοποίησαν τεχνητά νευρωνικά δίκτυα ευθείας προώθησης πολλαπλών εσωτερικών επιπέδων για την εκτίμηση των περιεκτικοτήτων σε κοιτάσμα πορφυριτικού χαλκού. Οι Karageridis και Denby (1998) ασχολήθηκε με τη μοντελοποίηση περιεκτικότητας κοιτασμάτων. Μελέτησε τα δεδομένα γεωτρήσεων και στη συνέχεια χρησιμοποίησε νευρωνικά δίκτυα τύπου radial basis function, η οποία εφαρμόζεται στην απόσταση για τον υπολογισμό του βάρους κάθε νευρώνα, για τον υπολογισμό της περιεκτικότητας μεταλλευμάτων σε άγνωστες θέσεις. Ο Γαλετάκης (1999), μελέτησε τη χωρική μεταβολή του λόγου αποκάλυψης και της τέφρας επί ξηρού του λιγνίτη στην περιοχή του Λιγνιτικού Κέντρου Πτολεμαΐδας-Αμυνταίου χρησιμοποιώντας ένα ευθείας προώθησης (feedforward) νευρωνικό δίκτυο ενός κρυφού επιπέδου. Επίσης ο Γαλετάκης κ.ά (2000) χρησιμοποίησαν τα ίδιου τύπου τεχνητά νευρωνικά δίκτυα για την πρόβλεψη ποιοτικών παραμέτρων αδρανών υλικών. Ο Karageridis (2005) ερευνήσε το επίπεδο της εισόδου και τη σημαντικότητά του για την εκτίμηση των περιεκτικοτήτων ενός κοιτάσματος, χρησιμοποιώντας τεχνητά νευρωνικά δίκτυα τύπου radial basis function. Ο Tutmez (2009) με τη μέθοδο των τεχνητών νευρο-ασαφών δικτύων (Fuzzy-neural network), πραγματοποίησε εκτίμηση της περιεκτικότητας λιγνιτικού κοιτάσματος. Η σύγκριση των αποτελεσμάτων με άλλες συμβατικές μεθόδους έδειξε ότι αυτή η μέθοδος υπολογισμού είναι εξίσου αποτελεσματική. Οι Samanta & Bandopadhyay (2009) χρησιμοποίησαν τεχνητά νευρωνικά δίκτυα τύπου radial basis function, για τον υπολογισμό της περιεκτικότητας ενός κοιτάσματος χρυσού. Οι Li et al. (2010) μελέτησαν την εκτίμηση των περιεκτικοτήτων πορφυριτικού χαλκού με χρήση τεχνητών νευρωνικών δικτύων σε συνδυασμό με τη θεωρία των κυματιδίων (wavelet). Οι Mehdi et al. (2010) χρησιμοποίησαν τεχνητά νευρωνικά δίκτυα ευθείας προώθησης για την εκτίμηση των περιεκτικοτήτων ενός κοιτάσματος σιδήρου και σύγκριναν τα αποτελέσματά τους με τη μέθοδο Kriging. Οι Tahmasebi και Hezarkani (2012) χρησιμοποίησαν νευροασαφή δίκτυα (συνδυασμός νευρωνικών δικτύων με ασαφή λογική) με στόχο την εκτίμηση των χαρακτηριστικών ποιότητας των κοιτασμάτων. Οι Bahrami et al. (2016) χρησιμοποίησαν

τεχνητά νευρωνικά δίκτυα για την πρόβλεψη εισροής υπόγειων νερών σε υπαίθρια εκμετάλλευση. Οι Jafrasteh et al (2018) χρησιμοποίησαν τα νευρωνικά δίκτυα για την εκτίμηση της ποιότητας του χαλκού και μετά έγινε σύγκριση με τις γεωστατιστικές μεθόδους Ordinary Kriging και Indicator Kriging. Οι Kaplan & Topal (2020) χρησιμοποίησαν τα νευρωνικά δίκτυα μαζί με το κ πλησιέστερα δείγματα μοντέλο για ένα εκτιμήσουν την κατανομή περιεκτικότητας ενός κοιτάσματος.

Η διπλωματική αυτή εργασία διερευνά τη δυνατότητα χρησιμοποίησης των TNN για την εκτίμηση της περιεκτικότητας και των αποθεμάτων ενός κοιτάσματος χαλκού και στη συνέχεια τον καθορισμό των βέλτιστων ορίων εκσκαφής με την μέθοδο Pseudoflow. Η διάρθρωση της εργασίας, που αποτελείται από 7 κεφάλαια είναι όπως παρακάτω:

- Το κεφάλαιο 2 περιγράφει τις παραδοσιακές μεθόδους εκτίμησης αποθεμάτων – ποιότητας κοιτασμάτων και τις μεθόδους προσδιορισμού των βέλτιστων ορίων εκσκαφής.
- Το κεφάλαιο 3 αναφέρεται στην ανάπτυξη και λειτουργία των τεχνητών νευρωνικών δικτύων.
- Το κεφάλαιο 4 αναφέρεται στη γεωλογία του κοιτάσματος και στην επεξεργασία και στατιστική ανάλυση των στοιχείων των γεωτρήσεων.
- Στο κεφάλαιο 5 αναπτύχθηκε το μοντέλο του τεχνητού νευρωνικού δικτύου, εφαρμόστηκε στο ψηφιακό μοντέλο του κοιτάσματος και τέλος συγκρίθηκε με τις γεωστατιστικές μεθόδους Kriging και IDS.
- Στο κεφάλαιο 6 το ψηφιακό μοντέλο περιεκτικότητας μετασχηματίζεται σε οικονομικό μοντέλο και εφαρμόζεται ο αλγόριθμος Psudoflow για να βρεθούν τα βέλτιστα όρια εκσκαφής.
- Τέλος στο κεφάλαιο 7 δίνονται συμπεράσματα και προτάσεις για περαιτέρω έρευνα.

2 Υφιστάμενη μεθοδολογία εκτίμησης αποθεμάτων

2.1 Υφιστάμενες μέθοδοι υπολογισμού αποθεμάτων και ποιότητας

2.1.1 Γεωμετρικές μέθοδοι υπολογισμού αποθεμάτων

Τα γεωμετρικά και τα ποιοτικά χαρακτηριστικά ενός κοιτάσματος επηρεάζουν καθοριστικά τη σχεδίαση των επιφανειακών εκμεταλλεύσεων. Οι συνηθέστερες μέθοδοι εκτίμησης αποθεμάτων βασίζονται σε μαθηματικούς, στατιστικούς και γεωμετρικούς υπολογισμούς, χρησιμοποιώντας συνήθως την παραδοχή της εξάρτησης των γεωμετρικών και ποιοτικών παραμέτρων από την απόσταση ή την ύπαρξη κατανομών συγκεκριμένων μορφών. Κύριο πλεονέκτημα των γεωμετρικών μεθόδων είναι η απλότητα, η δυνατότητα εκτέλεσης των υπολογισμών με το χέρι και η δυνατότητα απεικόνισης των παραμέτρων του κοιτάσματος υπό μορφή χαρτών με τρόπο κατανοητό από τους ασχολούμενους με την εκμετάλλευση των ορυκτών υλών. Οι κύριες γεωμετρικές μέθοδοι είναι (*Γαλετάκης, 2012*):

- Μέθοδος των τομών (κατακόρυφες ή οριζόντιες).
- Μέθοδος των τριγώνων.
- Μέθοδος των πολυγώνων.
- Μέθοδος των ορθογωνίων μπλοκ.

Κύριο μειονέκτημα των παραπάνω μεθόδων αποτελεί η αδυναμία εκτίμησης του σφάλματος των υπολογισμών, η δυσκολία αλλαγής των παραμέτρων της εκμετάλλευσης που επιλέχθηκαν αρχικά και η αποδοχή συγκεκριμένου σχήματος και διακύμανσης των παραμέτρων του κοιτάσματος. Για τους λόγους αυτούς έχουν σήμερα αντικατασταθεί από άλλες μεθόδους με επικρατέστερες τις γεωστατικές (Kriging) αλλά και απλούστερες, όπως η μέθοδος των αντιστρόφων αποστάσεων με δημοφιλέστερη τη μέθοδο των αντιστρόφων αποστάσεων στο τετράγωνο (Inverse Distance Squared ή IDS).

2.1.2 Ανάπτυξη ψηφιακών μοντέλων του κοιτάσματος

Βασικό βήμα για τη σχεδίαση μιας εκμετάλλευσης με ηλεκτρονικό υπολογιστή αποτελεί η ανάπτυξη του ψηφιακού μοντέλου του κοιτάσματος. Ο υπολογισμός γίνεται με ειδικές μεθόδους παρεμβολής, χρησιμοποιώντας τα γνωστά στοιχεία από τις γεωτρήσεις και τις άλλες ερευνητικές εργασίες. Οι μέθοδοι αυτοί λαμβάνουν υπόψιν τις τιμές της υπολογιζόμενης παραμέτρου των γειτονικών στην κυψελίδα στοιχείων των γεωτρήσεων, τις αποστάσεις τους και τις σχετικές τους θέσεις ως προς αυτή. Οι πιο γνωστές είναι οι μέθοδοι των αντιστρόφων αποστάσεων, των κινητών μέσων, οι στατιστικές, οι γεωστατιστικές και οι πλέον πρόσφατες που βασίζονται στα νευρωνικά δίκτυα και στους γενετικούς αλγόριθμους. Τα βασικά βήματα στη δημιουργία ενός ψηφιακού μοντέλου κοιτάσματος είναι:

- Υπολογισμός των σύνθετων δειγμάτων από τα δείγματα πυρήνων των γεωτρήσεων.

- Καθορισμός της θέσης και των στοιχειωδών διαστάσεων του μοντέλου.
- Υπολογισμός του ψηφιακού μοντέλου επιφανείας για την περιοχή που καταλαμβάνει το μοντέλο του κοιτάσματος.
- Απόδοση τιμής της παραμέτρου του κοιτάσματος που εξετάζεται (π.χ. περιεκτικότητα) σε κάθε ένα στοιχειώδη όγκο (Γαλετάκης, 2012).

2.1.3 Μέθοδος αντιστρόφων αποστάσεων (Inverse Distance)

Η μέθοδος των αντιστρόφων αποστάσεων είναι μια μέθοδος σταθμικού μέσου, όπου για τον υπολογισμό της τιμής ενός στοιχειώδους όγκου (block) του κοιτάσματος χρησιμοποιούνται τα πλησιέστερα δείγματα που περιέχονται εντός μιας προκαθορισμένης περιοχής που ορίζεται από μια σφαίρα. Ο συντελεστής στάθμισης για κάθε δείγμα που λαμβάνεται υπόψη στον υπολογισμό, είναι το αντίστροφο της απόστασης του δείγματος από το κέντρο του στοιχειώδους όγκου υψωμένου σε μια δύναμη m . Ο υπολογισμός με τη μέθοδο αυτή, που είναι εμπειρική, βασίζεται στην αρχή ότι όσο πλησιέστερα στο μπλοκ είναι ένα δείγμα τόσο περισσότερο πρέπει να αυξάνεται ο συντελεστής επιρροής του στη διαμόρφωση του σταθμικού μέσου του μπλοκ. Η πιο συνηθισμένη παραλλαγή της μεθόδου των αντιστρόφων αποστάσεων είναι η μέθοδος του τετραγώνου των αντιστρόφων αποστάσεων, γνωστή και ως IDS (Inverse Distance Squared method). Για τον υπολογισμό της περιεκτικότητας g_{block} του κέντρου ενός μπλοκ χρησιμοποιώντας τα k πλησιέστερα δείγματα, χρησιμοποιείται η εξ' (221):

$$g_{block} = \frac{\sum_{i=1}^k \left(\frac{1}{d_i}\right)^2 \cdot g_i}{\sum_{i=1}^k \left(\frac{1}{d_i}\right)^2} \quad (2.1)$$

όπου d_i η απόσταση του δείγματος i από το κέντρο του μπλοκ και g_i η περιεκτικότητα του δείγματος i (Γαλετάκης, 2012).

2.1.4 Μέθοδος Kriging

Η μέθοδος Kriging αποτελεί τη βέλτιστη γραμμική αμερόληπτη μεσοσταθμική μέθοδο εκτίμησης. Στα πλεονεκτήματά της, πλην του θεωρητικού υπόβαθρου της γεωστατιστικής στο οποίο στηρίζεται, σε αντίθεση με την εμπειρική μέθοδο IDS, είναι η δυνατότητα που παρέχει να εκτιμηθεί εκτός της μέσης τιμής ενός μπλοκ και η διακύμανσή του. Εξασφαλίζεται δηλαδή μία ένδειξη κατά πόσο η εκτιμώμενη μέση τιμή ενός μπλοκ είναι αξιόπιστη. Η εφαρμογή των μεθόδων Kriging βασίζεται στις επόμενες παραδοχές (Χριστόπουλος, 2014):

- Θεωρείται ότι το τυχαίο πεδίο είναι δυνατό να αναλυθεί σε δύο συνιστώσες, η πρώτη συνιστώσα εκφράζει την τάση, δηλαδή τη μεταβολή της μέσης τιμής και η δεύτερη την διακύμανση του πεδίου εκατέρωθεν της τάσης. Ως «τυχαίο πεδίο» μπορεί να θεωρηθεί ένα σύνολο τυχαίων μεταβλητών που περιγράφουν τη χωροχρονική μεταβολή του σχετικού φυσικού μεγέθους όπως οι τιμές συγκεντρώσεων. Σε αντίθεση με τις συναρτήσεις οι οποίες έχουν μια συγκεκριμένη μαθηματική έκφραση, ένα τυχαίο πεδίο που δεν έχει σαφή

μαθηματική έκφραση αντιπροσωπεύει ένα σύνολο δυνατών καταστάσεων. Ένα τυχαίο πεδίο μπορεί να θεωρηθεί ως μία πολυδιάστατη τυχαία μεταβλητή, λόγω της αλληλεξάρτησης των φυσικών μεγεθών.

- Μια εκτίμηση που βασίζεται στην ελαχιστοποίηση της διασποράς του σφάλματος είναι ακριβής αν η συνάρτηση κατανομής πιθανότητας του πεδίου είναι συμμετρική. Συνήθως θεωρείται ότι η διακύμανση περιγράφεται από ένα στατιστικά ομοιογενές πεδίο, έτσι ώστε η συνάρτηση συνδιασποράς και το βαριόγραμμα να εξαρτώνται αποκλειστικά από την απόσταση μεταξύ δυο σημείων. Η συνθήκη ομοιογένειας δεν είναι απόλυτα απαραίτητη και στην περίπτωση της μεθόδου του καθολικού Kriging περιορίζεται μόνο στην εξάρτηση του βαριογράμματος.

Χρησιμοποιούνται ευρέως έξι διαφορετικές εφαρμογές της μεθόδου Kriging ([Χριστόπουλος, 2014](#)):

- Απλό Kriging (simple Kriging): εφαρμόζεται υπό την προϋπόθεση ότι η μέση τιμή των δειγμάτων είναι γνωστή σε όλη την περιοχή.
- Κανονικό Kriging (ordinary Kriging): εφαρμόζεται όταν η μέση τιμή του πεδίου θεωρείται σταθερή στην περιοχή εκτίμησης, αλλά η τιμή της μέσης τιμής μπορεί να μεταβάλλεται από γειτονιά σε γειτονιά.
- Καθολικό Kriging (universal Kriging): εφαρμόζεται όταν τα δεδομένα παρουσιάζουν τοπική τάση. Η μέση τιμή του πεδίου μεταβάλλεται εντός της γειτονιάς του σημείου εκτίμησης. Σε αυτή την περίπτωση χρησιμοποιείται σαν πρότυπο τάσης ένας γραμμικός συνδυασμός γνωστών συναρτήσεων.
- Kriging σε πεδίο δεικτών (indicator Kriging): εφαρμόζεται όταν η συνάρτηση κατανομής της πιθανότητας είναι ασύμμετρη και παρουσιάζει σημαντική πυκνότητα στις υψηλές τιμές. Σε αυτές τις περιπτώσεις δεν είναι δυνατό να χρησιμοποιηθεί το κανονικό πρότυπο κατανομής πιθανότητας. Η επανάληψη της εκτίμησης για πολλαπλές τιμές του ορίου (multiple indicator Kriging) επιτρέπει την εκτίμηση της αθροιστικής κατανομής πιθανότητας.
- Συνδυαστικό Kriging (co-Kriging): χρησιμοποιείται προκειμένου να συμπεριληφθούν στη διαδικασία εκτίμησης ιδιότητες (π.χ. περιεκτικότητες) που είναι συσχετισμένες με τις ιδιότητες που επιδιώκεται να εκτιμηθούν.
- Kriging μετασχηματισμένων μεταβλητών (lognormal Kriging) : υποθέτει ότι τα δεδομένα ακολουθούν την λογαριθμοκανονική κατανομή.

Πριν από την εφαρμογή της μεθόδου kriging είναι απαραίτητη η στατιστική ανάλυση των στοιχείων των γεωτρήσεων, για να υπολογιστούν οι βασικές στατιστικές παράμετροι (μέση τιμή, τυπική απόκλιση, διάμεσος, επικρατούσα τιμή, μέτρο ασυμμετρίας και κύρτωσης) και να διαπιστωθεί ποια είναι η θεωρητική κατανομή των στοιχείων των γεωτρήσεων. Το ημιβαριόγραμμα είναι το βασικό γεωστατιστικό εργαλείο που χρησιμοποιείται για να εκτιμηθεί η μεταβολή μιας χωρομεταβλητής $Z(x_i)$. Οι παράμετροι του ημιβαριογράμματος καθορίζουν την χωρική εξάρτηση των τιμών του πεδίου σε δύο γειτονικά σημεία. Για τον ευκολότερο υπολογισμό

του ημιβαριογράμματος είναι προτιμότερο να ομαδοποιηθούν τα δείγματα και να δημιουργηθούν τα σύνθετα δείγματα (composites) σύμφωνα με τη γεωμετρία της βαθμίδας και στη συνέχεια να γίνει ο υπολογισμός του $\gamma(h)$. Το πειραματικό ημιβαριόγραμμα ορίζεται βάσει της [εξ. \(2.2\)](#):

$$\gamma(h) = \frac{1}{2 \cdot n(h)} \sum_{i=1}^{n(h)} [Z(x_i) - Z(x_i + h)]^2 \quad (2.2)$$

όπου $\gamma(h)$ η τιμή του ημιβαριογράμματος, $n(h)$ ο αριθμός των δειγμάτων που απέχουν απόσταση h , $Z(x_i)$ τιμή της μεταβλητής που εξετάζεται στη θέση x_i και $Z(x_i + h)$ η τιμή της μεταβλητής που εξετάζεται στη θέση $x_i + h$.

2.2 Μέθοδοι καθορισμού των βέλτιστων ορίων εκσκαφής υπαίθριας εκμετάλλευσης

2.2.1 Μέθοδος κινητού κώνου

Η μέθοδος του κινητού κώνου αποτελεί τη δημοφιλέστερη μέθοδο εύρεσης των τελικών ορίων μιας υπαίθριας εκσκαφής με βάση οικονομικά κριτήρια. Η αρχή της μεθόδου βασίζεται στη συστηματική μετακίνηση ενός ανεστραμμένου κώνου με γωνία ίση με την τελική γωνία πρανούς. Πιο συγκεκριμένα, ο ανεστραμμένος κώνος τοποθετείται σε κάθε στοιχειώδη όγκο μεταλλεύματος του τρισδιάστατου ψηφιακού μοντέλου και υπολογίζεται σε κάθε θέση ποιο είναι το οικονομικό αποτέλεσμα της εξόρυξης του υπόψη μπλοκ. Ο κώνος στη συνέχεια μετακινείται στο επόμενο μπλοκ έως ότου εξαντλήσει όλα τα μπλοκ του μοντέλου. Σε κάθε μια θέση υπολογίζεται το οικονομικό αποτέλεσμα και ο κώνος κινείται μέχρι την θέση όπου το οικονομικό αποτέλεσμα είναι οριακό (έσοδα = έξοδα). Με τη μέθοδο του κινητού κώνου επίσης υπολογίζεται ένα δεύτερο όριο εκμετάλλευσης, το βέλτιστο όριο εκμετάλλευσης, που περικλείεται από το οριακό, που δίνει ποιο μέγεθος και σχήμα μεγιστοποιεί το οικονομικό όφελος με τις παρούσες τιμές του μεταλλεύματος και με τα παρόντα κόστη. Το οικονομικό μέγεθος που συνήθως μεγιστοποιείται είναι η καθαρά παρούσα αξία (Net Present Value) ([Γαλετάκης, 2012](#)).

2.2.2 Αλγόριθμος Lerchs-Grossman

Ο αλγόριθμος Lerchs-Grossman ([1965](#)), που βασίζεται στη θεωρία γραφημάτων, είναι η μοναδική μαθηματική μέθοδος που αποδεδειγμένα υπολογίζει την βέλτιστη εκσκαφή. Για την εύρεση της βέλτιστης εκσκαφής ο αλγόριθμος χρησιμοποιεί τις οικονομικές αξίες των μπλοκ (block values) και τις διασυνδέσεις τους (structure arcs). Ο αλγόριθμος ξεκινάει την σάρωση των μπλοκ από το κατώτερο επίπεδο και εντοπίζει εκείνα με θετικές οικονομικές αξίες. Κάθε μπλοκ κατά τη διάρκεια σάρωσης μπορεί να χαρακτηριστεί ως εξορύξιμο ή όχι. Τα εξορυσσόμενα μπλοκ συνδέονται και δημιουργούν κλάδους και στο τέλος εντοπίζεται εκείνη η εκσκαφή με τη μεγαλύτερη οικονομική αξία ([Γαλετάκης, 2012](#)).

3 Τεχνητά νευρωνικά δίκτυα

3.1 Χαρακτηριστικά και δομή τεχνητών νευρωνικών δικτύων

Τα τεχνητά νευρωνικά δίκτυα είναι ίσως μια από τις πλέον χαρακτηριστικές εφαρμογές βιομηχανικής, μια που η δομή και η λειτουργία τους είναι εμπνευσμένη από τις αντίστοιχες λειτουργίες του ανθρώπινου εγκεφάλου. Τα νευρωνικά δίκτυα είναι μη γραμμικά συστήματα τα οποία αποτελούνται από ένα σύνολο νευρώνων οι οποίοι είναι συνδεδεμένοι μεταξύ τους με τις συνάψεις. Έχουν την ικανότητα να μαθαίνουν από την υπάρχουσα εμπειρία-γνώση, να γενικεύουν την αποκτηθείσα γνώση, να εξάγουν χρήσιμες πληροφορίες από δεδομένα με θόρυβο, να αναπτύσσουν λύσεις σε σύντομο χρονικό διάστημα και έχουν μεγάλη προσαρμοστικότητα. Σήμερα χρησιμοποιούνται σε αρκετές τεχνολογικές εφαρμογές με κυριότερη την αναγνώριση προτύπων (pattern recognition).

Ένα νευρωνικό δίκτυο αποτελείται από δύο μέρη: τους νευρώνες και τις συνδέσεις τους (συνάψεις). Ο νευρώνας λαμβάνει ένα ή περισσότερα σήματα (είσοδος), διεγείρεται (ενεργοποίηση) και αποστέλλει το αποτέλεσμα (έξοδος) στους άλλους νευρώνες, με τους οποίους είναι συνδεδεμένος με τις συνάψεις. Κάθε σύναψη χαρακτηρίζεται με ένα συντελεστή w_i , που δείχνει πόσο ισχυρά συνδεδεμένοι είναι οι δύο νευρώνες. Το πρώτο επίπεδο (Input layer) παίρνει τις τιμές X_i , ακολουθεί το κρυφό επίπεδο το οποίο μπορεί να είναι παραπάνω από ένα και έχει όσους νευρώνες χρειάζονται για να λυθεί το πρόβλημα και τέλος το επίπεδο εξόδου (output layer) το οποίο παίρνει τις τιμές y_i . Κάθε X_i ενώνεται με κάθε νευρώνα και έχει ένα βάρος w_i και η τιμή του νευρώνα υπολογίζεται από την [εξ. \(3.1\)](#):

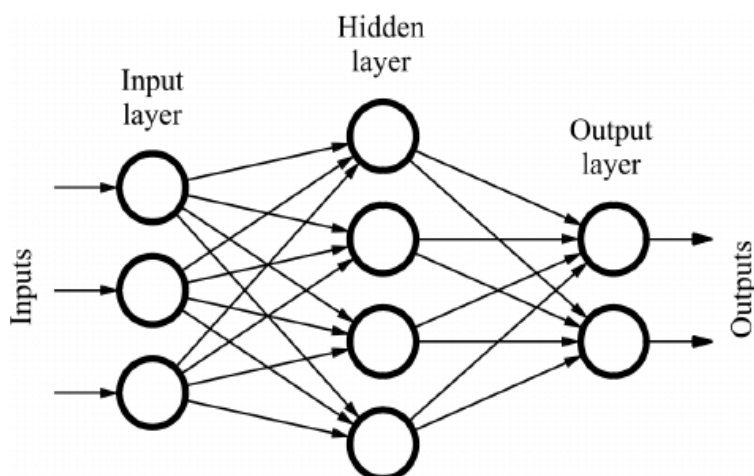
$$I = \sum_{i=1}^n (X_i \cdot w_i) \quad (3.1)$$

και μετά χρησιμοποιεί μια συνάρτηση, η οποία λέγεται συνάρτηση ενεργοποίησης (activation function). Μερικές από τις πιο διαδεδομένες συναρτήσεις ενεργοποίησης είναι:

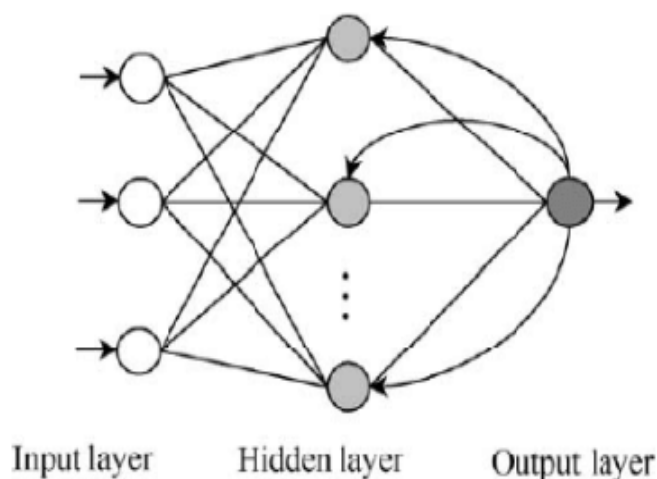
- Γραμμική (Linear).
- Σιγμοειδής (Sigmoid).
- Διορθωμένη γραμμική μονάδα (Relu).
- Γραμμική μονάδα σφάλματος Gauss (Gelu).

Τα ΤΝΔ με βάση την φορά εξόδων ενός νευρώνα χωρίζονται σε νευρωνικά δίκτυα ευθείας προώθησης (feedforward) ([Σχήμα 3.1](#)) και αναδρομικού τύπου (feedback) ([Σχήμα 3.2](#)).

Το feedforward νευρωνικό δίκτυο ήταν ο πρώτος και απλούστερος τύπος τεχνητού νευρωνικού δικτύου που επινοήθηκε. Σε αυτό το δίκτυο, οι πληροφορίες κινούνται προς μία μόνο κατεύθυνση, προς τα εμπρός από τους κόμβους εισόδου, μέσω των κρυφών κόμβων και προς τους κόμβους εξόδου. Δεν υπάρχουν κύκλοι ή βρόχοι στο δίκτυο. Στο feedback νευρωνικό δίκτυο δεν τίθεται περιορισμός στην κατεύθυνση, αφού κάθε επίπεδο μπορεί να τροφοδοτεί το ίδιο ή το προηγούμενο επίπεδο ([Γαλετάκης κ.ά., 2000](#)).



Σχήμα 3-1 Δομή δικτύου ευθείας προώθησης (feedforward) (Fadja et al 2018).



Σχήμα 3-2 Δομή δικτύου αναδρομικού τύπου (feedback) (İslamoğlu 2016).

3.2 Μέθοδοι εκπαίδευσης τεχνητών νευρωνικών δικτύων

Μια βασική ιδιότητα των ΤΝΔ είναι η εκπαίδευση, η οποία επιτυγχάνεται μέσω την αλλαγής των τιμών των βαρών για καλύτερη πληροφορία στην έξοδο. Η εκπαίδευση χωρίζεται σε δύο κατηγορίες, την μάθηση με επίβλεψη και την μάθηση χωρίς επίβλεψη.

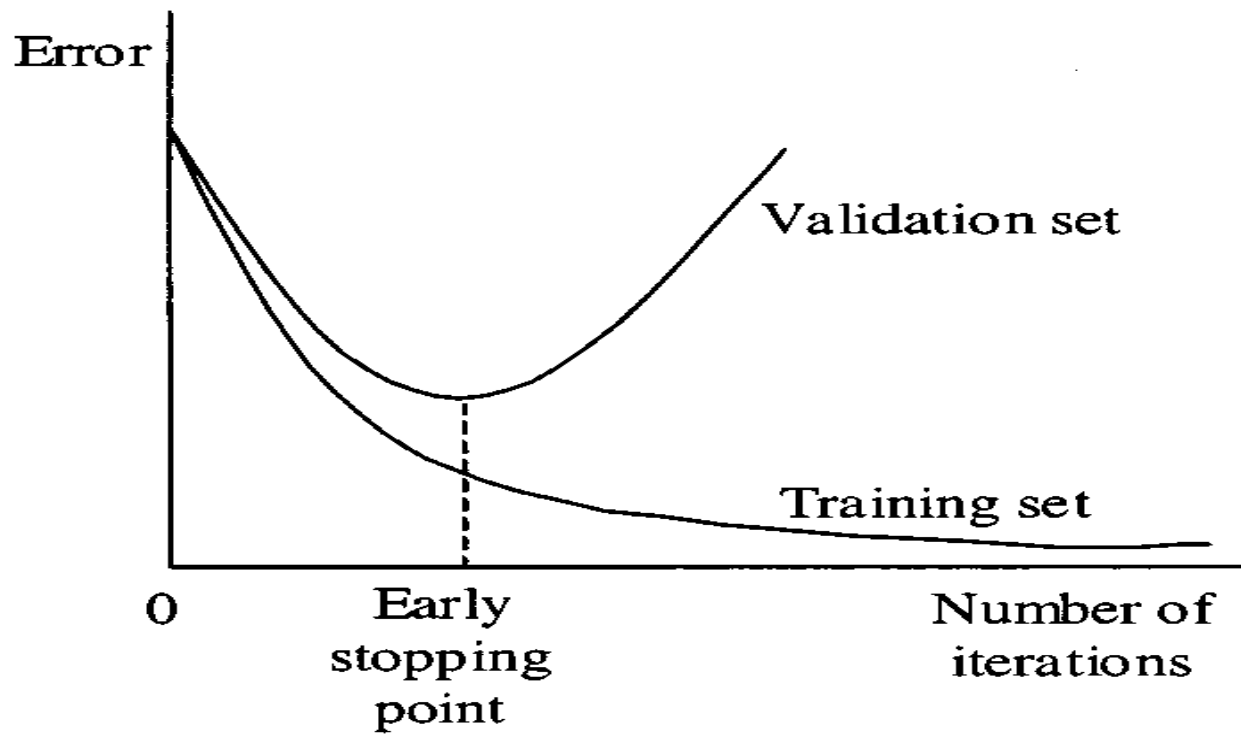
Μάθηση με επίβλεψη: Αυτή η μάθηση χρειάζεται έναν εξωτερικό παρατηρητή. Ο πιο συχνός τύπος είναι αυτός που η πληροφορία του σφάλματος που χρησιμοποιείται για την αλλαγή των βαρών στέλνεται πίσω, αντίθετα δηλαδή από τη διεύθυνση διάδοσης της διέγερσης (back-propagation algorithm) (Rumelhart et al., 1986). Γίνεται προσπάθεια ελαχιστοποίησης της τιμής που δίνει το δίκτυο με την τιμή που μπήκε στην είσοδο, κάτω από ένα σφάλμα.

Μάθηση χωρίς επίβλεψη: Στην μάθηση χωρίς επίβλεψη απλώς δίνεται η πληροφορία στο δίκτυο, αλλά δεν δίνεται με τους αντίστοιχους στόχους όπως προηγουμένως και έτσι δεν γίνεται κανένας έλεγχος ή σύγκριση για την πορεία του σφάλματος. Το δίκτυο δεν χρησιμοποιεί κάποια εξωτερική παράμετρο για την αλλαγή των βαρών. Υπάρχει συγκεκριμένη διαδικασία που ακολουθείται και καταλήγει σε εκπαίδευση του δικτύου. Το δίκτυο χρησιμοποιεί έναν εσωτερικό έλεγχο, ψάχνει να βρει κάποιες τάσεις ή κανονικότητα στα σήματα εισόδου και προσπαθεί ώστε οι έξοδοι να έχουν τα ίδια χαρακτηριστικά όπως και οι εισοδοί. Έτσι προκύπτει η αυτο-εποπτευόμενη εκπαίδευση, διότι το δίκτυο ελέγχει τον εαυτό του και διορθώνει τα σφάλματα στα δεδομένα με ένα μηχανισμό ανάδρασης(feedback). Ο τρόπος αυτός δεν συναντάται τόσο συχνά όπως η εποπτευόμενη εκπαίδευση (*Αργυράκης, 2001*).

Κατά τη διάρκεια της εκπαίδευσης υπάρχει ο κίνδυνος το νευρωνικό δίκτυο να υπερεκπαιδευτεί (overtraining) ή να υποεκπαιδευτεί (undertraining). Η ικανότητα της γενίκευσης του νευρωνικού δικτύου επηρεάζεται αρνητικά κατά την υπερεκπαίδευση. Εάν ένα νευρωνικό δίκτυο παρουσιάζει μεγαλύτερη πολυπλοκότητα από την επιθυμητή, έχει ως αποτέλεσμα να μαθαίνει την λεπτομέρεια και τον θόρυβο των δειγμάτων. Επομένως, το νευρωνικό δίκτυο δεν είναι σε θέση να επιτύχει τους στόχους της εκπαίδευσης και την γενίκευση σε άγνωστα δείγματα. Για την αποφυγή της υπερεκπαίδευσης και για να γίνει έλεγχος της γενίκευσης ενός δικτύου γίνεται εφαρμογή της διαδικασίας της πρόωρης διακοπής (early stopping) (*Morgan and Bourland, 1989*).

Τα δείγματα τα οποία θα χρησιμοποιηθούν για την εκπαίδευση του νευρωνικού δικτύου χωρίζονται σε δεδομένα εκπαίδευσης (training data), σε δεδομένα επικύρωσης (validation data) και σε δεδομένα δοκιμών (testing data). Τα training data χρησιμοποιούνται για την εκπαίδευση του δικτύου για να βρει τα σωστά βάρη, τα validation data χρησιμοποιούνται για την αξιολόγηση του δικτύου σε καινούργια δεδομένα και τα testing data για τον έλεγχο της γενίκευσης. Ο σκοπός της εκπαίδευσης είναι να μειωθεί το σφάλμα χωρίς να γίνει υπερεκπαίδευση.

Όπως φαίνεται στο *Σχήμα 3.3* κατά τη διαδικασία της εκπαίδευσης το σφάλμα των εκτιμήσεων μειώνεται σταδιακά με την πάροδο των κύκλων εκπαίδευσης (epochs) για το σύνολο των δεδομένων της εκπαίδευσης. Αντίθετα για το σύνολο των δεδομένων ελέγχου το σφάλμα μειώνεται έως μια ελάχιστη τιμή και στη συνέχεια αυξάνεται. Στο σημείο αυτό που επιτυγχάνεται το ελάχιστο σφάλμα για το σύνολο των δεδομένων ελέγχου, η εκπαίδευση του νευρωνικού δικτύου σταματά για αποφυγή υπερεκπαίδευσης.



Σχήμα 3-3 Απεικόνιση της τεχνικής έγκαιρης διακοπής της εκπαίδευσης ([Ramazan, 2001](#)).

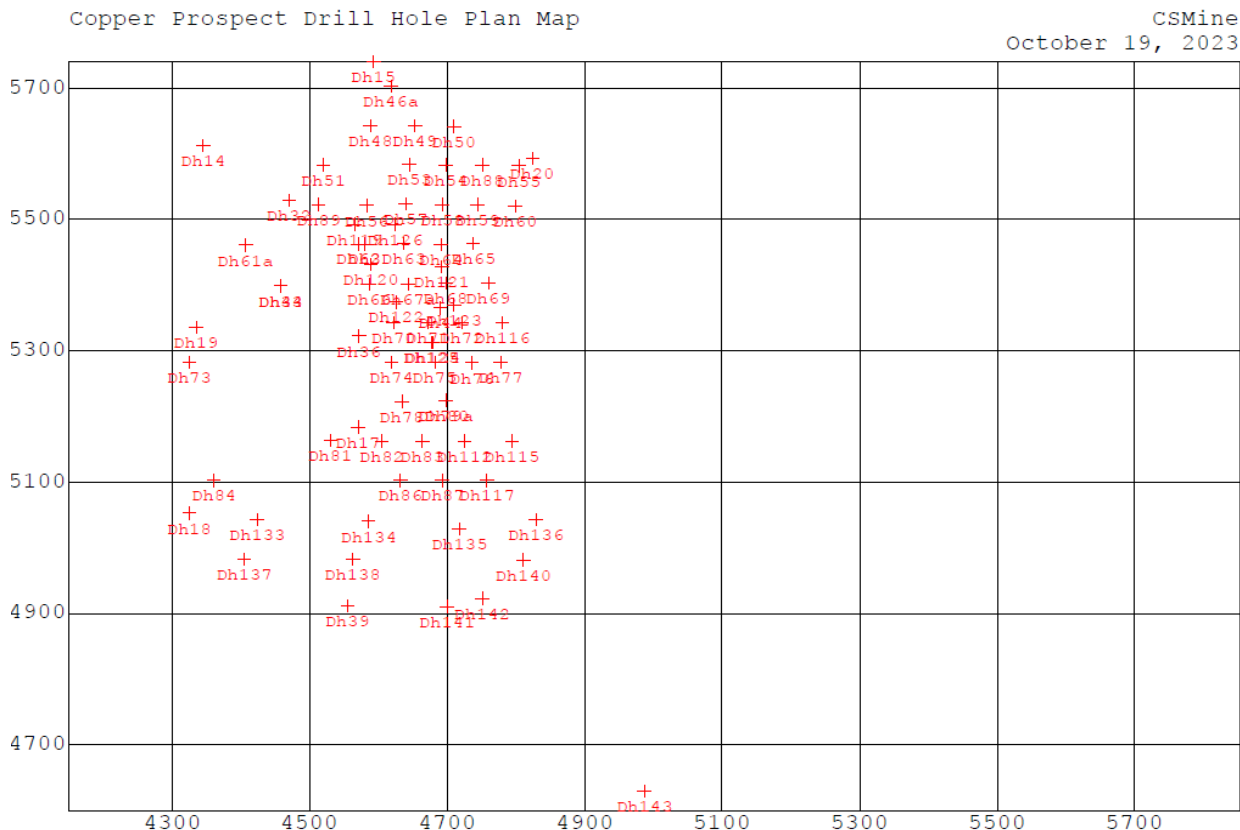
4 Επεξεργασία-αξιολόγηση δεδομένων

4.1 Περιγραφή του κοιτάσματος χαλκού

Τα δεδομένα που χρησιμοποιούνται στην εργασία αυτή έχουν ληφθεί από ένα κοιτάσμα χαλκού/μολυβδαινίου στην περιοχή Andina της Χιλής. Το τμήμα Andina της εταιρείας Codelco εξορύσσει το κοιτάσμα χαλκού/μολυβδαινίου Rio Blanco που βρίσκεται 80 χιλιόμετρα βορειοανατολικά του Σαντιάγο στην Χιλή. Η πλησιέστερη μεγάλη πόλη είναι το Los Andes City, που απέχει περίπου 40 χιλιόμετρα βορειοδυτικά. Αν και η ύπαρξη του κοιτάσματος ήταν γνωστή από το 1920, η εξόρυξη δεν έγινε προχώρησε μέχρι το 1970. Επί του παρόντος, το τμήμα Andina εκμεταλλεύεται δύο ορυχεία: Rio blanco, μια υπόγεια εκμετάλλευση και Sur Sur, μια υπαίθρια εκμετάλλευση. Μαζί παράγουν 220000 τόνους συμπύκνωμα μεταλλεύματος χαλκού και 1900 τόνους μολυβδαίνιο ετησίως. Στο ορυχείο Sur Sur, το οποίο άρχισε να λειτουργεί το 1981, εξορύσσονται 7 εκατομμύρια τόνοι μεταλλεύματος χαλκού/μολυβδαινίου και 15 εκατομμύρια τόνοι στείρων. Δεδομένου ότι το υψόμετρο της επιφάνειας κυμαίνεται από 3500 έως 4200m πάνω από την επιφάνεια της θάλασσας, η εκμετάλλευση είναι αρκετά δύσκολη. Λόγω των κλιματικών περιορισμών, η υπαίθρια εκμετάλλευση μπορεί να λειτουργήσει μόνο 320 ημέρες κάθε χρόνο. Αυτό απαιτεί την αποθήκευση του μεταλλεύματος σε μια παλιά υπόγεια εκμετάλλευση, που τροφοδοτούν ένα υπόγειο εργοστάσιο στον πυθμένα του.

Από οικονομικής άποψης η πιο σημαντικός σχηματισμός είναι εκείνος που περιέχει το ορυκτό τουρμαλίνη (BXT) και ο οποίος περιλαμβάνει το 90% της μεταλλοφορίας και αποτελεί το κύριο μέρος της. Η εξάπλωση τους είναι από βορρά-νότο, ενώ οι άλλοι τύποι πετρωμάτων, γρανοδιορίτης Κασκάντα (GDCC), κροκαλοπαγές Monolito (BXMN), κροκαλοπαγές με τουρμαλίνη (BTBXMN), κροκαλοπαγές βάσης (BXT0) και κροκαλοπαγές με τουρμαλίνη (BYTTO), βρίσκονται όλοι στα δυτικά του σχηματισμού BXT. Οι σχηματισμοί αυτοί έχουν χαμηλή περιεκτικότητα ή είναι στείρα (*Hustrulid and Kuchta, 2006*).

Το κοιτάσμα εξερευνήθηκε με 76 κατακόρυφες γεωτρήσεις δειγματοληψίας με διαφορετικό βάθος, οι θέσεις των οποίων δίνονται στο [Σχήμα 4.1](#).



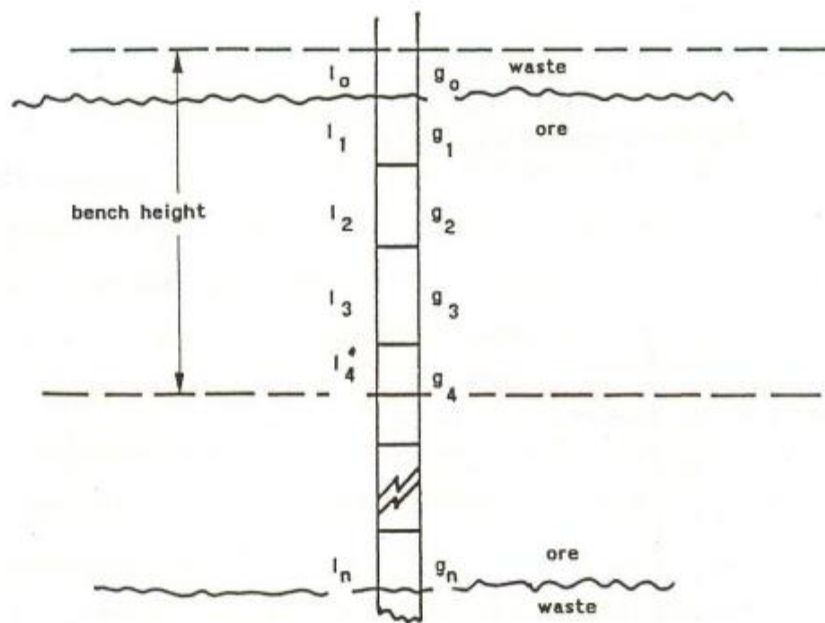
Σχήμα 4.1 Χάρτης θέσεων γεωτρήσεων κοιτάσματος (διαστάσεις κατά τους άξονες X και Y σε ft) (Hustrulid and Kuchta, 2006).

4.2 Πρωτογενή στοιχεία γεωτρήσεων και δημιουργία σύνθετων δειγμάτων

Χρησιμοποιούνται τα αναλυτικά στοιχεία των γεωτρήσεων και με την ενοποίησή τους σχηματίζονται τα σύνθετα δείγματα. Βασική παράμετρος για τον υπολογισμό των σύνθετων δειγμάτων είναι το ύψος της βαθμίδας εκμετάλλευσης που θα επιλεγεί. Ο υπολογισμός των σύνθετων δειγμάτων είναι απαραίτητος έτσι ώστε να δημιουργηθεί το μοντέλο του κοιτάσματος. Η δημιουργία τους έγινε με τη χρήση του λογισμικού CSMine (Hustrulid and Kuchta, 2006). Τα σύνθετα δείγματα υπολογίζονται από το υψόμετρο κάθε γεώτρησης, με αριθμό (number of intervals) και ύψος βαθμίδων (compositing interval) 20 και 25 ft αντίστοιχα. Τα σύνθετα δείγματα που προκύπτουν με βάση αυτές τις παραδοχές είναι 858 από τις αρχικές 76 γεωτρήσεις.

Για τον υπολογισμό του σύνθετου δείγματος λαμβάνονται υπόψιν όλα τα δείγματα πυρήνων της γεώτρησης που εμπίπτουν εντός του ύψους της βαθμίδας και έχουν χαρακτηριστεί ως μετάλλευμα (ore) και όχι τα άγονα υλικά (waste). Έχουν καταγραφεί από τους πυρήνες των γεωτρήσεων τα n δείγματα με μήκη και περιεκτικότητες l_1, l_2, \dots, l_n και g_1, g_2, \dots, g_n αντίστοιχα. Η μέση περιεκτικότητα ενός σύνθετου δείγματος με μήκος ίσο με το ύψος μιας βαθμίδας, όπως φαίνεται στο Σχήμα 4.2, υπολογίζεται από την $\epsilon\zeta$ (4.1):

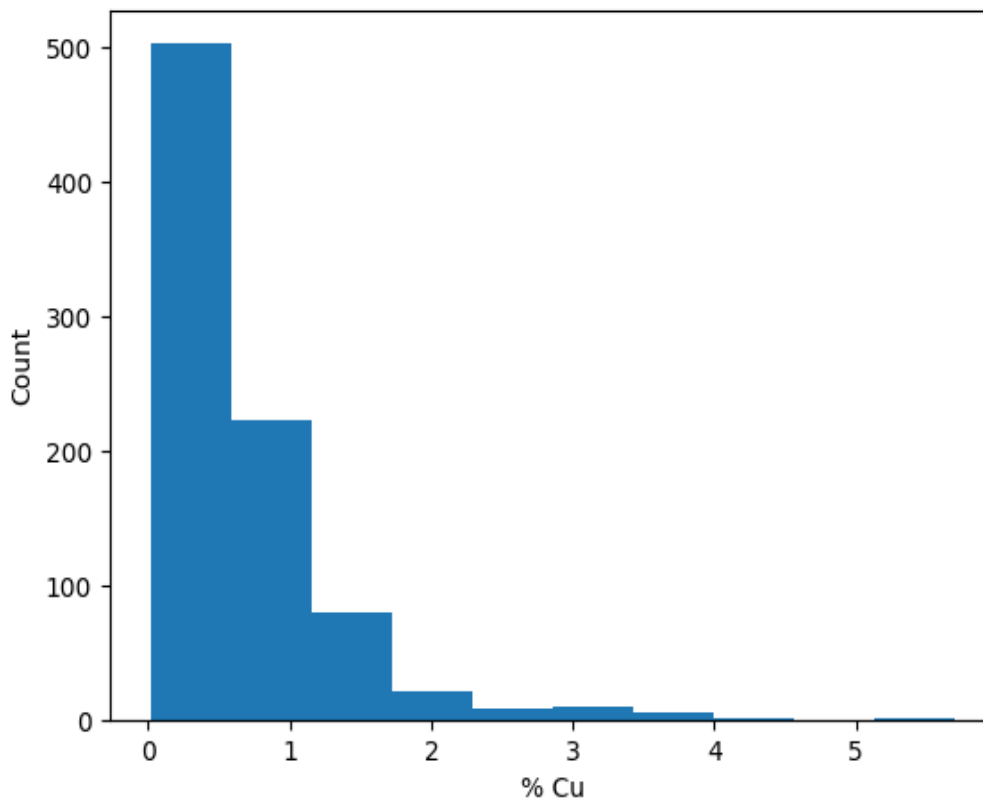
$$g_{\text{βαθμίδας}} = \frac{\sum_{i=1}^n g_i \cdot l_i}{\sum_{i=1}^n l_i} \quad (4.1)$$



Σχήμα 4.2 Δημιουργία σύνθετου δείγματος βαθμίδας με βάση τους πυρήνες των γεωτρήσεων (Hustrulid and Kuchta, 2006).

4.3 Στατιστική ανάλυση και υπολογισμός ημιβαριογράμματος για τα σύνθετα δείγματα

Για την στατιστική ανάλυση των σύνθετων δειγμάτων αναπτύχθηκε κώδικας σε γλώσσα προγραμματισμού python και των βιβλιοθηκών της όπως φαίνεται στο [Παράρτημα Ι](#). Τα αποτελέσματα έδειξαν πως το μεγαλύτερο πλήθος των δειγμάτων αντιστοιχεί σε πολύ μικρές τιμές περιεκτικότητας ([Σχήμα 4.3](#)), ενώ η προκύπτουσα κατανομή προσομοιάζει με τη λογαριθμική.



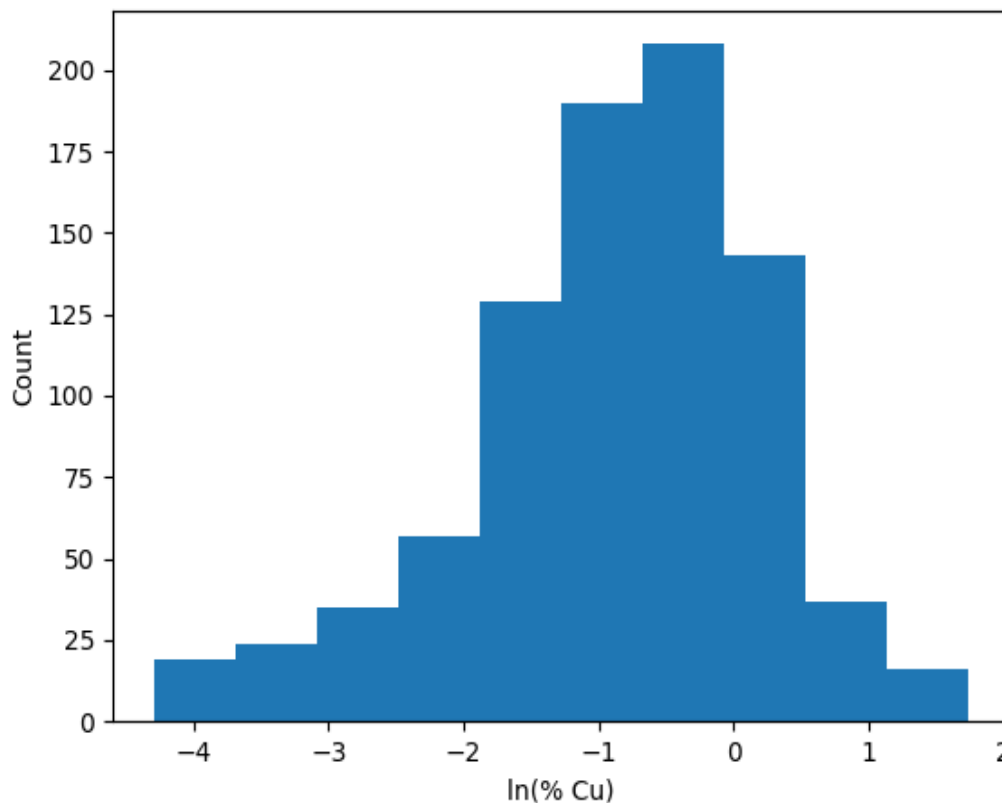
Σχήμα 4.3 Ιστόγραμμα της περιεκτικότητας σε % Cu για τα σύνθετα δείγματα.

Στον ακόλουθο [πίνακα 4.1](#) φαίνονται και τα στατιστικά μεγέθη για την περιεκτικότητα σε % Cu.

Πίνακας 4.1 Στατιστικά μεγέθη για την περιεκτικότητα σε % Cu.

Στατιστικά μεγέθη	Τιμή
Αριθμός δειγμάτων (Number of Samples)	858
Αριθμητικός μέσος όρος (Arithmetic Average)	0.667
Διακύμανση (Variance)	0.489
Τυπική απόκλιση (Standard Deviation)	0.700
Συντελεστής ασυμμετρίας (Coefficient of Skewness)	2.765
Συντελεστής κύρτωσης (Coefficient of Kurtosis)	11.012
Γεωμετρικός μέσος (Geometric Mean)	0.409
Διάμεσος (Median)	0.458
Τροποποιημένος μέσος (10%) (10% Trim Range)	0.546

Στη συνέχεια έγινε μετασχηματισμός των αρχικών τιμών χρησιμοποιώντας το φυσικό λογάριθμο για να γίνει πιο εύκολη και γρήγορη η εκπαίδευση του νευρωνικού δικτύου και να έρθουν τα δεδομένα πιο κοντά στην κανονική κατανομή. Στο [Σχήμα 4.4](#) παρατηρείται ότι η προκύπτουσα κατανομή είναι αρκετά πιο κοντά στην κανονική κατανομή.



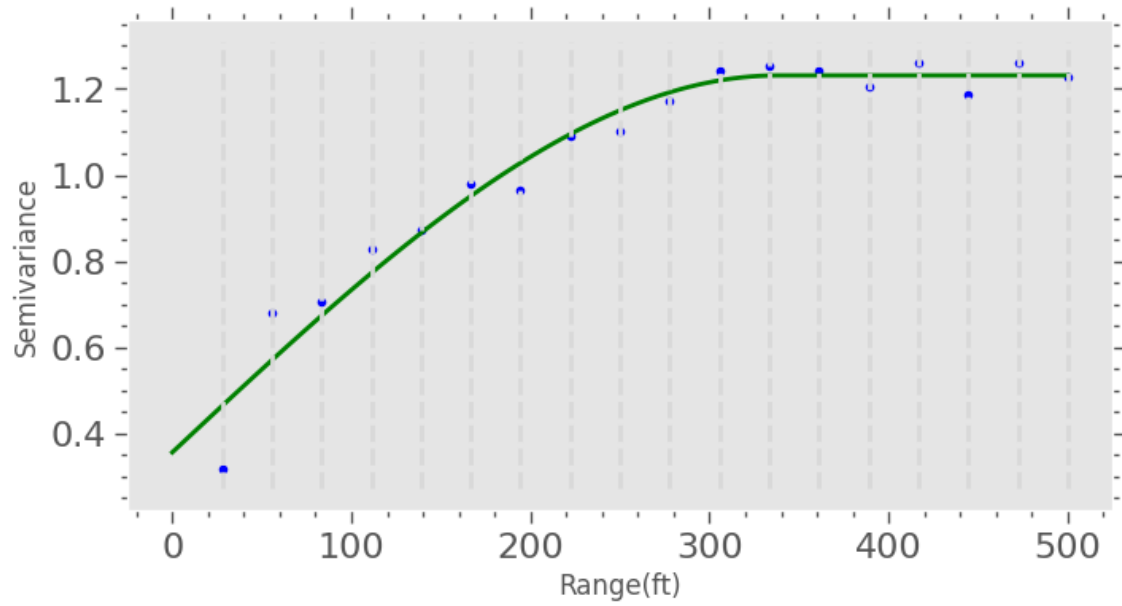
Σχήμα 4.4 Ιστόγραμμα αποτελεσμάτων λογαρίθμου της περιεκτικότητας σε % Cu των σύνθετων δειγμάτων.

Στον ακόλουθο [πίνακα 4.2](#) παρουσιάζονται τα στατιστικά μεγέθη για τον λογάριθμο της περιεκτικότητας σε % Cu.

Πίνακας 4.2 Στατιστικά μεγέθη για τον λογάριθμο της περιεκτικότητας σε % Cu..

Στατιστικά μεγέθη	Τιμή
Αριθμός δειγμάτων (Number of Samples)	858
Αριθμητικός μέσος (Arithmetic Average)	-0.893
Διακύμανση (Variance)	1.182
Τυπική απόκλιση (Standard Deviation)	1.087
Συντελεστής ασυμμετρίας (Coefficient of Skewness)	-0.617
Συντελεστής κύρτωσης (Coefficient of Kurtosis)	0.401
Διάμεσος (Median)	-0.781
Τροποποιημένος μέσος (10%) (10% Trim Range)	-0.817

Η χωρική μεταβολή των μετασχηματισμένων δεδομένων (λογάριθμοι της περιεκτικότητας) αποδίδεται ικανοποιητικά από ένα ημιβαριόγραμμα σφαιρικού τύπου, όπως φαίνεται στο [Σχήμα 4.5](#). Η ακτίνα επιρροής είναι 340ft και οι παράμετροι $C_0=0.36$ και $C=0.87$. Το οποίο δείχνει ότι υπάρχει χωρική συσχέτιση έως τα 340ft και λόγω του C_0 υπάρχει σχετικά μεγάλη διακύμανση σε πολύ μικρές αποστάσεις πριν αρχίσει να επιδρά η χωρική συσχέτιση.



Σχήμα 4.5 Ημιβαριόγραμμα λογαριθμισμένων περιεκτικότητων %Cu.

5 Ανάπτυξη και εφαρμογή νευρωνικών δικτύων

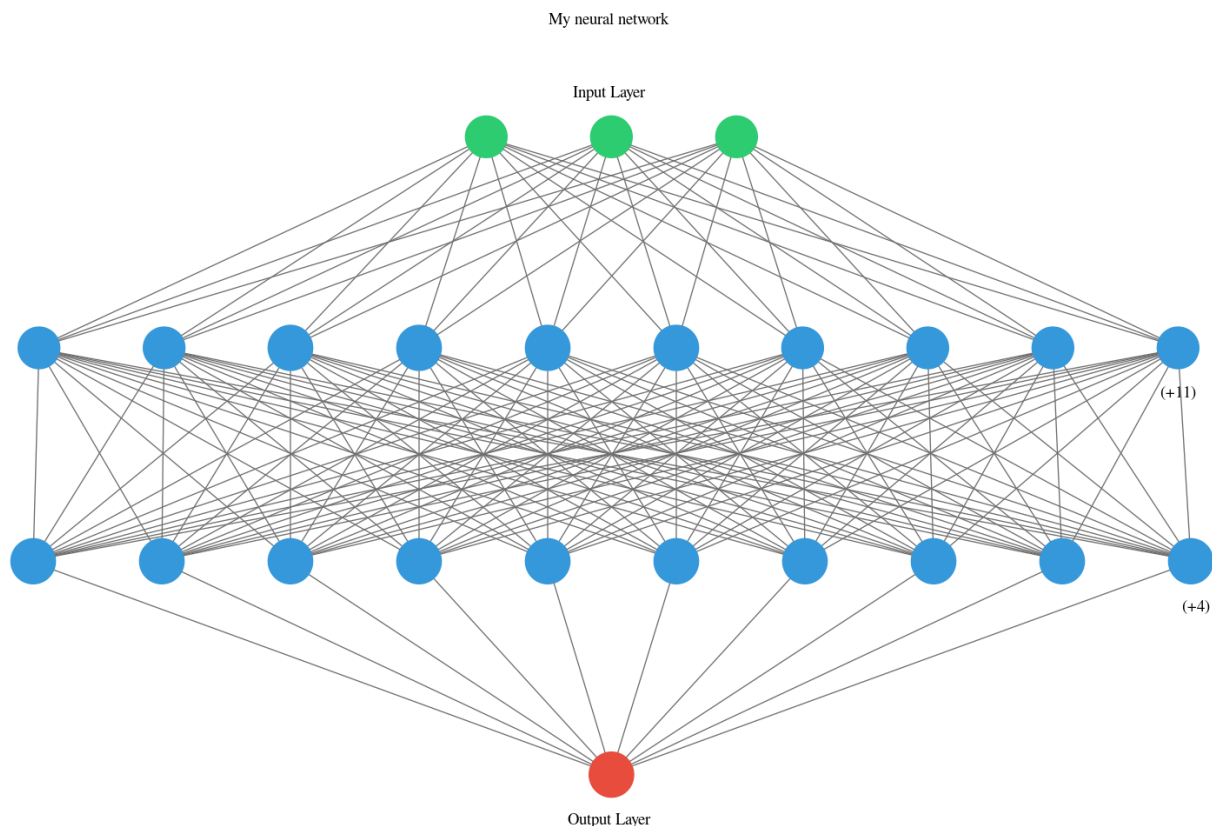
5.1 Ανάπτυξη νευρωνικού δικτύου

Για την ανάπτυξη του νευρωνικού μοντέλου γράφτηκε κώδικας σε γλώσσα προγραμματισμού python και των βιβλιοθηκών της όπως φαίνεται στο [Παράρτημα II](#). Αναπτύχθηκαν 3 διαφορετικές παραλλαγές νευρωνικών δικτύων. Στην πρώτη περίπτωση αναπτύχθηκε ένα νευρωνικό δίκτυο το οποίο εκπαιδεύτηκε με τη χρήση όλων των σύνθετων δειγμάτων των γεωτρήσεων. Στη συνέχεια αναπτύχθηκε ένα δεύτερο νευρωνικό δίκτυο με δύο κρυφά επίπεδα που εκπαιδεύτηκε με αναδειγματοληψία (resampling). Δηλαδή δημιουργήθηκαν καινούργια δεδομένα εκπαίδευσης για να γίνει καλύτερη αντιπροσώπευση των δεδομένων με μεγάλη περιεκτικότητα χαλκού. Τέλος, αναπτύχθηκε ένα τρίτο με αναδειγματοληψία, αλλά με ένα κρυφό επίπεδο και με 100 νευρώνες. Αναπτύχθηκαν οι 3 παραλλαγές για να εξεταστεί η επίδραση που έχει η αναδειγματοληψία και αν είναι πιο γρήγορη η εκπαίδευση με ένα κρυφό επίπεδο και πολλούς νευρώνες ή δύο κρυφά επίπεδα με πιο λίγους νευρώνες συνολικά. Στον παρακάτω [πίνακα 5.1](#) φαίνονται το μέσο τετραγωνικό σφάλμα (MSE) και ο συντελεστής R^2 για τα δεδομένα της εκπαίδευσης και του ελέγχου της γενίκευσης για κάθε παραλλαγή νευρωνικού δικτύου. Παρατηρείται ότι μετά την αναδειγματοληψία (2^ο και 3^ο νευρωνικό δίκτυο) μειώθηκαν τα σφάλματα και αυξήθηκε το R^2 για τα δεδομένα της εκπαίδευσης, ενώ για τα δεδομένα του ελέγχου γενίκευσης οι αλλαγές ήταν μικρές.

Πίνακας 5.1 Μέσο τετραγωνικό σφάλμα εκπαίδευσης (training) και ελέγχου (testing) και οι αντίστοιχοι συντελεστές R^2 για τα τρία νευρωνικά δίκτυα

	MSE εκπαίδευσης	MSE ελέγχου γενίκευσης	R^2 εκπαίδευσης	R^2 ελέγχου γενίκευσης
1 ^ο νευρωνικό δίκτυο	0.62	0.76	0.58	0.53
2 ^ο νευρωνικό δίκτυο	0.19	0.70	0.82	0.54
3 ^ο νευρωνικό δίκτυο	0.27	0.77	0.71	0.53

Όπως φαίνεται και στο [Σχήμα 5.1](#) επιλέχθηκε το 2^ο νευρωνικό δίκτυο με 2 κρυφά επίπεδα επειδή τα αποτελέσματα είναι καλύτερα, λόγω χαμηλότερου σφάλματος, και ήταν 4 φορές πιο γρήγορη στην εκπαίδευση. Η δομή του νευρωνικού δικτύου αποτελείται από 3 εισόδους (X,Y και Z συντεταγμένες του σύνθετου δείγματος), 2 κρυφά επίπεδα με 21 και 14 νευρώνες με relu και gelu συναρτήσεις ενεργοποίησης αντίστοιχα και τέλος 1 έξοδο την %Cu.



Σχήμα 5.1 Δομή επιλεχθέντος νευρωνικού δικτύου.

Το σύνολο των δεδομένων που χρησιμοποιήθηκαν για την εκπαίδευση, τον έλεγχο της εκπαίδευσης και τον έλεγχο της γενίκευσης των νευρωνικών δικτύων αποτελείται από 858 σύνθετα δείγματα. Από αυτά το 70% των δειγμάτων χρησιμοποιήθηκε για την εκπαίδευση (training set), το 15% για τον έλεγχο της εκπαίδευσης (validation set) και το 15% για τον έλεγχο της γενίκευσης (testing set). Ο χωρισμός τους έγινε με τυχαίο τρόπο. Αντί των περιεκτικότητας του Cu χρησιμοποιήθηκαν οι λογαριθμισμένες τιμές τους με την εντολή `np.log(x)` της python όπου συγκρίθηκαν με τις τιμές του νευρικού δικτύου για να μειωθεί το σφάλμα. Ο μετασχηματισμός αυτός βοήθησε στην αποφυγή των αρνητικών τιμών που δεν έχουν φυσικό νόημα στο πρόβλημα αυτό. Επίσης έγινε κανονικοποίηση των δεδομένων εισόδου. Αυτό γίνεται με την εντολή `StandardScaler()` της python και εφαρμόζει την παρακάτω [εξ. \(4.1\)](#) σε κάθε X,Y και Z.

$$x_i = \frac{X_i - X_{avg}}{std} \quad (5.1)$$

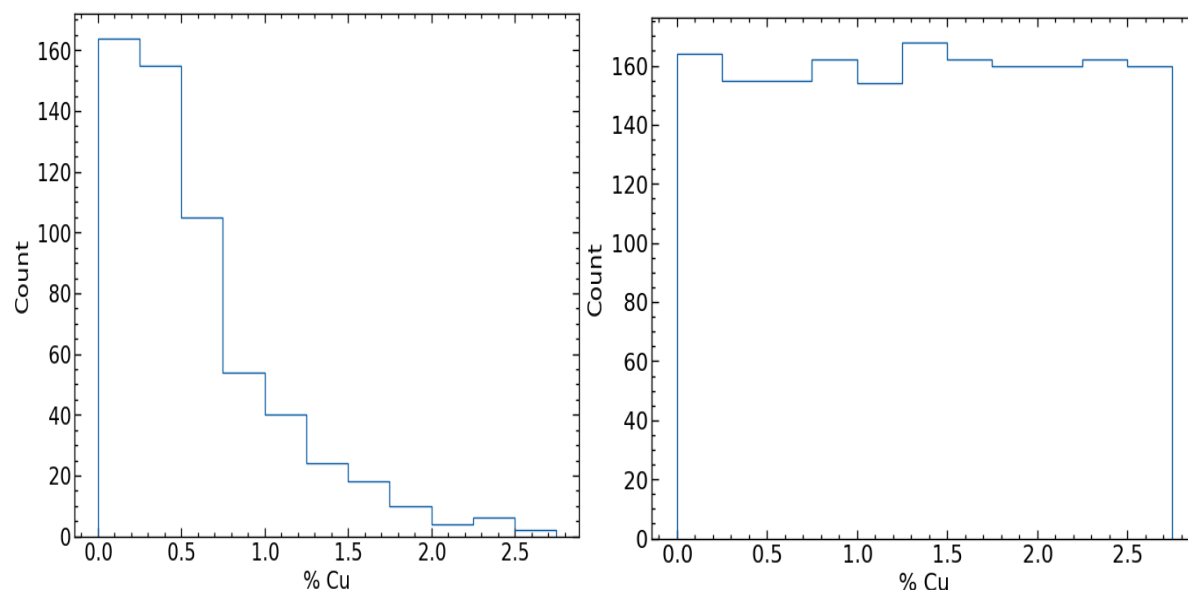
όπου: X_i είναι η τιμή X,Y ή Z

X_{avg} είναι η μέση τιμή των X,Y ή Z

std είναι η τυπική απόκλιση των X,Y ή Z

x_i είναι η κανονικοποιημένη τιμή των X,Y ή Z

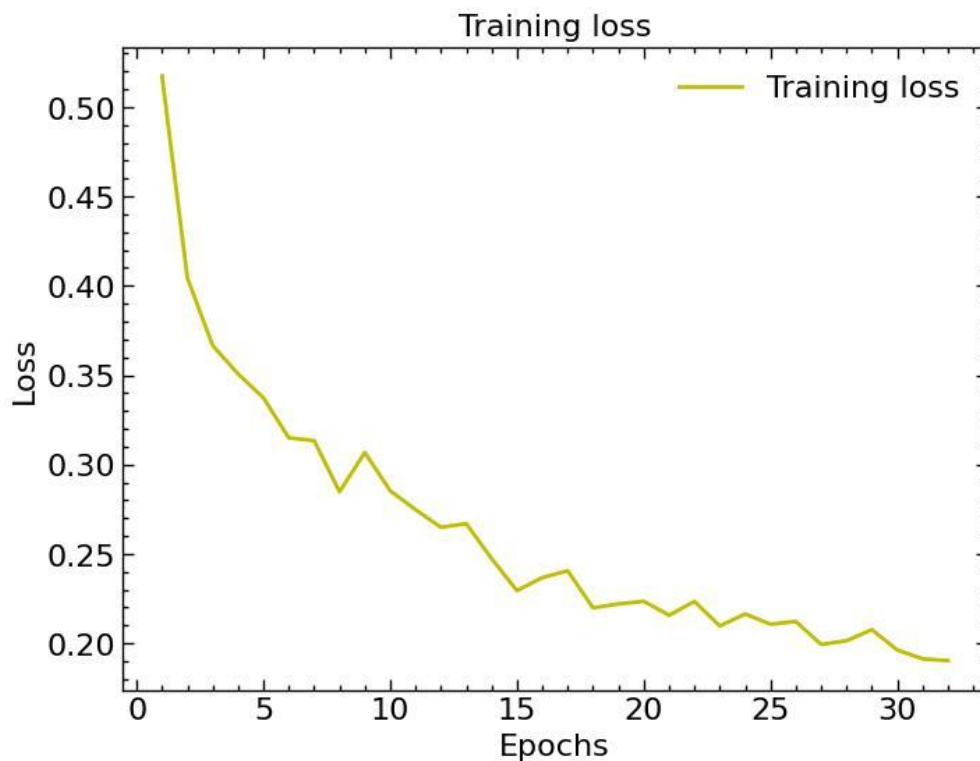
Επίσης τα δείγματα της εκπαίδευσης τροποποιήθηκαν με ειδική τεχνική αναδειγματοληψίας (resampling) έτσι ώστε να αντιπροσωπευτούν ισότιμα στην εκπαίδευση του νευρωνικού δικτύου όλες οι κλάσεις τιμών των περιεκτικότητας του % Cu. Στο παρακάτω [Σχήμα 5.2](#) φαίνεται το ιστόγραμμα κατανομής της περιεκτικότητας σε Cu % των αρχικών δεδομένων εκπαίδευσης και των τροποποιημένων με βάση την τεχνική της αναδειγματοληψίας.



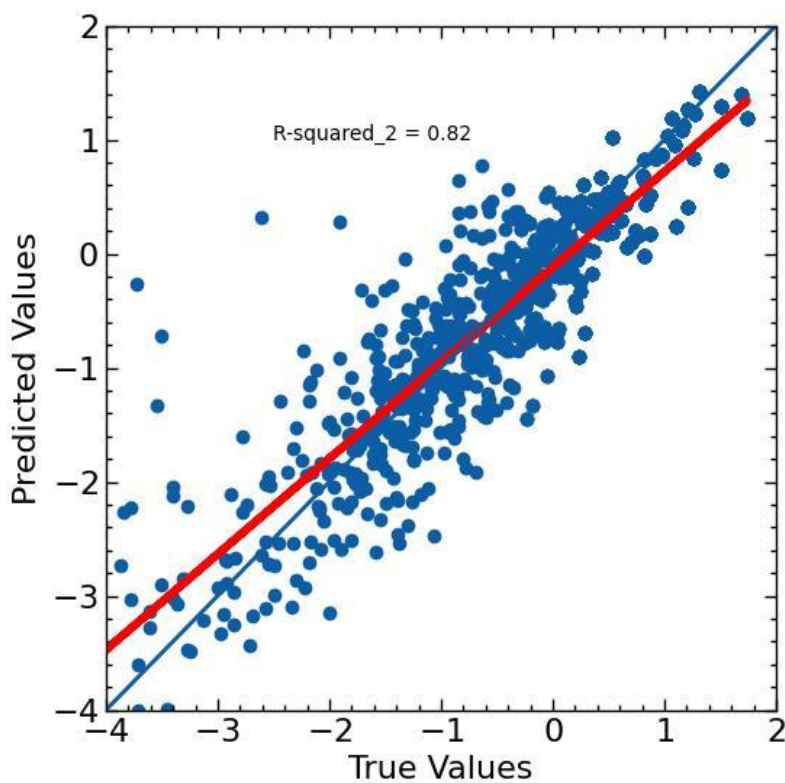
Σχήμα 5.2 Ιστόγραμμα κατανομής της περιεκτικότητας σε Cu % των αρχικών δεδομένων εκπαίδευσης και των τροποποιημένων με βάση την τεχνική της αναδειγματοληψίας.

5.2 Εφαρμογή νευρωνικού δικτύου για την κατασκευή του ψηφιακού μοντέλου του κοιτάσματος και τον υπολογισμό των αποθεμάτων και της ποιότητας

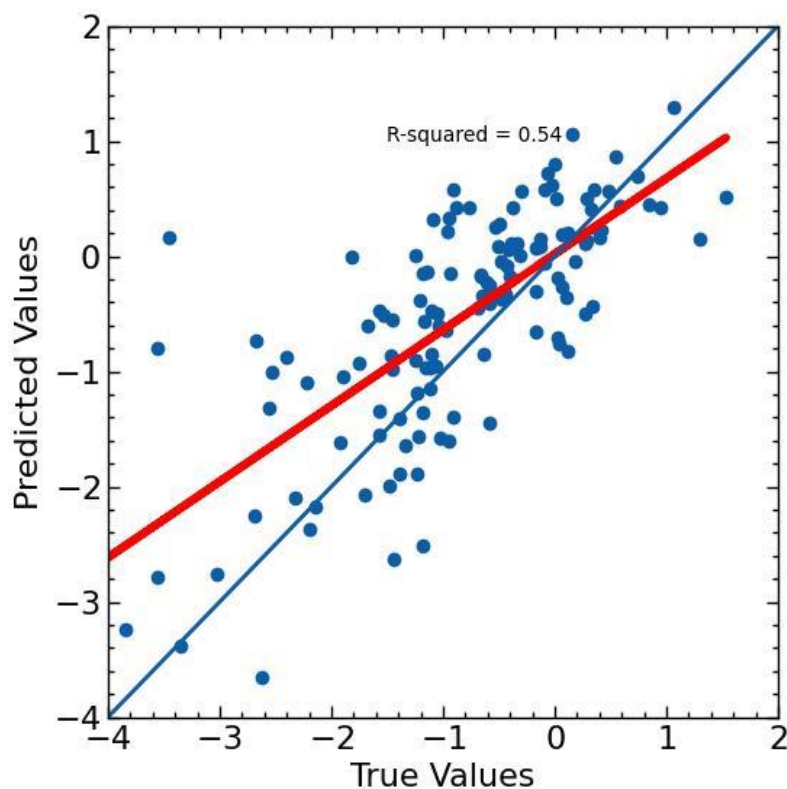
Κατά την εκπαίδευση του νευρωνικού δικτύου το μέσο τετραγωνικό σφάλμα (mse) μειώθηκε σταδιακά από την τιμή 0.5 μέχρι την τιμή 0.19 μετά από 32 κύκλους εκπαίδευσης (epochs) όπου σταμάτησε και η εκπαίδευση, όπως φαίνεται στο [Σχήμα 5.3](#). Από το σημείο αυτό και μετά δεν υπάρχει μεταβολή του σφάλματος με τους κύκλους εκπαίδευσης. Στο [Σχήμα 5.4](#) παρουσιάζεται το διάγραμμα συσχέτισης των πραγματικών τιμών με τις εκτιμήσεις του νευρωνικού δικτύου, όπου η κόκκινη γραμμή είναι η γραμμή τάσης για τις εκτιμήσεις ενώ η μπλε είναι η ιδανική γραμμή, για τα δεδομένα της εκπαίδευσης. Τόσο από το σχήμα όσο και από την τιμή του συντελεστή γραμμικής συσχέτισης R^2 που υπολογίστηκε (0.82) φαίνεται ότι η συσχέτιση των πραγματικών τιμών με τις εκτιμήσεις του νευρωνικού δικτύου εκπαίδευσης είναι αρκετά υψηλή, γεγονός που δείχνει ότι έχει εκπαιδευτεί επαρκώς. Το αντίστοιχο διάγραμμα για τα δεδομένα του ελέγχου της γενίκευσης (testing), που φαίνεται στο [Σχήμα; 5.5](#), δείχνει ότι το νευρωνικό δίκτυο έχει επιτύχει μια σχετική καλή ικανότητα γενίκευσης.



Σχήμα 5.3 Διάγραμμα μέσου τετραγωνικού σφάλματος σε σχέση με τους κύκλους εκπαίδευσης (epochs).



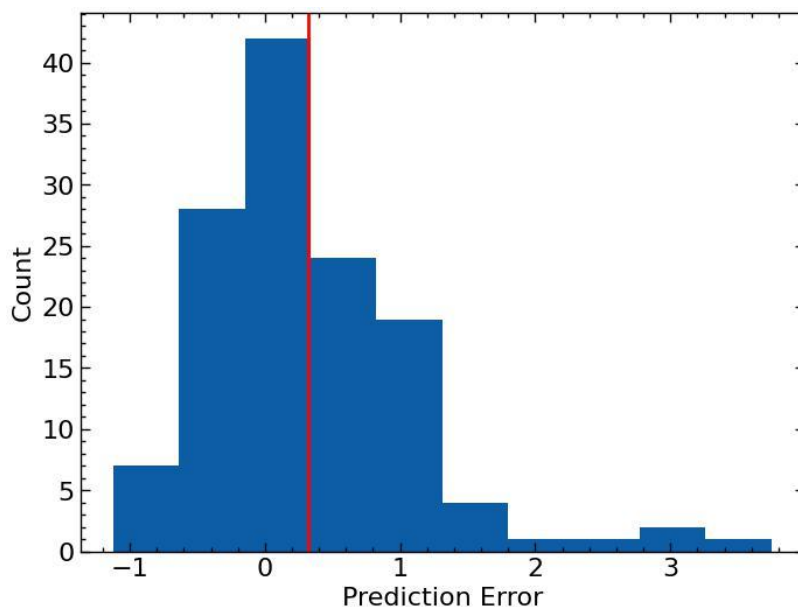
Σχήμα 5.4 Διάγραμμα συσχέτισης τιμών μοντέλου και πραγματικών τιμών (λογάριθμοι της %Cu) για τα δεδομένα εκπαίδευσης (training). Η μπλε γραμμή είναι η ιδανική ευθεία $Y=X$ και η κόκκινη η γραμμή τάσης.



Σχήμα 5.5 Διάγραμμα συσχέτισης τιμών μοντέλου και πραγματικών τιμών (λογάριθμοι της %Cu) για τα δεδομένα ελέγχου της γενίκευσης (testing). Η μπλε γραμμή είναι η ιδανική ευθεία $Y=X$ και η κόκκινη η γραμμή τάσης.

Στα παραπάνω διαγράμματα παρατηρείται ότι στις μικρές τιμές (λογάριθμοι της %Cu) γίνεται μια υπερεκτίμηση, στις μεγάλες τιμές μια υποεκτίμηση ενώ στις τιμές γύρω από το 0 η εκτίμηση έχει το μικρότερο σφάλμα.

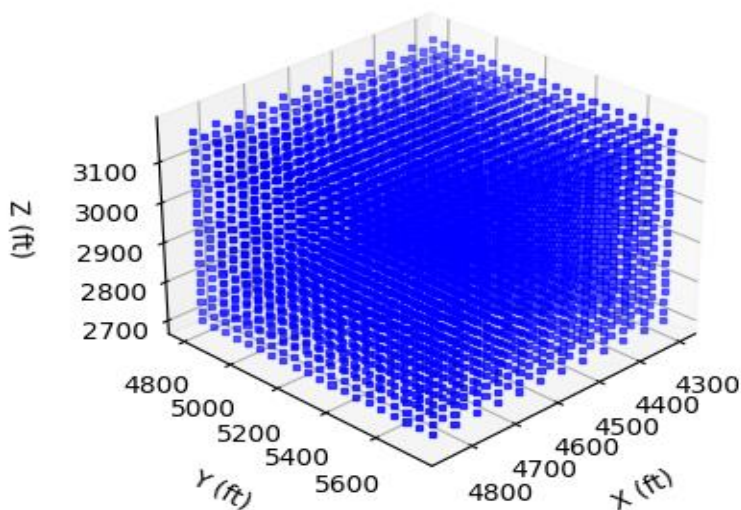
Στο [Σχήμα 5.6](#) παρουσιάζεται η κατανομή του σφάλματος εκτίμησης της περιεκτικότητας σε % Cu (σφάλμα εκτίμησης= πραγματική τιμή – εκτιμηθείσα τιμή), η οποία παρουσιάζει μια μικρή απόκλιση από την κανονική κατανομή με επιμήκυνση του άκρου της προς τα δεξιά.



Σχήμα 5.6 Κατανομή του σφάλματος εκτίμησης της περιεκτικότητας σε % Cu από το νευρωνικό δίκτυο.

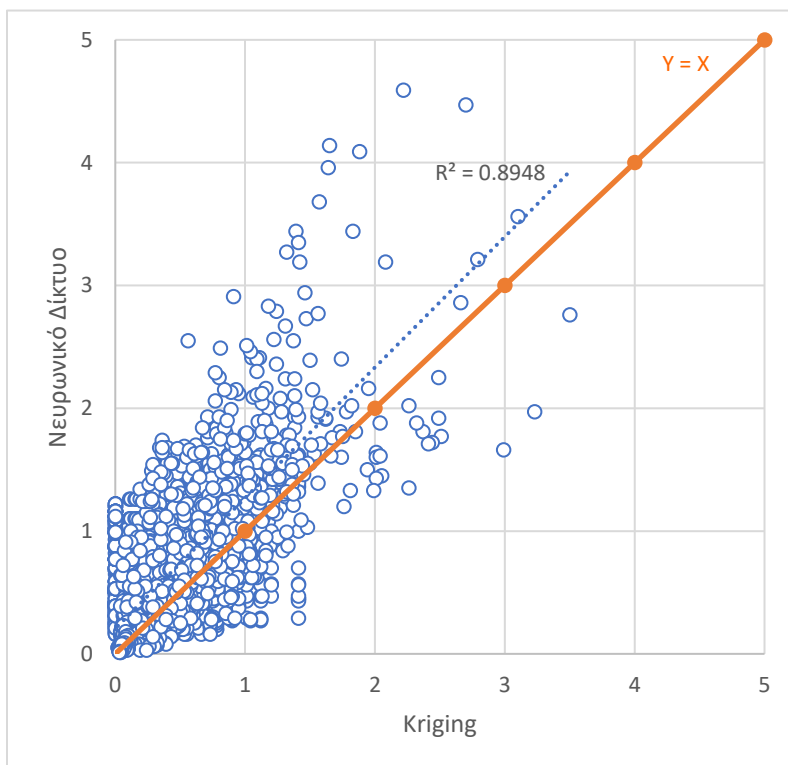
5.3 Εφαρμογή του νευρωνικού δικτύου για την ανάπτυξη του ψηφιακού μοντέλου και σύγκριση με τη μέθοδο IDS και Kriging

Στη συνέχεια δημιουργήθηκε το ψηφιακό μοντέλο του κοιτάσματος του χαλκού με χρήση του λογισμικού CSMine. Οι διαστάσεις του μπλοκ επιλέχθηκαν $\Delta x = \Delta y = 50\text{ft}$ και $\Delta z = 25\text{ft}$ και ο αριθμός των μπλοκ ανά άξονα, $N_x = 12$ και $N_y = N_z = 20$ άρα συνολικά $12 \times 20 \times 20 = 4800$ μπλοκ. Στο παρακάτω Σχήμα 5.7 φαίνονται τα μπλοκ του ψηφιακού μοντέλου που δημιουργήθηκε

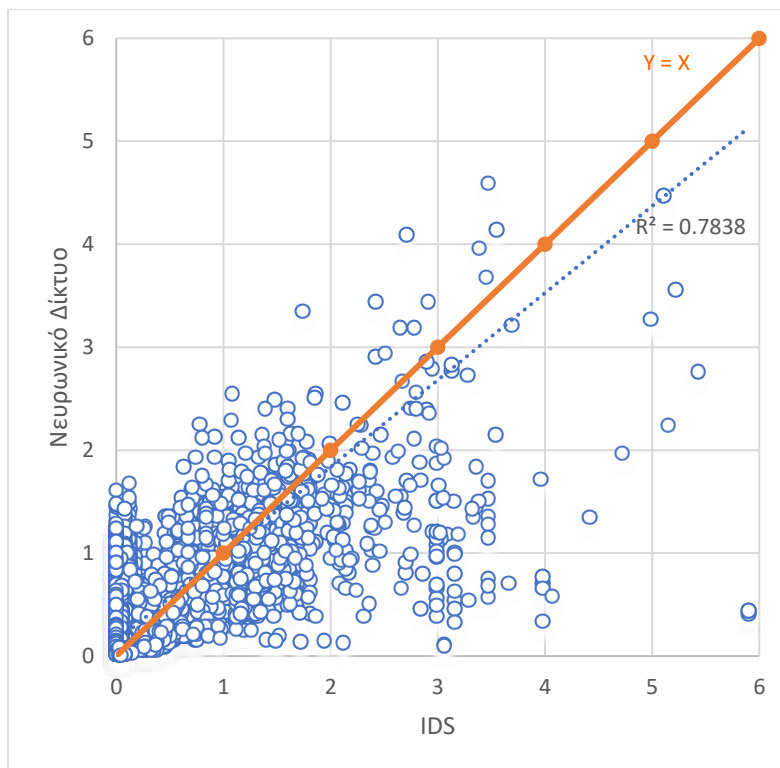


Σχήμα 5.7 Τρισδιάστατη απεικόνιση του ψηφιακού μοντέλου

Έπειτα οι συντεταγμένες των μπλοκ εισήχθησαν στο εκπαιδευμένο νευρωνικό δίκτυο που αναπτύχθηκε ανωτέρω για να γίνει η εκτίμηση της περιεκτικότητας του χαλκού σε κάθε μπλοκ. Τα αποτελέσματα του νευρωνικού δικτύου συγκρίθηκαν με τα αποτελέσματα που προέκυψαν μέσω της γεωστατιστικής μεθόδου Kriging (Σχήμα 5.7) και με τα αποτελέσματα της μεθόδου IDS (Σχήμα 5.8). Παρατηρείται ότι οι τιμές που υπολογίστηκαν από το νευρωνικό δίκτυο έχουν καλή συσχέτιση με εκείνες των μεθόδων Kriging και IDS.



Σχήμα 5.8 Διάγραμμα συσχέτισης των περιεκτικότητων σε % Cu των νευρωνικών δικτύων σε σχέση με την μέθοδο Kriging.



Σχήμα 5.9 Διάγραμμα συσχέτισης των περιεκτικότητων σε % Cu των νευρωνικών δικτύων σε σχέση με την μέθοδο IDS.

Πριν γίνει ο υπολογισμός των βέλτιστων ορίων εκσκαφής, πρέπει να εισαχθούν τα οικονομικά στοιχεία και να υπολογιστεί η αξία κάθε μπλοκ. Ως τιμή του μετάλλου Cu χρησιμοποιήθηκε η μέση τιμή των δέκα τελευταίων ετών και τα υπόλοιπα οικονομικά στοιχεία έχουν παρθεί από παλαιότερες περιπτώσεις (Arteaga, 2014) με αναπροσαρμογή στις σημερινές συνθήκες. Τα στοιχεία αυτά είναι:

- Το κόστος εξόρυξης, μεταφοράς και διοικητικό κόστος για μεταλλευμα ή άγονα (Mining Cost): 2.5 \$/ton (A)
- Επιπλέον κόστος μεταφοράς για κάθε κατώτερη βαθμίδα από την πρώτη, λόγω αύξησης της απόστασης μεταφοράς (incremental cost): 0.01 \$/βαθμίδα (B)
- Κόστος εμπλουτισμού μεταλλεύματος Cu (processing cost): 10\$/ton (Γ)
- Κόστος μεταλλουργικής επεξεργασίας μεταλλεύματος Cu (treatment cost): 0.15 \$/lb Cu (Δ)
- Τιμή μετάλλου Cu (metal price): 3.5 \$/lb (E)
- Ανάκτηση μετάλλου Cu: 0.9 (ΣΤ)
- Τελική γωνία κλίσης εκμετάλλευσης: 45°
- Πυκνότητα μεταλλεύματος και αγόνων: 2.4 t/m³ (H)
- V_{block} =Όγκος μπλοκ που υπολογίζεται από τις διαστάσεις του ΔX, ΔY, ΔZ.

Ο υπολογισμός της οικονομικής αξίας του μπλοκ μεταλλεύματος BV_m που βρίσκεται στη βαθμίδα i γίνεται από την παρακάτω εξ. (5.2):

$$BV_m (\$) = V_{\text{μπλοκ}} \cdot H \cdot ((E - \Delta) \cdot 22.04 \cdot \Sigma T \cdot \% Cu) - \Gamma - A - (i - 1) \cdot B \quad (5.2)$$

Ο υπολογισμός του κόστους του μπλοκ αγόνων BV_w γίνεται από την [εξ. \(5.3\)](#):

$$BV_w(\$) = -A - (Z_{\text{max}} - Zi) * B \quad (5.3)$$

Σε κάθε μπλοκ του μοντέλου γίνονται δυο υπολογισμοί, θεωρώντας ότι μπορεί να είναι μετάλλευμα ή άγονο και ως αξία μπλοκ BV θεωρείται η μεγαλύτερη τιμή των BV_m και BV_w [εξ. \(5.4\)](#):

$$BV (\$) = \max \{BV_w, BV_m\} \quad (5.4)$$

6 Αλγόριθμος ψευδοροής για την εύρεση των βέλτιστων ορίων

6.1 Από το Lerchs-Grossmann στον Pseudoflow: μια σύντομη αναδρομή

Για την εύρεση των βέλτιστων ορίων εκσκαφής μια υπαίθριας εκμετάλλευσης χρησιμοποιούνται δύο βασικές παράμετροι:

1. οι οικονομικές αξίες των στοιχειωδών όγκων (μπλοκ) του ψηφιακού μοντέλου του κοιτάσματος και
2. η κλίση εκμετάλλευσης η οποία εισάγει περιορισμούς στη διαδοχή εξόρυξης των μπλοκ.

Η βέλτιστη εκσκαφή είναι εκείνη που προκύπτει μετά την αφαίρεση όλων εκείνων των μπλοκ που ικανοποιούν τις απαιτήσεις των κλίσεων της εκμετάλλευσης και μεγιστοποιούν το οικονομικό αποτέλεσμα. Οι μελέτες για την επίλυση του προβλήματος βελτιστοποίησης μιας υπαίθριας εκμετάλλευσης με ανάπτυξη κατάλληλων αλγορίθμων για υπολογιστές ξεκίνησαν τη δεκαετία του 1960.

Ο αλγόριθμος Lerchs-Grossmann (LG), που βασίζεται στη θεωρία των γράφων ή γραφημάτων (Graph theory), δημοσιεύθηκε το 1965 και ήταν μια από τις πρώτες μεθόδους υπολογισμού των βέλτιστων ορίων εκσκαφής μιας υπαίθριας εκμετάλλευσης (*Lerchs και Grossman, 1965*). Ο αλγόριθμος LG, επειδή έχει τη σχετική θεωρητική τεκμηρίωση, δίνει πάντα τη βέλτιστη εκσκαφή σε αντίθεση με άλλους ευρετικούς αλγορίθμους (π.χ. κινητός κώνος, Korobov κ.α.) που δεν δίνουν πάντα την βέλτιστη εκσκαφή.

Στη δεκαετία του 1980, υλοποιήθηκε η πρώτη εφαρμογή του με ανάπτυξη σχετικού λογισμικού με την ονομασία Whittle Three-D και η μέθοδος LG αποτέλεσε το πρότυπο για τη βελτιστοποίηση των ορίων εκσκαφής μια υπαίθριας εκμετάλλευσης στη μεταλλευτική βιομηχανία. Το κύριο μειονέκτημα με τη μέθοδο LG είναι ο σημαντικός χρόνος που απαιτείται για τον προσδιορισμό του βέλτιστου ορίου, ιδιαίτερα αν πρόκειται για ψηφιακά μοντέλα κοιτασμάτων με μεγάλο αριθμό μπλοκ.

Για την επίλυση του προβλήματος αυτού η έρευνα επικεντρώθηκε στην ανάπτυξη πιο αποτελεσματικών αλγορίθμων βελτιστοποίησης. Ο Picard (*1976*) απέδειξε ότι το πρόβλημα βελτιστοποίησης των ορίων θα μπορούσε να λυθεί με τους πιο αποτελεσματικούς αλγορίθμους της μέγιστης ροής. Οι Goldberg και Tarjan (*1988*) ανέπτυξαν έναν εξαιρετικά αποδοτικό αλγόριθμο μέγιστης ροής που ονομάζεται μέθοδος Push-Relabel. Τέλος ο Hochbaum (*2008*) ανέπτυξε έναν αλγόριθμο ψευδοροής (Pseudoflow), ο οποίος αποδείχθηκε ότι είναι πιο αποτελεσματικός από τον LG και άλλους διαδομένους αλγορίθμους μέγιστης ροής, όπως η μέθοδος Push-Relabel (*Chandran και Hochbaum, 2009 και Muir, 2004*).

6.2 Αλγόριθμος ψευδοροής

Η κατανόηση του τρόπου λειτουργίας του αλγόριθμου ψευδοροής απαιτεί σημαντική γνώση των μαθηματικών και της επιστήμης των υπολογιστών και περιλαμβάνει 2 βασικά στάδια:

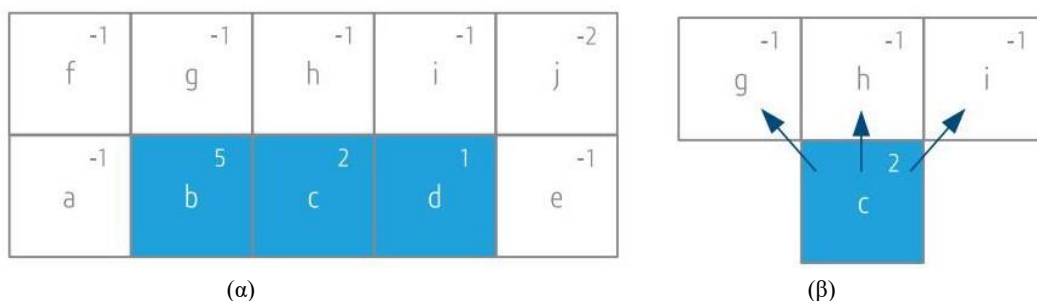
- Την μοντελοποίηση του προβλήματος της εύρεσης των βέλτιστων ορίων εκσκαφής μια υπαίθριας εκμετάλλευσης με βάση τη θεωρία των γράφων.
- Την επίλυση του προβλήματος της εύρεσης των μπλοκ της εκσκαφής που ικανοποιούν τις απαιτήσεις των κλίσεων της εκμετάλλευσης και μεγιστοποιούν το οικονομικό αποτέλεσμα.

6.2.1 Έννοιες από τη θεωρία των γράφων και τη βελτιστοποίηση των ορίων εκσκαφής μια υπαίθριας εκμετάλλευσης

Όπως αναφέρθηκε ήδη η εύρεση των βέλτιστων ορίων εκσκαφής μια υπαίθριας εκμετάλλευσης βασίζεται στο ψηφιακό μοντέλο του κοιτάσματος (block model) και χρησιμοποιούνται δύο βασικές παράμετροι εισόδου στον αλγόριθμο βελτιστοποίησης:

- οι οικονομικές αξίες των στοιχειωδών όγκων (μπλοκ) του ψηφιακού μοντέλου του κοιτάσματος, και
- η κλίση εκμετάλλευσης η οποία εισάγει περιορισμούς στη διαδοχή εξόρυξης των μπλοκ.

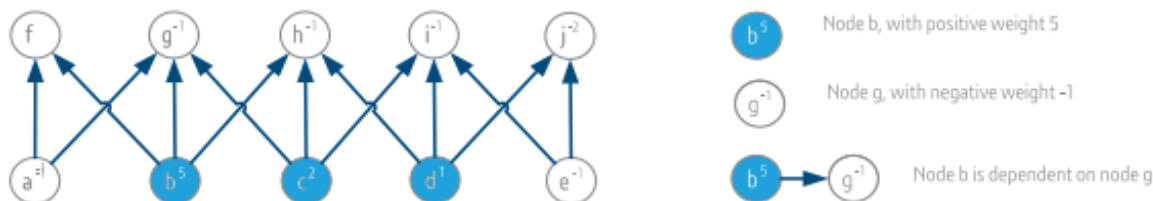
Η απαίτηση συγκεκριμένων κλίσεων στην εκσκαφή και η ακολουθία εξόρυξης των μπλοκ μπορούν να εκφραστούν ως εξαρτήσεις μεταξύ των μπλοκ. Για παράδειγμα, στο [Σχήμα 6.1α](#), φαίνεται ένα απλό μοντέλο 2D ενός κοιτάσματος που αποτελείται από 10 μπλοκ με ονομασία από «a» έως «j», ενώ η οικονομική αξία κάθε μπλοκ σημειώνεται στην επάνω δεξιά γωνία του μπλοκ. Τα λευκά μπλοκ είναι εκείνα με τις αρνητικές αξίες που αντιπροσωπεύουν τα άγονα υλικά ή μετάλλευμα χαμηλής ποιότητας, ενώ τα μπλε είναι εκείνα με τις θετικές αξίες που αντιπροσωπεύουν το μέταλλευμα. Για να διατηρηθούν κλίσεις 45 μοιρών κατά την εκσκαφή του μπλοκ "c" απαιτείται πρώτα να αφαιρεθούν τα μπλοκ "g", "h" και "i". Ο περιορισμός αυτός απεικονίζεται ως εξάρτηση του μπλοκ "c" από τα μπλοκ "g", "h" και "i" και απεικονίζεται με τόξα που ξεκινούν από το μπλοκ "c" και καταλήγουν τα μπλοκ "g", "h" και "i", όπως φαίνεται στο [Σχήμα 6.1β](#).



Σχήμα 6.1 (α) Απλό 2D μοντέλο 10 μπλοκ (β) Εξαρτήσεις μεταξύ των μπλοκ λόγω της απαιτούμενης κλίσης εκμετάλλευσης και της διαδοχής των μπλοκ (Bai et al, 2017). .

Το πρόβλημα βελτιστοποίησης των ορίων της εκσκαφής είναι να βρεθεί ένα σύνολο από μπλοκ που υπακούει στους περιορισμούς εξάρτησης και δίνει την υψηλότερη συνολική οικονομική αξία. Αυτό το πρόβλημα μπορεί να αναπαρασταθεί μέσω της θεωρίας των γράφων ή γραφημάτων. Ένας γράφος ή γραφήμα είναι μια εννοιολογική δομή που αποτελείται από κόμβους (nodes) και τόξα

(arcs). Στην περίπτωση της βελτιστοποίησης εκσκαφής, ένας κόμβος αντιπροσωπεύει ένα μπλοκ και ένα τόξο μεταξύ δύο κόμβων αντιπροσωπεύει τη σχέση εξάρτησης δύο μπλοκ για την ακολουθία εκσκαφής και τον περιορισμό κλίσης. Ένας κόμβος μπορεί να φέρει μια τιμή βάρους, για να αντιπροσωπεύει την οικονομική αξία (τιμή του μπλοκ). Το [Σχήμα 6.2](#) δείχνει πως το μοντέλο των μπλοκ του [Σχήματος 6.1](#) μετασχηματίστηκε σε ένα κατευθυνόμενο γράφημα για την αναπαράσταση του προβλήματος βελτιστοποίησης της εκσκαφής.



Σχήμα 6.2 Κατευθυνόμενο γράφημα που προέκυψε από το 2D μοντέλο των 10 μπλοκ του [Σχήματος 6.1a](#) (Bai et al, 2017).

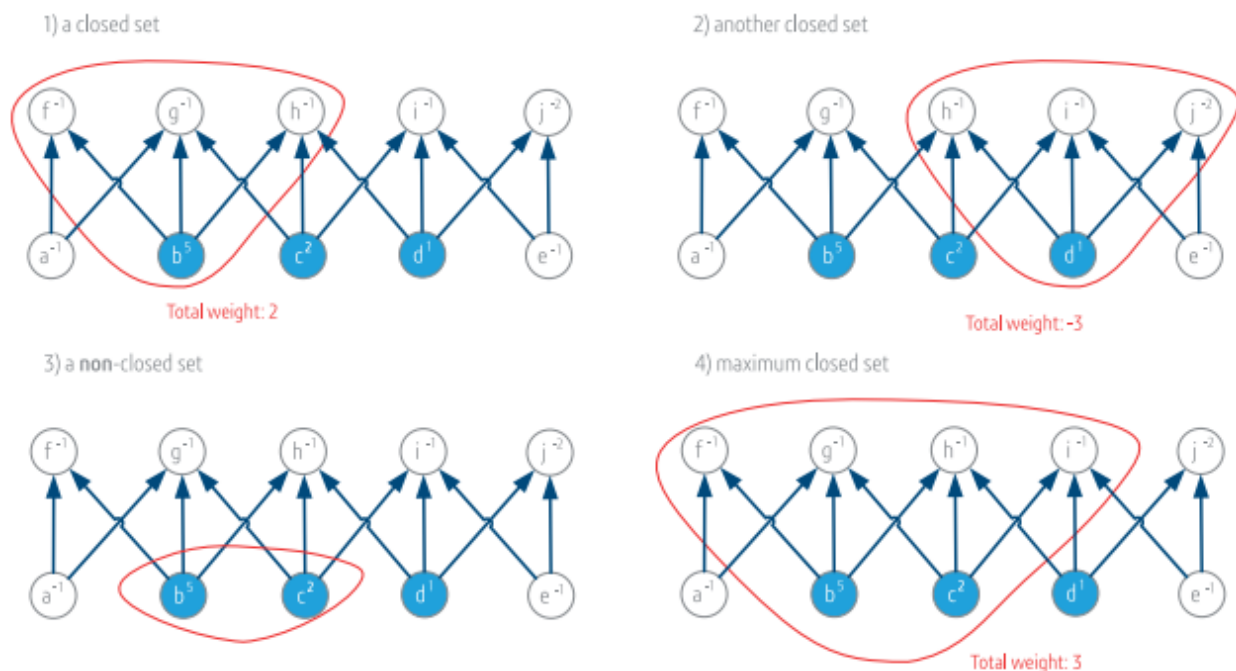
Στη συνέχεια θα χρησιμοποιηθεί το παράδειγμα αυτό για να περιγραφεί πώς χρησιμοποιείται η μέθοδος που βασίζεται στο γράφημα για να υπολογιστεί η βέλτιστη εκσκαφή. Η προσέγγιση είναι γενική και μπορεί να επεκταθεί σε μεγαλύτερα μοντέλα κοιτασμάτων και σε τρεις διαστάσεις που περιέχουν μεγάλο αριθμό κόμβων και τόξων.

6.2.2 Το πρόβλημα του εντοπισμού των βέλτιστων ορίων μιας εκσκαφής με τον αλγόριθμο ψευδοροής (Pseudoflow algorithm)

Η ιδέα της επίλυσης του προβλήματος της βελτιστοποίησης με βάση τον αλγόριθμο της ψευδοροής βασίζεται στην θεώρηση ότι η εξόρυξη των μπλοκ με θετικές οικονομικές αξίες μπορούν να θεωρηθούν ως μια ροή η οποία καθώς κινείται από κόμβο σε κόμβο, σύμφωνα με το κατευθυνόμενο γράφημα, και χρησιμοποιείται για να αντισταθμίζει – «ξεπληρώνει» τις αρνητικές αξίες των κόμβων που αντιστοιχούν σε άγονα υλικά ή μετάλλευμα χαμηλής ποιότητας. Οι κόμβοι με τις θετικές αξίες αποτελούν τις «δεξαμενές» και η αξία του κόμβου αντιπροσωπεύει την ποσότητα που περιέχει, ενώ οι κόμβοι με αρνητικές αξίες είναι οι καταναλωτές και η οικονομική τους αξία δείχνει την κατανάλωση που κάνουν όταν διέρχεται από αυτούς. Η κίνηση γίνεται μέσω των τόξων που συνδέουν τους κόμβους που θεωρούνται ως οι «αγωγοί». Η κίνηση αυτή πραγματοποιείται από τους κατώτερους κόμβους προς τους κόμβους της επιφανείας σύμφωνα με τις υπάρχουσες εξαρτήσεις των κόμβων και για τον λόγο αυτό τα γραφήματα αυτά χαρακτηρίζονται κατευθυνόμενα. Οπότε το πρόβλημα είναι να βρεθεί το κλειστό εκείνο υποσύνολο των κόμβων του δικτύου που δίνει την μέγιστη ροή. Ένα υποσύνολο κόμβων είναι κλειστό αν μπορεί να αφαιρεθεί χωρίς να απαιτείται η αφαίρεση και άλλων κόμβων που δεν ανήκουν σε αυτό.

Στο [Σχήμα 6.3](#) φαίνονται για τις περιπτώσεις 1 και 2 κλειστά υποσύνολα. Στην περίπτωση 1 πρόκειται για το υποσύνολο των κόμβων $\{b, f, g, h\}$ με συνολική αξία $5-1-1-1=2$ και στην περίπτωση 2 για το υποσύνολο των κόμβων $\{d, h, i, j\}$ με συνολική αξία $1-1-1-2=-3$. Αντίθετα στην περίπτωση 3 το υποσύνολο $\{b, c\}$ είναι μη κλειστό, αφού για να αφαιρεθούν τα b, c θα πρέπει πρώτα να έχουν αφαιρεθεί τα f, g, h και i .

Η περίπτωση 4 του σχήματος 5.3 αποτελεί το κλειστό υποσύνολο $\{b, c, f, h, i\}$ με τη μέγιστη αξία $5+2-1-1-1=3$. Το κλειστό αυτό υποσύνολο αυτό με τη μέγιστη αξία αποτελεί τη βέλτιστη εκσκαφή. Το μέγιστο αυτό κλειστό υποσύνολο εντοπίζει με άμεση αναζήτηση ο αλγόριθμος LG. Η αναζήτηση αυτή είναι αρκετά χρονοβόρα όσο αυξάνει ο αριθμός των κόμβων και των εξαρτήσεων, ενώ ο αλγόριθμος ψευδοροής αποδεικνύεται αρκετά πιο γρήγορος έως και 14 ώρες.



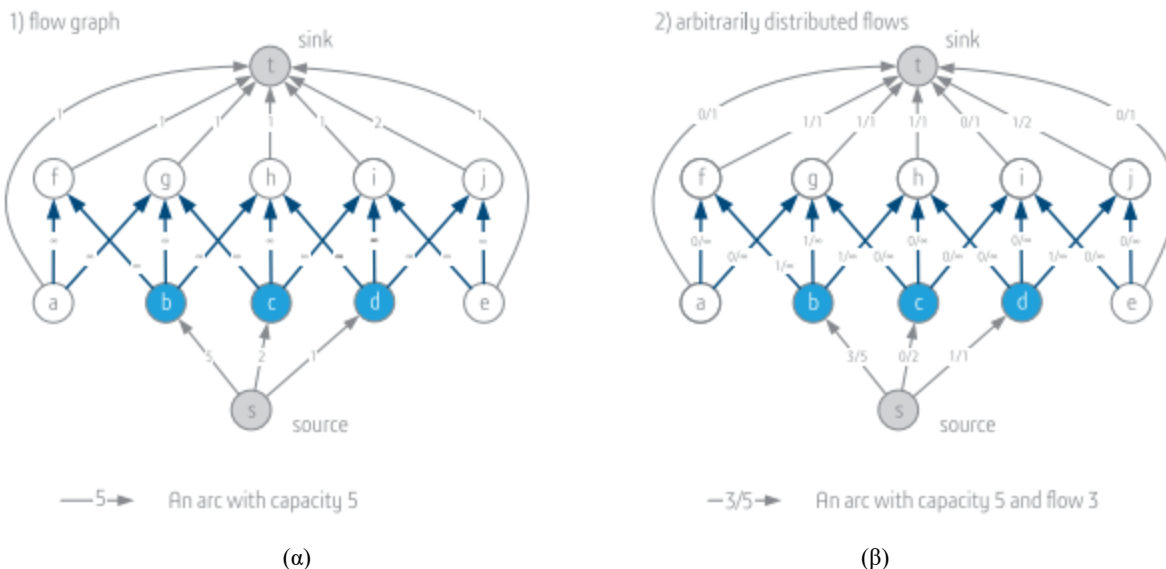
Σχήμα 6.3 Περίπτωσης 1 και 2: Κλειστά υποσύνολα (closed set) του κατευθυνόμενου γραφήματος. Περίπτωση 3: Μη κλειστό υποσύνολο (non-closed set). Περίπτωση 4: Μέγιστο κλειστό υποσύνολο. (Bai et al, 2017).

6.2.3 Υπολογισμός του μέγιστου κλειστού υποσυνόλου με τον αλγόριθμο ψευδοροής

Τα βήματα εφαρμογής του αλγορίθμου για την εύρεση μέγιστου κλειστού υποσυνόλου με βάση τον αλγόριθμο ψευδοροής είναι:

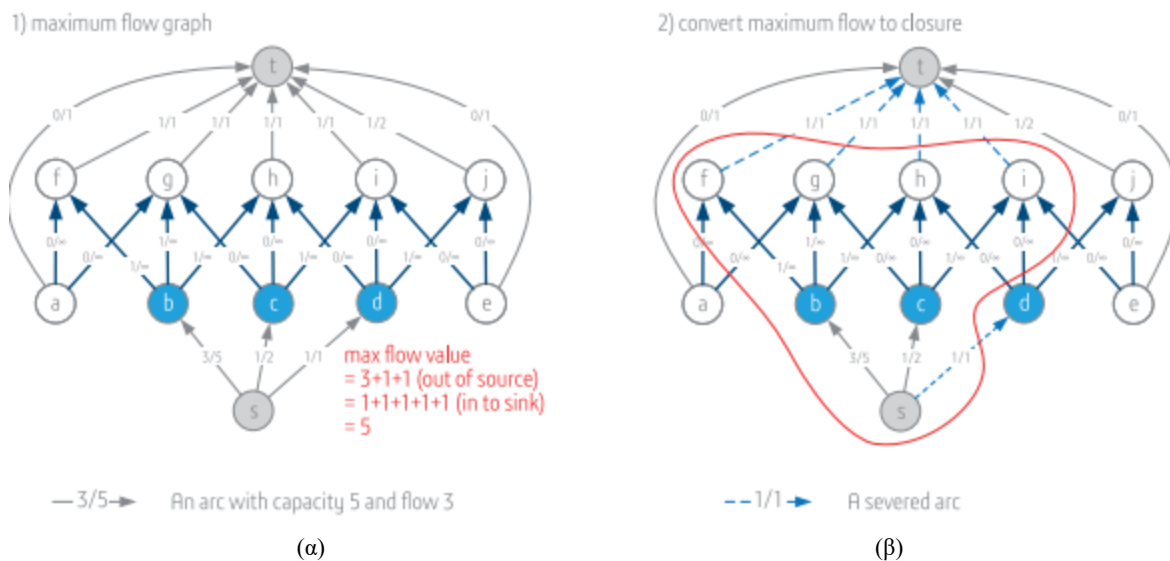
- 1) Δημιουργία του ψηφιακού μοντέλου του κοιτάσματος που αποτελείται από τα μπλοκ με τις οικονομικές αξίες, όπως αναφέρθηκε παραπάνω (Σχήμα 6.1).
- 2) Δημιουργία του κατευθυνόμενου γραφήματος από το μοντέλο των μπλοκς και των εξαρτήσεων που προκύπτουν λόγω της κλίσης της εκμετάλλευσης. Το κατευθυνόμενο διάγραμμα αποτελείται από κόμβους και τόξα, όπως φαίνεται στο Σχήμα 6.2.
- 3) Προσθήκη δυο νέων κόμβων στην αρχή και στο τέλος του κατευθυνόμενου γραφήματος. Ο αρχικός κόμβος ονομάζεται πηγή (source) και αντιστοιχεί στο ξεκίνημα της ροής, ενώ ο τελικός ονομάζεται καταβόθρα (sink) και αντιστοιχεί στον κόμβο που καταλήγουν όλες οι ροές. Μετά από την προσθήκη αυτή υπάρχουν τρεις τύποι κόμβων:
 - Ο κόμβος source από όπου ξεκινάει το γράφημα.
 - Ο κόμβος sink όπου τελειώνει το γράφημα.
 - Οι ενδιάμεσοι κόμβοι που αντιστοιχούν στα μπλοκ.

- 4) Σύνδεση κόμβων γραφήματος με τον κόμβο source και sink και απόδοση βαρών στα τόξα του γραφήματος.
- Ενώνεται με τόξα ο κόμβος source με όλους τους θετικούς κόμβους του γραφήματος και όλοι οι αρνητικοί κόμβοι με τον κόμβο sink, όπως φαίνεται στο [Σχήμα 6.4α](#).
 - Αποδίδονται βάρη στα τόξα με βάση τις τιμές των κόμβων που συνδέουν, ενώ οι κόμβοι εμφανίζονται πλέον στο γράφημα χωρίς τιμές. Ο τρόπος απόδοσης των βαρών στα τόξα είναι ο εξής:
 - Τόξο από source προς θετικό κόμβο: Το βάρος του τόξου θα είναι η τιμή του θετικού κόμβου. Τα βάρη αυτά θεωρούνται ως οι μέγιστες ροές που μπορούν διέλθουν από τα τόξα.
 - Τόξο από αρνητικό κόμβο προς sink: Το βάρος του τόξου θα είναι η απόλυτη τιμή του κόμβου αρχής του τόξου. Τα βάρη αυτά θεωρούνται ως οι μέγιστες ροές που μπορούν διέλθουν από τα τόξα.
 - Τόξο από κόμβο προς κόμβο: εάν επιτρέπεται από την γωνία κλίσης, το βάρος του τόξου θα είναι η τιμή του κόμβου αρχής του τόξου. Για τα τόξα αυτά δεν υφίσταται περιορισμός ως προς τη μέγιστη επιτρεπόμενη τιμή ροής.
- 5) Απόδοση αρχικών τυχαίων τιμών στις ροές.
- Αποδίδονται τυχαίες τιμές στις ροές που εκκινούν από το κόμβο source προς τους θετικούς κόμβους λαμβάνοντας υπόψη ότι αυτές δεν πρέπει να υπερβαίνουν τη μέγιστη επιτρεπτή τιμή, όπως έχει οριστεί στο προηγούμενο βήμα. Σε κάθε τόξο σημειώνεται η τιμή της ροής που αποδόθηκε τυχαία και η μέγιστη επιτρεπόμενη. Στο [Σχήμα 6.4β](#) φαίνεται ότι η τιμή ροής από τον κόμβο s (sink) προς τον κόμβο b έχει την τιμή 3 ενώ η μέγιστη είναι η 5 (σημειώνεται ως 3/5).
 - Στη συνέχεια οι τιμές αυτές κατανέμονται τυχαία στους επόμενους κόμβους σύμφωνα με τα τόξα που τους συνδέουν λαμβάνοντας υπόψη ότι πρέπει σε κάθε κόμβο να τηρείται το ισοζύγιο ροής, δηλαδή το σύνολο των εισερχομένων ροών πρέπει να είναι ίσο με το σύνολο των εξερχομένων. Στο [Σχήμα 5.4β](#) φαίνεται ότι η εισερχόμενη ροή (3) στον κόμβο b κατανέμεται στους κόμβους f (1), g (1) και h(1). Το ισοζύγιο ροής τηρείται αφού $3=1+1+1$. Η απόδοση τιμών συνεχίζεται έως ότου αποδοθούν σε όλο το δίκτυο, όπως φαίνεται στο [Σχήμα 6.4β](#).



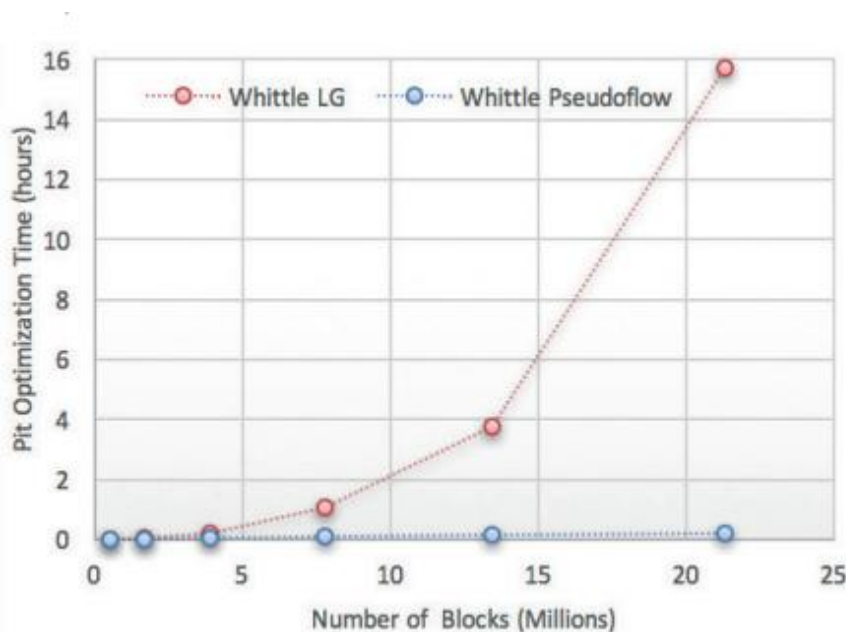
Σχήμα 6.4 α) Σύνδεση κόμβων με κόμβο source και κόμβο sink και απόδοση βαρών στα τόξα. β) Απόδοση τυχαίων τιμών λαμβάνοντας υπόψη τις μέγιστες επιτρεπόμενες και τα ισοζύγια ροής (Bai et al., 2017).

- 6) Εύρεση μέγιστης ροής. Για τον υπολογισμό της μέγιστης ροής ακολουθείται μια επαναληπτική διαδικασία όπου πραγματοποιούνται αλλαγές των ροών μέχρι να βρεθεί η μέγιστη ροή. Υπάρχουν πολλοί αλγόριθμοι υπολογισμού μέγιστης ροής αλλά ο αλγόριθμος της ψευδοροής έχει αποδειχθεί ότι είναι μία από τις πιο αποτελεσματικές μεθόδους μέχρι σήμερα σε σχέση με το χρόνο που απαιτείται για την εύρεση της μέγιστης τιμής.
- 7) Υπολογισμός των προς εξόρυξη μπλοκ Μόλις ο αλγόριθμος υπολογίσει την μέγιστη ροή (όπως φαίνεται στο Σχήμα 6.5α) τότε η λύση αυτή χρησιμοποιείται στον υπολογισμό του μέγιστου κλειστού υποσυνόλου, που αποτελεί την βέλτιστη εκσκαφή. Για να γίνει αυτό πρώτα εντοπίζονται όλα τα «κορεσμένα» τόξα (είναι εκείνα στα οποία η ροή είναι ίση με τη μέγιστη επιτρεπόμενη). Στο Σχήμα 6.5β φαίνονται ως μπλε διακεκομμένα τόξα. Αυτό διαχωρίζει τον κόμβο sink από τα μπλοκ των αγόνων υλικών (αρνητικές τιμές) που μπορούν να εξορυχθούν (μπλοκ f,g,h και i), και επίσης αποκόπτει τα τόξα από τον κόμβο source προς τα μπλοκ μεταλλεύματος (όπως το μπλοκ d) που δεν μπορούν να εξορυχθούν αφού οι οικονομικές τους αξίες δεν είναι αρκετές για να αντισταθμίσουν τις αρνητικές τιμές των υπερκείμενων αγόνων. Τέλος εντοπίζονται όλοι οι κόμβοι στους οποίους μπορεί να φτάσει ο κόμβος πηγής. Αυτοί οι κόμβοι αποτελούν το μέγιστο κλείσιμο (Bai, et al., 2017). Στο Σχήμα 6.5β φαίνεται ότι η βέλτιστη εκσκαφή περιλαμβάνει τα μπλοκ {b,c,f,g,h,i}.



Σχήμα 6.5 α) Λύση μέγιστης ροής β) εύρεση βέλτιστων ορίων με το γράφημα μέγιστης ροής (Bai et al, 2017).

Ο αλγόριθμος Pseudoflow είναι σημαντικά πιο γρήγορος για την εύρεση των βέλτιστων ορίων ενός ορυχείου σε σχέση με τον κλασσικό αλγόριθμο LG ιδιαίτερα σε περιπτώσεις όπου το ψηφιακό μοντέλο του κοιτάσματος αποτελείται από μεγάλο αριθμό μπλοκ. Στο Σχήμα 6.6 φαίνεται ο χρόνος που απαιτείται για την βελτιστοποίηση σε σχέση με τον αριθμό των μπλοκ για τις μεθόδους LG και Pseudoflow, όπως υλοποιούνται μέσω του εμπορικού λογισμικού Whittle LG και Whittle Pseudoflow της εταιρείας Geovia (GEOVIA Whittle – Pseudoflow Method for Pit Optimization – White Paper).



Σχήμα 6.6 Μεταβολή του χρόνου εύρεσης των βέλτιστων ορίων ορυχείου σε σχέση με τον αριθμό των μπλοκ για τις μεθόδους LG και Pseudoflow [Geovia, Whittle – Pseudoflow Method for Pit Optimization white paper].

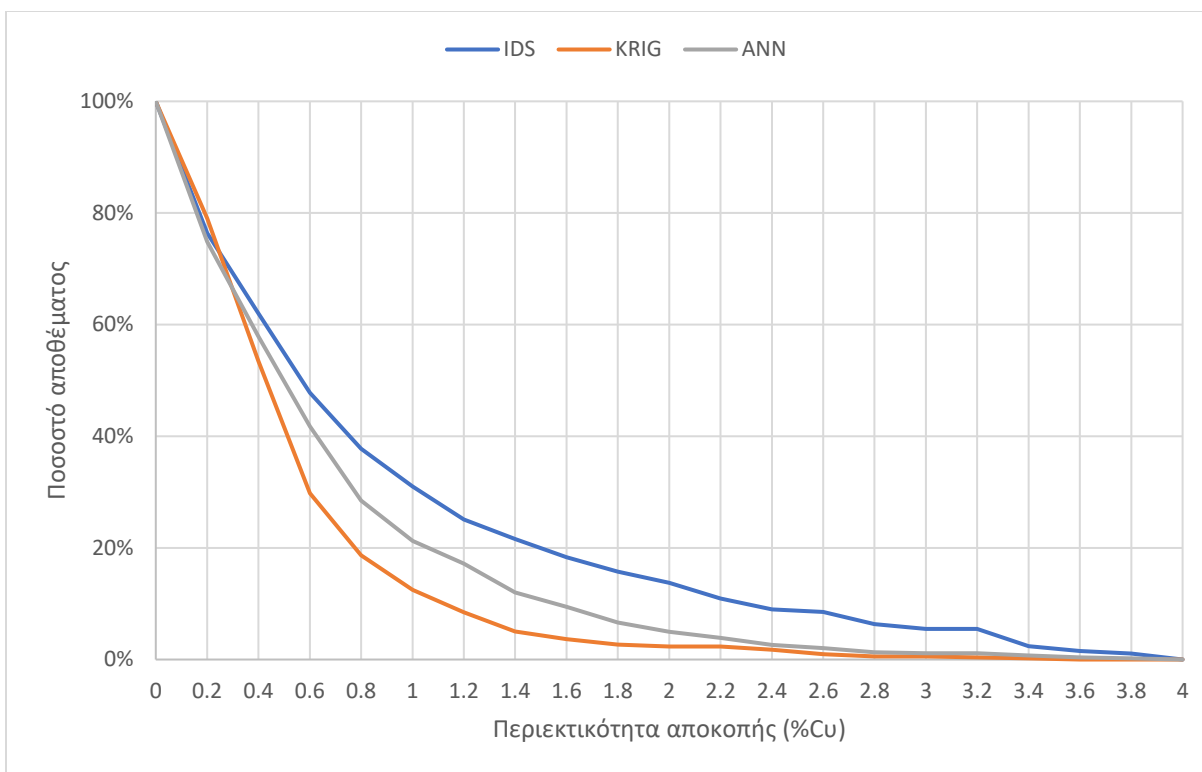
6.3 Εφαρμογή του αλγορίθμου Pseudoflow στο κοίτασμα Andina

Χρησιμοποιήθηκαν τρία ψηφιακά μοντέλα του κοιτάσματος Cu της Andina των 4800 μπλοκ έκαστο. Στο πρώτο οι περιεκτικότητες υπολογίστηκαν με τα νευρωνικά δίκτυα, στο δεύτερο με τη μέθοδο Kriging και στο τρίτο με τη μέθοδο IDS. Με βάση αυτά δημιουργήθηκαν στη συνέχεια 3 μοντέλα με τις οικονομικές αξίες των μπλοκ χρησιμοποιώντας τα ίδια οικονομικά στοιχεία (κόστος εξόρυξης, επεξεργασίας, τιμές μετάλλου κ.α.) και για τις τρεις περιπτώσεις. Τα μπλοκ με τις οικονομικές αξίες εισήχθησαν στο λογισμικό, στο οποίο έγιναν μετατροπές να θέτει την αξία του μπλοκ 0 όταν είναι μπλοκ αέρα και να βγάζει μια λίστα με τα εξορυχθέντα μπλοκ και δίνεται στο [Παράρτημα III](#) και υλοποιεί τον αλγόριθμο Pseudoflow σε Python για την εύρεση των βέλτιστων ορίων εκσκαφής. Στον [πίνακα 6.1](#) παρουσιάζονται τα αποτελέσματα του αλγορίθμου. Όπως φαίνεται από τα αποτελέσματα στην περίπτωση του νευρωνικού δικτύου ο αλγόριθμος Pseudoflow έχει υπολογίσει μεγαλύτερο αριθμό μπλοκ (1066) εντός της βέλτιστης εκσκαφής με συνολική οικονομική αξία 151.267.491\$. Για την περίπτωση της μεθόδου Kriging και IDS ο αλγόριθμος Pseudoflow έχει υπολογίσει μικρότερο αριθμό μπλοκ που θα εξορυχθούν, 1031 και 969 αντίστοιχα. Οι οικονομική αξία για τη μέθοδο Kriging (101.604.063\$) είναι η μικρότερη ενώ η οικονομική αξία για τη μέθοδο IDS (206.532.406\$) η μεγαλύτερη. Παρά το γεγονός ότι οι διαφορές στον αριθμό των μπλοκ εντός της βέλτιστης εκσκαφής δεν διαφέρουν σημαντικά, οι διαφορές ανάμεσα στην οικονομική αξία είναι σημαντικές καθώς και η μέση οικονομική αξία ανά μπλοκ, όπως φαίνεται στον πίνακα 5.1. Η διαφορά αυτή των οικονομικών τιμών που προκύπτει για τα μπλοκ της βέλτιστης εκσκαφής σχετίζεται άμεσα με τις εκτιμήσεις της περιεκτικότητας σε %Cu κάθε μεθόδου.

Πίνακας 6.1 Αποτελέσματα του αλγορίθμου Pseudoflow για την εύρεση των βέλτιστων ορίων εκσκαφής.

	Αριθμός των μπλοκ νευρωνικού δικτύου	Αριθμός των μπλοκ μεθόδου Kriging	Αριθμός των μπλοκ μεθόδου IDS
Σύνολο μπλοκ μοντέλου	4800	4800	4800
Εντός της βέλτιστης εκσκαφής	1066	1031	969
Οικονομική αξία των μπλοκ (\$)	151.267.491	101.604.063	206.532.406
Μέση οικονομική αξία μπλοκ (\$)	141.902	98.549	213.134

Στο [Σχήμα 6.7](#) παρουσιάζονται οι καμπύλες αποθεμάτων – ποιότητας για τα τεχνητά νευρωνικά δίκτυα, για τη μέθοδο Kriging και για τη μέθοδο IDS, όπως υπολογίστηκαν με βάση τις εκτιμήσεις των τριών μεθόδων. Παρατηρείται ότι για το ίδιο ποσοστό αποθέματος η μέθοδος Kriging δίνει τη μικρότερη περιεκτικότητα, ακολουθεί η μέθοδος των νευρωνικών δικτύων και τέλος η IDS που δίνει την μεγαλύτερη.



Σχήμα 6.7 Καμπύλη αποθεμάτων-ποιότητας με βάση τα νευρωνικά δίκτυα, τη μέθοδο Kriging, και τη μέθοδο IDS.

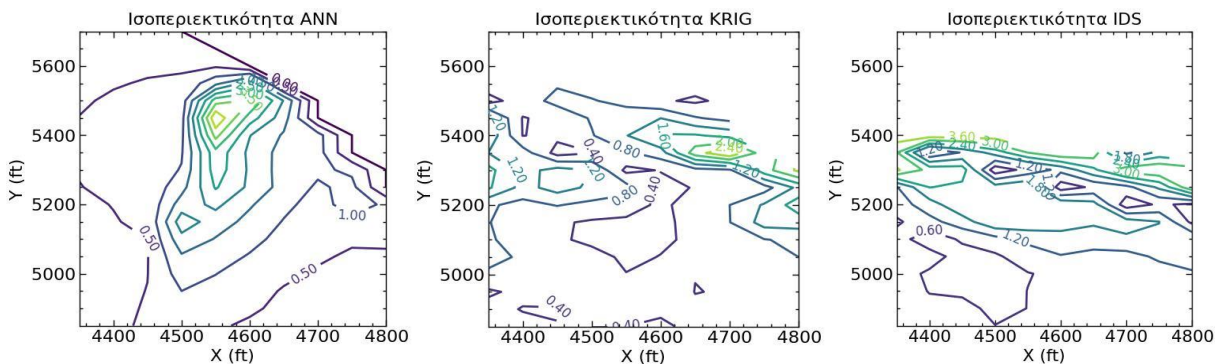
Στον [Πίνακα 5.2](#) δίνονται οι κλάσεις των περιεκτικοτήτων των μπλοκ της βέλτιστης εκσκαφής για τις τρεις μεθόδους. Παρατηρείται ότι στη μέθοδο IDS εμφανίζεται μεγαλύτερος αριθμός μπλοκ με μεσαία και υψηλή περιεκτικότητα ($\text{Cu}\% \geq 1.0$) σε σχέση με τις άλλες μεθόδους. Το γεγονός αυτό επιβεβαιώνει και τις διαφορές στις οικονομικές αξίες.

Πίνακας 6.2 Εκτίμηση αποθεμάτων ανά κατηγορία περιεκτικότητας Cu %.

Περιεκτικότητες	Αριθμός μπλοκ (νευρωνικά δίκτυα)	Αριθμός μπλοκ (Kriging)	Αριθμός μπλοκ (IDS)
$\text{Cu}\% < 1.0$	427	454	314
$1.0 \leq \text{Cu}\% \leq 2.0$	90	54	80
$\text{Cu}\% \geq 2.0$	30	12	74

Για την περαιτέρω διερεύνηση και σύγκριση των εκτιμήσεων των τριών μεθόδων κατασκευάστηκαν καμπύλες ισοπεριεκτικότητας της %Cu. Αναπτύχθηκε σχετικός κώδικας σε γλώσσα προγραμματισμού python μαζί με τις βιβλιοθήκες της για την δημιουργία χαρτών ισοπεριεκτικότητας ([Παράρτημα II](#)). Επιλέχθηκαν οι τιμές των περιεκτικοτήτων σε Cu για τα μπλοκ της βαθμίδας $Z=3075\text{ft}$ (σε αυτή την βαθμίδα υπήρχαν τα πιο πολλά μπλοκ) και κατασκευάστηκαν οι χάρτες ισοπεριεκτικότητας με βάση τις εκτιμήσεις από το νευρωνικό δίκτυο (ANN), τη μέθοδο Kriging και την μέθοδο IDS και δίνονται στο [Σχήμα 6.8](#). Παρατηρείται ότι, αν και οι τρεις μέθοδοι έχουν περίπου τις ίδιες μέγιστες τιμές διαφέρουν σημαντικά. Η μέθοδος των νευρωνικών δικτύων δίνει πολύ λιγότερα τοπικά ακρότατα από τις μεθόδους Kriging και IDS.

Φαίνεται λοιπόν ότι έχει χαρακτηριστικά γενικού εκτιμητή (Global estimator) σε σχέση με τις μεθόδους Kriging και IDS που είναι τοπικοί εκτιμητές.



Σχήμα 6.8 Χάρτες ισοπεριεκτικότητας Cu % που προέκυψαν από τις εκτιμήσεις των τεχνητών νευρωνικών δικτύων, Kriging .και IDS, αντίστοιχα για την βαθμίδα Z=3075ft.

7 Συμπεράσματα και προτάσεις

Χρησιμοποιήθηκαν νευρωνικά δίκτυα ευθείας προώθησης με οπισθοδιάδοση του σφάλματος (feedforward with backpropagation) αποτελούμενα από 3 εισόδους, μια έξοδο και ένα ενδιάμεσο επίπεδο νευρώνων για την εκτίμηση των αποθεμάτων και της ποιότητας κοιτάσματος χαλκού που πρόκειται να εξορυχθεί με υπαίθρια εκμετάλλευση. Αναπτύχθηκαν 3 διαφορετικές παραλλαγές νευρωνικών δικτύων. Στην πρώτη περίπτωση αναπτύχθηκε ένα νευρωνικό δίκτυο το οποίο εκπαιδεύτηκε με τη χρήση όλων των δειγμάτων των γεωτρήσεων. Στη συνέχεια αναπτύχθηκε ένα δεύτερο νευρωνικό δίκτυο που εκπαιδεύτηκε με δείγματα αναδειγματοληψίας με δυο κρυφά επίπεδα. Τέλος, αναπτύχθηκε ένα τρίτο πάλι με αναδειγματοληψία αλλά με ένα κρυφό επίπεδο. Τα αποτελέσματα που προέκυψαν συγκρίθηκαν με εκείνα της μεθόδου kriging και της IDS και έδειξαν ότι τα νευρωνικά δίκτυα που εκπαιδεύτηκαν με την αναδειγματοληψία (resampling) είχαν καλύτερα αποτελέσματα, ιδιαίτερα για εκτιμήσεις σε μεγαλύτερες τιμές περιεκτικότητας.

Στη συνέχεια κατασκευάστηκε το ψηφιακό μοντέλο του κοιτάσματος και έγινε ο υπολογισμός των οικονομικών αξιών των μπλοκ για τα μοντέλα που προέκυψαν από το νευρωνικό δίκτυο, τη μέθοδο Kriging και της μέθοδο IDS. Τα βέλτιστα όρια εκσκαφής υπολογίστηκαν με τον αλγόριθμο Pseudoflow. Τα αποτελέσματα έδειξαν ότι παρά το γεγονός ότι οι διαφορές στον αριθμό των μπλοκ εντός της βέλτιστης εκσκαφής δεν διαφέρουν σημαντικά για τις τρεις μεθόδους, οι διαφορές ανάμεσα στην οικονομική αξία είναι σημαντικές καθώς και η μέση οικονομική αξία ανά μπλοκ. Η μέθοδος Kriging έδωσε τη μικρότερη οικονομική αξία, στη συνέχεια ήταν τα νευρωνικά δίκτυα και τέλος την μεγαλύτερη την έδωσε η μέθοδος IDS. Η διαφορά αυτή των οικονομικών τιμών που προκύπτει για τα μπλοκ της βέλτιστης εκσκαφής σχετίζεται άμεσα με τις εκτιμήσεις της περιεκτικότητας σε %Cu της κάθε μεθόδου. Πιο συγκεκριμένα στη μέθοδο IDS εμφανίζεται ο μεγαλύτερος αριθμός μπλοκ με μεσαία και υψηλή περιεκτικότητα ($Cu\% \geq 1.0$) σε σχέση με τις άλλες μεθόδους. Επίσης από την κατασκευή των χαρτών ισοπεριεκτικότητας %Cu φάνηκε ότι η μέθοδος των νευρωνικών δικτύων έχει περισσότερο χαρακτηριστικά γενικού εκτιμητή. Για την βελτίωση της ακρίβειας των εκτιμήσεών του και για να του προσδοθούν περισσότερο χαρακτηριστικά τοπικού εκτιμητή προτείνονται τα παρακάτω για περαιτέρω διερεύνηση.

- Μεγαλύτερος αριθμός εισόδων νευρωνικού δικτύου. Εκτός από τις συντεταγμένες X,Y και Z θα μπορούσε να μπει σαν είσοδος και το είδος πετρώματος που βρίσκεται η μεταλλοφορία ή να μπει τιμή βαρύτητας για κάθε δείγμα ή και οι περιεκτικότητες γειτονικών δειγμάτων.
- Αυτοματοποίηση της διαδικασίας εύρεσης του βέλτιστου νευρωνικού δικτύου.
- Δημιουργία πολλαπλών κλάσεων αποθεμάτων ποιότητας για να μετατραπεί το πρόβλημα εκτίμησης των αποθεμάτων από πρόβλημα εκτίμησης σε πρόβλημα ταξινόμησης, όπου τα νευρωνικά δίκτυα υπερέχουν σε τέτοιου είδους προβλήματα.

Βιβλιογραφία

Διεθνής βιβλιογραφία

1. Arteaga Munoz, F., (2014). *The mining rate in open pit mine planning*. MPhil Thesis School of Mechanical and Mining Engineering The University of Queensland.
2. Bai, V.X., Turczynski, G., Baxter, N., Place, D., Sinclair-Ross, H., Ready, S., & Whittle, G.E. (2017). Geovia Whittle Pseudoflow Method for Pit Optimization. White Paper.
3. Bahrami, S, Ardejani, F & Baafi, E. (2016). *Application of artificial neural network coupled with genetic algorithm and simulated annealing to solve groundwater inflow problem to an advancing open pit mine*. Journal of Hydrology. 536. <https://doi.org/10.1016/j.jhydrol.2016.03.002>.
4. Chandran, B. G. & Hochbaum, D. S., (2009). *A Computational Study of the Pseudoflow and Push-Relabel Algorithms for the Maximum Flow Problem*. Operations Research, vol. 57(2), pp.358–376. <https://doi.org/10.1287/opre.1080.0572>.
5. Fadja, A.N, Lamma, E & Riguzzi, F. (2018). Vision Inspection with Neural Networks. *CEUR Workshop Proceedings*, 2272.
6. Goldberg, A. V. & Tarjan, R. E., (1988). *A new approach to the maximum-flow problem..* Journal of the Association for Computing Machinery, vol. 35(4), pp. 921–940. <https://doi.org/10.1145/48014.61051>.
7. Hustrulid, W. & Kuchta, M. (2006). *Open Pit Mine Planning & Design: CSMine software package*. p. 836, Balkema.
8. Jafrasteh, B., Fathianpour, N. & Suárez, A., (2018). *Comparison of machine learning methods for copper ore grade estimation* Comput Geosci 22(5), 1371–1388. <https://doi.org/10.1007/s10596-018-9758-0>.
9. İslamoğlu, E. (2016). *Forecasting Gold Prices Time Series by Using Joint Analysis and Separately Analysis*. 475-487.
10. Kapageridis, I., (2005). *Input space configuration effects in neural network-based grade estimation*. Computers and geosciences. 31(6), pp. 704-717. <https://doi.org/10.1016/j.cageo.2005.01.001>.
11. Kapageridis, I. K. & B. Denby, (1998). *Neural network modelling of ore grade spatial variability*. In International Conference on Artificial Neural Networks, Springer, London. pp. 209-214. https://doi.org/10.1007/978-1-4471-1599-1_28.
12. Kaplan, U.E. & Topal, E. A., (2020). *New Ore Grade Estimation Using Combine Machine Learning Algorithms*. Minerals, 10(10), 847. <https://doi.org/10.3390/min10100847>.

13. Lerchs, H.& Grossmann, I. F., (1965) *Optimum design of open-pit mines*. Transactions CIM, vol. LXVIII, pp. 17–24.
14. Li, X.L.; Xie, Y.L.; Guo, Q.J.; Li, L.H., (2010). *Adaptive ore grade estimation method for the mineral deposit evaluation*. *Math. Comput. Modelling*, 52(11-12), 1947–1956. <https://doi.org/10.1016/j.mcm.2010.04.018>.
15. Badel,M., Angorani, S., & Shariat Panahi, M. (2010). *The application of median indicator kriging and neural network in modeling mixed population in an iron ore deposit*. *Computers and geosciences*, 37(4), pp. 530-540. <https://doi.org/10.1016/j.cageo.2010.07.009>.
16. Morgan, N. & Bourlard, Hervé. (1989). *Generalization and Parameter Estimation in Feedforward Netws: Some Experiments*. *Neural Information Processing Systems* (1989).
17. Muir, D. C. W., (2004). *Pseudoflow, New Life for Lerchs-Grossmann Pit Optimisation*. in *Orebody Modelling and Strategic Mine Planning Conference*, Perth, Australia, 2004, vol. 14, pp. 97–104
18. Picard, J.C., (1976). *Maximal closure of a graph and applications to combinatorial problems*. *Management Science*, vol. 22(11), pp. 1268–1272. <https://doi.org/10.1287/mnsc.22.11.1268>.
19. Ramazan, Gençay & Min, Qi (2001). *Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging*. *Neural Networks, IEEE Transactions on*. 12. 726 - 734. 10.1109/72.935086.
20. Rumelhart, D., Hinton, G. & Williams, R., (1986). *Learning internal representations by error propagation*. *Parallel distributed processing*, 1(5), pp. 318-362.
21. Samanta, B. & S. Bandopadhyay, (2009). *Construction of a radial basis function network using an evolutionary algorithm for grade estimation in a placer gold deposit*. *Computers & Geosciences*, 35(8), pp. 1592-1602. <https://doi.org/10.1016/j.cageo.2009.01.006>.
22. Tahmasebi, P. & A. Hezarkhani, (2012). *A hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation*. *Comput Geosci.*, 42. <https://doi.org/10.1016/j.cageo.2012.02.004>.
23. Tutmez, B., (2009). *Use of hybrid intelligent computing in mineral resources evaluation*. *Applied soft computing*, 9(3), pp. 1023-1028. <https://doi.org/10.1016/j.asoc.2009.02.001>.
24. Xiping, W & Yingxin, Z. (1993). *Reserve estimation using neural network techniques*. *Computers & Geosciences*. 19, 567-575. [https://doi.org/10.1016/0098-3004\(93\)90082-G](https://doi.org/10.1016/0098-3004(93)90082-G).

Ελληνική βιβλιογραφία

1. Αργυράκης, Π., (2001). *Νευρωνικά Δίκτυα και Εφαρμογές*. Τόμος Β', Ελληνικό Ανοικτό Πανεπιστήμιο, Σχολή Θετικών Επιστημών και Τεχνολογίας.

2. Γαλετάκης, Μ., (1999). *Εφαρμογή τεχνητών νευρωνικών δικτύων για την επίλυση προβλημάτων που σχετίζονται με τον σχεδιασμό υπαιθρίων εκμεταλλεύσεων*. Ορυκτός Πλούτος, τεύχος 113, σελ. 47-56.
3. Γαλετάκης, Μ., Μαρκόπουλος, Θ., Χρηστίδης, Γ., Ρεπούσκου, Ε. και Αλεβίζος, Γ., (2000). *Πρόβλεψη ποιοτικών παραμέτρων αδρανών υλικών με χρήση τεχνητών νευρωνικών δικτύων*. 3ο Συνέδριο Ορυκτού Πλούτου, τεύχος 113, σελ. 47-56.
4. Γαλετάκης, Μ., (2012). *Σχεδιασμός εκμεταλλεύσεων με ηλεκτρονικό υπολογιστή*. Σημειώσεις Μαθήματος, Πολυτεχνείο Κρήτης, Σχολή Μηχανικών Ορυκτών Πόρων, Χανιά.
5. Χριστόπουλος, Δ., (2014). *Εφαρμοσμένη Γεωστατιστική*. Σημειώσεις Μαθήματος, Πολυτεχνείο Κρήτης, Χανιά.

Παράρτημα Ι: Κώδικας σε γλώσσα προγραμματισμού python, για την στατιστική και χωρική ανάλυση των γεωτρητικών δεδομένων

```
from skgstat import Variogram, OrdinaryKriging
import pandas as pd
import scienceplots
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import skew, kurtosis
plt.style.use(['science','notebook'])
from statistics import geometric_mean
from scipy import stats
plt.style.use('ggplot')
### diavazw ta dedomena
data = pd.read_csv('https://raw.githubusercontent.com/risi007/test123/main/test%20-%20Sheet1.csv')
cu = data['% Cu'].values
### synarthsh gia istogramma
def plot_histogram(data, num_bins=10):
    # Calculate histogram data
    hist, bins = np.histogram(data, bins=num_bins)
    bins = [float(b) for b in bins]
    bin_width = bins[1] - bins[0]
    interval = [f"{bins[i]:.2f}-{bins[i+1]:.2f}" for i in range(len(bins)-1)]
    frequency = hist
    relative_frequency = hist / len(data) * 100
    cumulative_frequency = np.cumsum(relative_frequency).astype(float)

    # Plot the histogram
    plt.hist(data, bins=num_bins)
    plt.xlabel('Cu')
    plt.ylabel('Count')

    # Print the results
    print("Interval\tFrequency\tRelative Freq.\tCumulative Freq.")
    for i in range(len(interval)):
        print(f"{interval[i]:10}\t{frequency[i]:5}\t{relative_frequency[i]:15.2f}\t{cumulative_frequency[i]:15.2f}")
    ")

### synarthsh gia statistika megethh
def calculate_statistics(data):
    n_samples = len(data)
    mean = np.mean(data)
```

```
variance = np.var(data, ddof=1)
std_dev = np.std(data, ddof=1)
skewness = skew(data)
kurtosis_value = kurtosis(data)

geo_mean = np.exp(np.mean(np.log(data)))
if np.isnan(geo_mean):
    geo_mean = 0

median = np.median(data)
trimmed_mean = np.mean(data[(data >= np.percentile(data, 90)) & (data <= np.percentile(data, 10))])
midrange = (np.min(data) + np.max(data)) / 2
mad = np.mean(np.abs(data - np.mean(data)))

print(f"Number of samples: {n_samples}")
print(f"Arithmetic average: {mean:.4f}")
print(f"Variance: {variance:.4f}")
print(f"Standard deviation: {std_dev:.4f}")
print(f"Coefficient of skewness: {skewness:.4f}")
print(f"Coefficient of kurtosis: {kurtosis_value:.4f}")
print(f"Geometric mean: {geo_mean:.4f}")
print(f"Median: {median:.4f}")
print(f"10% trim range mean: {trimmed_mean:.4f}")
print(f"Midrange: {midrange:.4f}")
print(f"Mean absolute deviation: {mad:.4f}")

##ypologismos variogrammatos
cell_size = 27.78
num_cells = 18

vg = Variogram(
    data[['X', 'Y', 'Z']].values,
    values = ln_cu,
    use_nugget=True,
    normalize=False,
    fit_method='trf',
    maxlag=cell_size*num_cells,
    n_lags=18,
    direction=0,
    azimuth=90,

    tolerance=90,
    bin_func='even',
    lag_func='linear',
    coordinates_type='euclidean',
    vertical_cutoff=None,
```

```
vertical_scaler=None,  
vertical_anisotropy=None,  
horizontal_cutoff=None,  
horizontal_scaler=50,  
horizontal_anisotropy=None,  
cutoff=np.inf,  
half_window_size=90,  
verbose=False,  
n_jobs=1,  
backend=None  
)  
print(vg)  
vg.plot(show=False, hist=False);
```

Παράρτημα II: Κώδικας σε γλώσσα προγραμματισμού python, για τα αναπτυχθέντα νευρωνικά δίκτυα και την δημιουργία χαρτών

```
from sklearn.preprocessing import StandardScaler
import tensorflow as tf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scienceplots
from sklearn.model_selection import train_test_split

### bibliothikes pou xreiazomai
data = pd.read_csv('https://raw.githubusercontent.com/risi007/test123/main/test%20-%20Sheet1.csv')
data = pd.DataFrame(data)

### diavazw ta dedomena
input = data.drop(("Cu"), axis=1)
output = data["Cu"]

### xwrizw ta dedomena se input kai output
output = output.astype('float64')
input_train, input_test, output_train, output_test = train_test_split(input, output, test_size=0.15, random_state=42)

### xwrizw ta dedomena se train kai test
input_train, input_val, output_train, output_val = train_test_split(input_train, output_train, test_size=0.18, random_state=42)

### xwrizw ta train dedomena se train kai validation
input_train_2 = pd.read_csv('https://raw.githubusercontent.com/risi007/test123/main/resample.csv')
### diavazw ta train meta apo resample
output_train_2 = input_train_2.loc[:, 'Cu']
input_train_2 = input_train_2.drop(("Cu"), axis=1)
### xwrizw ta dedomena tou resample se input kai output
output_train_2 = np.log(output_train_2)
output_val = np.log(output_val)
output_test = np.log(output_test)

### efarmozw ton fysiko logarithmo sta outputs
scaler = StandardScaler()
input_train_2 = scaler.fit_transform(input_train_2)
input_test = scaler.fit_transform(input_test)
input_val = scaler.fit_transform(input_val)
### efarmozw standardscaler sta input ((x-mean)/std)

### domh montelou
tf.random.set_seed(42)

model = tf.keras.Sequential([
```

```

tf.keras.layers.Dense(21, input_dim=3, activation='relu', name='hidden1'),
tf.keras.layers.Dense(14, activation='gelu', name='hidden2'),
tf.keras.layers.Dense(1, activation='linear', name='output')
])

model.compile(loss=tf.keras.losses.mse,
              optimizer=tf.keras.optimizers.Adam(learning_rate=0.01),
              )

hist = model.fit(input_train_2, output_train_2, validation_data=[input_val, output_val], epochs=32)
t = model.predict(input_test).flatten()
tr = model.predict(input_train_2).flatten()
### provlepsi toy monteloy gia ta test kai train dedomena
corr_matrix = np.corrcoef(output_test, t)
corr = corr_matrix[0,1]
R_squared = corr**2

print('R_squared = %0.2f' % R_squared)
### ypologismow tou R^2 gia ta test dedomena
corr_matrix_2 = np.corrcoef(output_train_2, tr)
corr_2 = corr_matrix_2[0,1]
R_squared_2 = corr_2**2

print('R_squared_2 = %0.2f' % R_squared_2)
### ypologismow tou R^2 gia ta train dedomena
plt.scatter(output_test, t)
plt.xlabel('True Values')
plt.ylabel('Predicted Values')
plt.axis('equal')
plt.axis('square')
z = np.polyfit(output_test, t, 1)
p = np.poly1d(z)
plt.plot(output_test, p(output_test), "r", lw=4)
R_squared = R_squared
__ = plt.text(-1.5, 1, 'R-squared = %0.2f' % R_squared)

_ = plt.plot([-100,100], [-100,100])
### diagramma true values - predicted values gia test dedomena
### kokkinh grammh trendline
### ble grammh Y=X
plt.scatter(output_train_2, tr)
plt.xlabel('True Values')
plt.ylabel('Predicted Values')

```

```

plt.axis('equal')
plt.axis('square')
z = np.polyfit(output_train_2, tr, 1)
p = np.poly1d(z)
plt.plot(output_train_2, p(output_train_2), "r", lw=4)
R_squared_2 = R_squared_2
__ = plt.text(-2.5, 1, 'R-squared_2 = %0.2f' % R_squared_2)
_ = plt.plot([-100,100], [-100,100])
### diagramma true values - predicted values gia train dedomena
### kokkinh grammh trendline
### ble grammh y=x
error = t - output_test
plt.hist(error)
plt.xlabel("Prediction Error")
_ = plt.ylabel("Count")
__ = plt.axvline(x=0, c='r')
### istogramma prediction error
loss = hist.history['loss']
epochs = range(1, len(loss) + 1)
plt.plot(epochs, loss, 'y', label="Training loss")
plt.title("Training loss")
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
### diagramma epochs-mse
df = pd.read_csv('https://raw.githubusercontent.com/risi007/test123/main/values4800.csv')
df = pd.DataFrame(df)
### diavazw ta dedomea tw n block
block_input = df
block_input = scaler.fit_transform(block_input)
### standard scaler sta input

block_output_ln = model.predict(block_input)
### provlepsi tw n dedomenwn
block_output = np.exp(block_output_ln)
### e^block_output in
block_output = pd.DataFrame(block_output)
filepath = 'preds8.xlsx'
block_output.to_excel(filepath, index=None)
### apothikeuw tis provlepseis

```

```
### xartes isoperiektikothtvn
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import rcParams

def plot_dataframe(df):
    data_2 = df.drop(('Z'), axis=1)
    Z = data_2.pivot_table(index='X', columns='Y', values=df.columns[3], aggfunc='mean').T
    X_unique = np.sort(data_2.X.unique())
    Y_unique = np.sort(data_2.Y.unique())
    X, Y = np.meshgrid(X_unique, Y_unique)
    rcParams['figure.figsize'] = 5, 5 # sets plot size
    fig = plt.figure()
    ax = fig.add_subplot(111)
    cp = ax.contour(X, Y, Z)
    cp.clabel(fontsize=12, inline=1, fmt='%.2f')
    plt.title('Contour plot')
    plt.xlabel('X (ft)')
    plt.ylabel('Y (ft)')
    plt.savefig('contour_plot.jpg')
```

Παράρτημα III: Κώδικας σε γλώσσα προγραμματισμού python, για τον αλγόριθμο υπολογισμού των βέλτιστων ορίων εκσκαφής

Στον αυθεντικό κώδικα <https://github.com/luisflarota/Ultimate-Pit-Limit---Pseudoflow>, έγιναν μερικές μετατροπές για να ανταποκρίνεται στις ανάγκες του προβλήματος που μελετήθηκε στην εργασία αυτή.

```
### prwto arxeio opou ginontai oi prakseis
from statistics import mode

import networkx as NetX
import numpy as np
import pandas as pd
import pseudoflow as pf
from plotly.subplots import make_subplots
from scipy import spatial

#transferring csv w/blockmodel to backend
class blockmodel(object):
    def __init__(self, bmodel):
        self.bmodel = bmodel

    def columns(self):
        return self.bmodel.columns.values.tolist()

    def summary(self, x, y, z, grade, density):
        def data():
            #See that index of x starts at 0 then... 1!
            self.x_axis = x
            self.y_axis = y
            self.z_axis = z
            self.gr_axis = grade
            self.ds_axis = density
            self.bmodel = self.bmodel.loc[:, [self.x_axis, self.y_axis, self.z_axis, self.gr_axis, self.ds_axis]]
            self.x = self.bmodel.loc[:, x]
            self.y = self.bmodel.loc[:, y]
            self.z = self.bmodel.loc[:, z]
            self.grade = self.bmodel.loc[:, grade]
            self.density = self.bmodel.loc[:, density]
            self.xlong = len(np.unique(self.x))
```



```

self.ylong = len(np.unique(self.y))
self.zlong = len(np.unique(self.z))
self.long = self.xlong * self.ylong * self.zlong
self.source = 0
self.minx = min(self.x)
self.maxx = max(self.x)
self.miny = min(self.y)
self.maxy = max(self.y)
self.minz = min(self.z)
self.maxz = max(self.z)
self.min_com = min(self.xlong, self.ylong)//2
self.sink = np.int(1 + self.long)
self.xunique = self.x.unique()
self.yunique = self.y.unique()
self.zunique = self.z.unique()
self.grademin = min(self.grade)
self.grademax = max(self.grade)
self.modex = int(mode(np.diff(np.unique(self.x))))
self.modey = int(mode(np.diff(np.unique(self.x))))
self.modez = int(mode(np.diff(np.unique(self.x))))

return data

#Say go or not given the #rows
def summary_2(self):
    return self.bmodel.shape[0]

#Outliers are in csv by purpose
def cleanning(self):
    self.outliers = []
    cols = [self.x_axis, self.y_axis, self.z_axis, self.gr_axis, self.ds_axis]

    mean_x, std_x = np.mean(self.x), np.std(self.x)
    mean_y, std_y = np.mean(self.y), np.std(self.y)
    mean_z, std_z = np.mean(self.z), np.std(self.z)

    #threshold for sd and mean
    threshold = 2
    for x,y,z,g,dens in zip(self.x, self.y, self.z, self.grade, self.density):
        x_score= (x - mean_x)/std_x
        y_score= (y - mean_y)/std_y
        z_score= (z - mean_z)/std_z
        x_up = x + self.modex
        x_low = x - self.modex

        y_up = y + self.modey

```

```

y_low = y - self.modey
z_up = z + self.modez
z_low = z - self.modez
if np.abs(x_score) > threshold or np.abs(y_score) > threshold or np.abs(z_score) > threshold:
    if type(g) == list:
        self.outliers.append([x,y,z, g[0],g[1], dens])
    else:
        self.outliers.append([x,y,z, g, dens])
elif (x_up not in self.xunique and x_low not in self.xunique) or (y_up not in self.yunique and y_low not
in self.yunique) or \
    (z_up not in self.zunique and z_low not in self.zunique):
    if type(g) == list:
        self.outliers.append([x,y,z, g[0],g[1], dens])
    else:
        self.outliers.append([x,y,z, g, dens])
out = np.array(self.outliers)

for a,b,c in zip(out[:,0], out[:,1], out[:,2]):
    delete = self.bmodel.loc[(self.x == a) & (self.y == b) & (self.z == c)]
    self.bmodel = self.bmodel.drop(delete.index)
return pd.DataFrame(self.outliers, columns = cols)

#Solving UPL problem
def upl(self, x,y,z,grade, density, mc, ic, pc, tc, mp, mr, prec):
    new_m = self.bmodel

    max_z = self.maxz
    if prec == '1-5 pattern':
        prec = 5
    elif prec == '1-9 pattern':
        prec = 9
    b_val = 'bvalue'
    c_off = 'cutoff'
    dic = 'dictator'
    node = 'node'
    new_m.loc[:,c_off] = np.array((mc + pc + (max_z - new_m.loc[:,z])*ic)/
        ((mp - tc) * mr * 22.04))

    new_m.loc[:,b_val] = np.where((new_m.loc[:,grade] > new_m.loc[:, c_off]),
        new_m.loc[:,'ton'] * (((mp - tc) * 22.04 * mr* new_m.loc[:,grade]) -

            pc - mc - (max_z - new_m.loc[:,z])*ic)
        , (0 - new_m.loc[:,'ton'] * mc))

    new_m.loc[new_m.loc[:,grade] == -2, b_val] = 0

```

```

new_m = new_m.reset_index(drop = True)
new_m.loc[:, node] = new_m.index + 1
new_m.loc[:, dic] = 0
#We get the new_m fot getting upl
#node:0, x:1, y:2, z:3, b_val:4, dic:5
new_m = new_m[[node, x, y, z, b_val, dic, density, grade]]
Graph = self.get_graph(nm = new_m, prec = prec)
breakpoints, cuts, info = pf.hpf(Graph, self.source, self.sink, const_cap="const", mult_cap="mult", lambda
aRange=[0], roundNegativeCapacity=False)

#Going over the cuts_items finding the nodes inside the resulting UPL.
Result_upl = {x:y for x, y in cuts.items() if y == [1] and x!=0}
InsideList = list(Result_upl.keys())

# Set all blocks as zero

for ind in InsideList:
    # Set blocks inside UPL as one
    new_m.loc[np.int(ind -1), dic] = 1

return new_m[new_m.loc[:,dic]==1]

def get_graph(self, nm, prec):
    Graph = NetX.DiGraph()
    col_compare = self.zunique[:::-1]
    if prec == 9:
        dista = (self.modex**2 + self.modey**2+ self.modez**2)**0.5
    elif prec == 5:
        dista = (self.modex**2 + self.modez**2)**0.5

    for i in range(self.min_com):
        #node:0, x:1, y:2, z:3, b_val:4, dic:5

        upper = np.array(nm[nm.iloc[:,3]== col_compare[i]])
        lower = np.array(nm[nm.iloc[:,3] == col_compare[i+1]])
        #Watch out with index block value
        self.CreateArcs(Graph = Graph, up = upper, low = lower, trigger = i, prec = prec, dist=dista)

        nm = nm[(nm.iloc[:,1]!= self.minx + i*self.modex) & (nm.iloc[:,1] != self.maxx - i*self.modex) & (nm
        .iloc[:,2] != self.miny + i*self.modey) & (nm.iloc[:,2] != self.maxy - i*self.modey)]
    return Graph

def CreateArcs(self,Graph,up, low, trigger, prec, dist):
    #Create internal arcs:

```

```

tree_upper = spatial.cKDTree(up[:,1:4])
mask = tree_upper.query_ball_point(low[:,1:4], r = dist + 0.01)
for _, g in enumerate(mask):
    if len(g) == prec:
        for reach in up[g][:,0]:
            Graph.add_edge(low[_][0], reach, const = 99e9, mult = 1)

#Create external arcs:
#if trigger+2 == min_compare:
# player = low
#else:
player = up
for node, capacity in zip(player[:,0], player[:,4]):
    cap_abs = np.absolute(np.around(capacity, decimals=2))
    if capacity < 0:
        Graph.add_edge(node, self.sink, const = cap_abs, mult = -1)
    else:
        Graph.add_edge(0, node, const = cap_abs, mult = 1)

### deytaro arxeio opou trexei h efarmogh
import re
from statistics import mode

import numpy as np
import pandas as pd
import plotly.graph_objects as go
import streamlit as st
from plotly.subplots import make_subplots
import os

import back

global colors
colors = ['#F9E79F', '#82E0AA', '#922B21', '#08056B', '33FFF4', 'FC33FF', 'FF8333', '070300']
#ranges = [[-1000000,0], [0,30000], [30000,1000000]]
rang_def = [[0,0.3], [0.3,0.6], [0.6,1], [1,10]]
def main():
    st.set_option('deprecation.showfileUploaderEncoding', False)

```

```

st.image('logo.jpg')

st.sidebar.write('Do you have a block model file?')
say_yes = st.sidebar.checkbox('Yes')
say_no = st.sidebar.checkbox('No')
if say_yes and say_no:
    st.sidebar.success('Please choose only one option')
elif say_no:
    #if st.sidebar.radio(",['Choose default file']"):
    st.sidebar.success('Loading default file')
    load_file('block_model_test.csv')
elif say_yes:
    file_ = st.sidebar.file_uploader('*Upload or drop the file:')
    if file_:
        load_file(file_)

def load_file(file_model):
    if file_model:
        separator = st.sidebar.radio('*csv file delimiter:', (',', ';'))
        if separator is not None:
            #We are calling the block model here:
            model = back.blockmodel(pd.read_csv(file_model, skipinitialspace= True, sep = separator))
            x = st.sidebar.selectbox('X coordinate:', model.columns())
            y = st.sidebar.selectbox('Y coordinate:', model.columns())
            z = st.sidebar.selectbox('Z coordinate:', model.columns())

            grade = st.sidebar.selectbox('Main Grade:', model.columns(), help='Please, units of grades must be in p
ercentage')
            density = st.sidebar.selectbox('Density:', model.columns())

            #Load archive:
            if st.sidebar.checkbox('Load'):
                try:
                    analize(model, x,y,z, grade, density)
                except:
                    st.info('Check your data carefully, i.e., units, coordinates, etc.')

def analize(model,x,y,z, grade,density):
    st.success('Based on your file:')
    model.summary(x,y,z, grade, density)()
    st.markdown('I got:')
    st.write('{} blocks in X'.format(model.xlong))
    st.write('{} blocks in Y'.format(model.ylong))
    st.write('{} blocks in Z'.format(model.zlong))

```

```

total_block_times = model.long
if total_block_times != model.summary_2():
    st.warning('Message: You need to have {:.0f} blocks. However, you got {:.0f} blocks'.format(total_block_times, model.bmodel.shape[0]))
    if st.checkbox('Check outliers in the block model:'):
        #Cleaning gets [outliers]
        outliers = model.cleanning()
        st.write('I am taking out the following blocks:')
        st.dataframe(outliers)
        print(outliers)
        #Asking one more time, if it wants to re-analyze de file
        analyze_again = st.radio('Analyze again?:', ['No', 'Yes'])
        if analyze_again == 'Yes':
            model = back.blockmodel(model.bmodel)
            analize(model, x,y,z, grade, density)

#If number of blocks = to unique values on x,y,z then:
elif total_block_times == model.summary_2():
    st.success('Awesome! Now, # blocks = # rows')
    #Calling visualizer that needs more development (blocks sizing)
    if st.checkbox('3D Block model visualization'):
        visualize_model(model, x,y,z, grade, density)

    if st.checkbox('Get Grade-Tonnage Distribution'):
        if type(grade) == str:
            grade = [grade]
        #Choosing main grade for the exercise
        select_mgrade = st.selectbox('Select your main grade:', grade)
        st.plotly_chart(call_grade_tonnage(model, select_mgrade, density))
        if st.checkbox('Get the Ultimate Pit Limit'):
            m_c = st.number_input('Mining Cost (USD/Ton.)', value = 2.5) #(USD/ Ton.)
            opt_increm = st.selectbox('Incremental Cost?', ['Yes', 'No'])
            if opt_increm == 'Yes':
                i_c = st.text_input('Incremental Cost (USD/ton.)', help='Does the mining cost increment while going deep? This is added to the mining cost while going 1 meter deep.',value = 0.01) #(USD/ Ton./ meter deep)
            else:
                i_c = 0
            p_c = st.number_input('Processing Cost (USD/ton.)', value = 10) #(USD/ Ton.)
            t_c = st.number_input('Treatment Cost (USD/lb.)', value = 0.15) #(USD/ lb.)
            m_p = st.number_input('Metal Price (USD/lb.)', value = 3.5) #(USD/ lb.)
            m_r = st.number_input('Metal Recovery (<1)', value = 0.9) #
            prec = st.selectbox('Select block precedences:', ['1-5 pattern', '1-9 pattern'], key='the_prec', help='45 degrees if 105 pattern, 40 degrees otherwise.')

            if st.button('Solve the Ultimate Pit Limit problem'):
                model_solve = model.upl(x,y,z, select_mgrade, density, m_c, float(i_c), p_c, t_c, m_p, m_r, prec)

```

```

st.write('Table of x, y, z, grade data:')
dataframe = model_solve
st.dataframe(dataframe)
csv = dataframe.to_csv(index=False).encode('utf-8')
st.download_button("Press to Download",csv,"Result_ANN_0.csv","text/csv",key='download-
csv')

return st.plotly_chart(visualize_upl(model_solve, x,y,z, select_mgrade, rang_def, colors),use_cont
ainer_width=True)

def visualize_model(model, x,y,z, grade, density):
    st.warning('Select the portion of the block model you want to see:')
    x_slider = st.slider('Range for X coordinates:', model.minx, model.maxx, (model.minx, model.minx + 5*mo
del.modex), model.modex)

    y_slider = st.slider('Range for Y coordinates:', model.miny, model.maxy, (model.miny, model.miny + 5*mo
del.modey), model.modey)
    z_slider = st.slider('Range for Z coordinates:', model.minz, model.maxz, (model.minz, model.minz + 5*mo
del.modez), model.modez)
    st.warning('How do you want to split grades?:')
    text_1 = 'Select ranges with only 1 decimal - i.e., 0.1, 0.2, 0.3, 2.0 (max. 8 numbers)'
    st.write(text_1)
    p_compile = re.compile(r'\d+\.\d+')
    min_grade, max_grade = min_max(model.bmodel, x,y,z, x_slider, y_slider, z_slider, grade)
    ranges = st.text_input('Minimun grade is: {}, and maximum one is: {}'.format(min_grade, max_grade))
    # Compile a pattern to capture float values
    # Convert strings to float
    # Asking to visualize
    ask_visualize = st.radio('Visualize:', ['No', 'Yes'])
    if ask_visualize == 'Yes':
        floats = [float(i) for i in p_compile.findall(ranges)]
        floats = list_maker(floats)
        st.plotly_chart(visualize(model.bmodel, x,y,z, x_slider, y_slider, z_slider, grade, floats, density, colors))

# Min_max to plot grade distribution
def min_max(model, x,y,z, x_slider, y_slider, z_slider, grade):
    for_plot = model.loc[((model.loc[:, x]>= x_slider[0]) & (model.loc[:,x] <= x_slider[1])) & \
        ((model.loc[:,y]>= y_slider[0]) & (model.loc[:,y] <= y_slider[1]))&\
        ((model.loc[:,z]>= z_slider[0]) & (model.loc[:,z] <= z_slider[1]))]
    min_ = round(min(for_plot.loc[:, grade]),3)
    max_ = round(max(for_plot.loc[:, grade]),3)
    return min_,max_

#Visualizing the whole, remember to use . as decimal separator
def visualize(model, x,y,z, x_slider, y_slider, z_slider, grade, floats, density, colors):
    i = 0
    for_plot = model.loc[((model.loc[:, x]>= x_slider[0]) & (model.loc[:,x] <= x_slider[1])) & \
        ((model.loc[:,y]>= y_slider[0]) & (model.loc[:,y] <= y_slider[1]))&\

```

```

        ((model.loc[:,z]>= z_slider[0]) & (model.loc[:,z] <= z_slider[1]))]
fig = go.Figure()
#set ranges with colors:
for value in floats:
    before = value[0]
    after = value[1]
    data_plot = for_plot.loc[(for_plot.loc[:,grade]>= before) & (for_plot.loc[:,grade]< after)]

    data_plot.loc[:, grade] = data_plot.loc[:, grade].round(3)
    data_plot.loc[:, 'g'] = '{}: '.format(grade) + data_plot.loc[:, grade].astype(str)
    #naming the plot on the legend
    name_legend = str(before) + " <=" + grade + "< " + str(after)
    fig.add_trace(go.Scatter3d(x = data_plot.loc[:, x], y = data_plot.loc[:, y], z = data_plot.loc[:, z], text= data_
plot.loc[:, 'g'], mode = 'markers' \
        , name = name_legend, marker = dict(color = colors[i], symbol = 'square', size = 4), showle
gend = True))
    i+=1
    title = '<b>Grade Distribution in the Block Model </b>'

fig.update_layout(margin = dict(r=100, t=25, b=40, l=100), title = title)

return fig

#Grade-tonnage curve
def call_grade_tonnage(model, grade, density):

    volume = model.modex * model.modey * model.modez
    modelo = model.bmodel
    modelo.loc[:, 'ton'] = volume * modelo.loc[:, density]
    modelo.loc[:, 'ton_g'] = modelo.loc[:, 'ton'] * modelo.loc[:, grade]/100

    cutoff_grade = list(np.arange(0,2, 0.05))
    ton_a_cutoff = []
    ton_cutfine = []
    for i in cutoff_grade:
        x = modelo[modelo.loc[:, grade]>=i]['ton'].sum()/1000000000
        y = modelo[modelo.loc[:, grade]>=i]['ton_g'].sum()/1000000000
        ton_a_cutoff.append(x)
        ton_cutfine.append(y)

    grade_ton_dist = pd.DataFrame({'Cutoff_grade': cutoff_grade, 'GTon_a_cutoff': ton_a_cutoff, 'GTon_cutfin
e': ton_cutfine})
    grade_ton_dist['Cut_Grade'] = grade_ton_dist['GTon_cutfine']*100/grade_ton_dist['GTon_a_cutoff']
    title = '<b>Grade - Tonnage Distribution for {} grade</b>'.format(grade)
    #8 We can plot it .. so now it may be familiar to you (PD: Forgot to grab some packages)
    fig1 = make_subplots(specs=[[{"secondary_y": True}]]))

```



```
fig1.add_trace(go.Scatter(x = grade_ton_dist['Cutoff_grade'], y = grade_ton_dist['GTon_a_cutoff'], name = '
Tonnage (Gtons.) vs. Cut-off grade (%)', secondary_y = False)
fig1.add_trace(go.Scatter(x = grade_ton_dist['Cutoff_grade'], y = grade_ton_dist['Cut_Grade'], name = 'Avg
. grade (%) vs Cut-off grade (%)', secondary_y = True)
fig1.update_xaxes(title_text = "<b>Cut-off grade (%)<b>")
fig1.update_yaxes(title_text="<b>Tonnage (GT.)</b>", secondary_y = False)
fig1.update_yaxes(title_text="<b>Avg. grade (%)</b>", secondary_y = True)
fig1.update_layout(title = title)
model.bmodel = modelo
return fig1
```

```
def list_maker(lista):
    list_ranges = []
    for i in range(len(lista)):
        if i == len(lista)-1:
            break
        else:
            bef = lista[i]
            after = lista[i+1]
            list_ranges.append([bef, after])
    return list_ranges
```

```
def visualize_upl(model, xcol, ycol, zcol, grade, ranges, colors):
    i = 0
    fig = make_subplots(rows = 9, cols = 2, specs = [[{'type': 'scatter3d', 'rowspan':6, 'colspan':2}, None],
                                                    [None, None],
                                                    [None, None],
                                                    [None, None],
                                                    [None, None],
                                                    [None, None],
                                                    [{'type': 'bar', 'rowspan':3, 'colspan':2}, None],
                                                    [None, None],
                                                    [None, None]])

    #set ranges with colors:

    for value in ranges:
        before = value[0]
        after = value[1]
        data_plot = model[(model.loc[:,grade]>= before) & (model.loc[:,grade]< after)]
        #This one is for the histogram
```

```

freq = data_plot.shape[0]
data_plot.loc[:,grade] = data_plot.loc[:,grade].round(3)
data_plot.loc[:, 'g'] = '{': '.format(grade) + data_plot.loc[:,grade].astype(str)
#naming the plot on the legend
name_legend = str(before) + " <=" + grade + "< " + str(after)
fig.add_trace(go.Scatter3d(x = data_plot.loc[:,xcol], y = data_plot.loc[:,ycol], z = data_plot.loc[:,zcol], tex
t= data_plot.loc[:, 'g'], mode = 'markers' \
, name = name_legend, marker = dict(color = colors[i], symbol = 'square', size = 4), showle
gend = True))
fig.add_trace(go.Bar(x = [name_legend], y = [freq], marker=dict(color = colors[i]), showlegend = False, n
ame = "", text = [freq], textposition = 'auto'), row = 7, col=1)
i+=1

#Include the precedence in the title!!!!
fig.update_layout(title = '<b>Ultimate Pit Limit</b>' + '<br>' + \
'<i>Undiscounted value of the pit: </i>' + '<b>USD {0:,.2f}</b>'.format(model.iloc[:,4].sum()),
annotations = [{ 'font': { 'size': 14 }, 'text': '<b>Histogram for grades within the ultimate pit limit</b>', '
x': 0.2, 'showarrow': False, 'y': 0.30,
'xanchor': 'center', 'xref': 'paper', 'yanchor': 'bottom', 'yref': 'paper'}],
yaxis2 = { 'anchor': 'x2', 'domain': [0.0, 0.5] },
legend = dict(x = 1.0, y = 1.0, font = dict(size = 13)))

fig.update_yaxes(title_text = 'Frequency', row = 9, col = 1)
fig.update_xaxes(title_text = 'Grade Ranges', row = 9, col = 1)
return fig
if __name__ == '__main__':
    main()

```