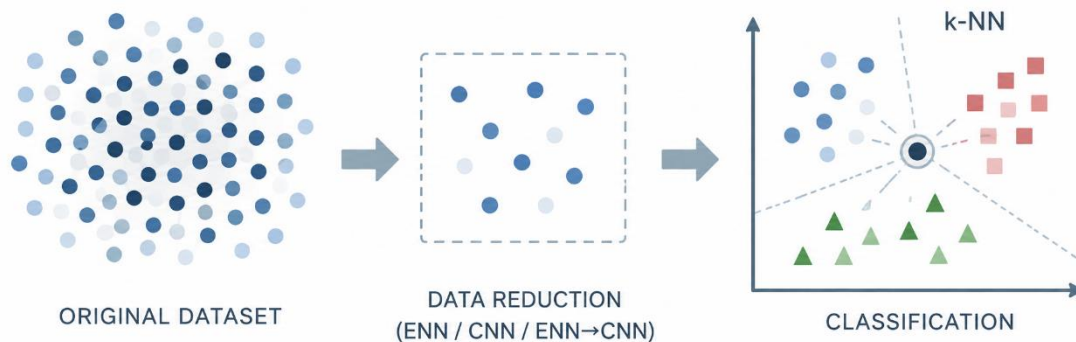


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
«ΑΞΙΟΛΟΓΗΣΗ ΤΕΧΝΙΚΩΝ ΜΕΙΩΣΗΣ
ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΤΟΥΣ ΑΛΓΟΡΙΘΜΟΥΣ
ΚΑΤΗΓΟΡΟΠΟΙΗΣΗΣ k-NN, Mutual k-NN,
Symmetric k-NN»



Του φοιτητή:
Γκούρα Ιωάννη
Αρ. Μητρώου: 134024

Επιβλέπων:
Ονοματεπώνυμο: Ουγιάρογλου
Στέφανος
Βαθμίδα: Επίκουρος Καθηγητής

Ημερομηνία 24/09/2026

Τίτλος Δ.Ε.: Αξιολόγηση τεχνικών μείωσης δεδομένων για τους αλγορίθμους κατηγοροποίησης k-NN, Mutual k-NN και Symmetric k-NN

Κωδικός Δ.Ε.: 26207

Ονοματεπώνυμο φοιτητή: Γκούρας Ιωάννης

Ονοματεπώνυμο εισηγητή: Ουγιάρογλου Στέφανος

Ημερομηνία ανάληψης Δ.Ε.: 23/03/2026

Ημερομηνία περάτωσης Δ.Ε.: 24/09/2026

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Γκούρα Ιωάννη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητα και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Στους γονείς μου για την αμέτρητη υπομονή, τη στήριξη και την ενθάρρυνσή τους σε όλη τη διάρκεια των σπουδών μου»

«Στον αδερφό μου και στη συντροφό μου που βρίσκονταν δίπλα μου και με στήριζαν σε όλη αυτή την προσπάθεια.»

Πρόλογος

Η επιλογή της συγκεκριμένης διπλωματικής εργασίας προέκυψε από το ενδιαφέρον μου για τον τομέα της ανάλυσης δεδομένων και της μηχανικής μάθησης και, ειδικότερα, για τους αλγορίθμους ταξινόμησης και τις δυνατότητες βελτίωσης της λειτουργίας τους. Η μελέτη των τεχνικών μείωσης δεδομένων σε συνδυασμό με τον k -NN και τις παραλλαγές του αποτέλεσε μια ευκαιρία να εξετάσω στην πράξη τη σχέση μεταξύ της ακρίβειας ταξινόμησης και του υπολογιστικού κόστους.

Μέσα από την εκπόνηση της εργασίας είχα τη δυνατότητα να εμβαθύνω στη λειτουργία των αλγορίθμων k -NN, Mutual k -NN και Symmetric k -NN, καθώς και στις τεχνικές μείωσης δεδομένων CNN και ENN. Παράλληλα, απέκτησα μεγαλύτερη εξοικείωση με την επεξεργασία και ανάλυση δεδομένων, την εκτέλεση πειραμάτων σε διαφορετικά σύνολα δεδομένων και την αξιολόγηση των αποτελεσμάτων με τη χρήση κατάλληλων μετρικών.

Η διαδικασία αυτή συνέβαλε στην καλύτερη κατανόηση τόσο της θεωρητικής πλευράς των μεθόδων που μελετήθηκαν όσο και της πρακτικής εφαρμογής τους και αποτέλεσε μια σημαντική εμπειρία στο πλαίσιο των σπουδών μου.

Περίληψη

Η παρούσα διπλωματική εργασία εξετάζει την εφαρμογή τεχνικών μείωσης δεδομένων σε αλγορίθμους ταξινόμησης που βασίζονται στους πλησιέστερους γείτονες. Συγκεκριμένα, μελετώνται ο κλασικός k-Nearest Neighbors (k-NN) και οι παραλλαγές Mutual k-NN και Symmetric k-NN, σε συνδυασμό με τις τεχνικές μείωσης δεδομένων Condensed Nearest Neighbor (CNN) και Edited Nearest Neighbor (ENN). Επιπλέον, εξετάζεται η διαδοχική εφαρμογή των δύο τεχνικών με τη μορφή ENN→CNN.

Στόχος της εργασίας είναι να διερευνηθεί κατά πόσο οι τεχνικές μείωσης δεδομένων μπορούν να εφαρμοστούν αποτελεσματικά όχι μόνο στον κλασικό k-NN, αλλά και στις παραλλαγές Mutual και Symmetric k-NN, καθώς και να αξιολογηθεί η επίδρασή τους στην απόδοση των αλγορίθμων.

Η πειραματική αξιολόγηση πραγματοποιήθηκε σε είκοσι σύνολα δεδομένων και για τιμές $k = 1, 3, 5, 7$. Οι μέθοδοι συγκρίθηκαν πριν και μετά τη μείωση των δεδομένων με βάση την ακρίβεια ταξινόμησης, τις μετρικές Precision, Recall και F1-Score, καθώς και τον απαιτούμενο CPU Time.

Τα αποτελέσματα έδειξαν ότι οι τεχνικές μείωσης δεδομένων μπορούν να περιορίσουν σημαντικά το μέγεθος του συνόλου εκπαίδευσης και το υπολογιστικό κόστος. Το ENN διατήρησε γενικά την ακρίβεια σε επίπεδα πιο κοντά σε εκείνα των αρχικών δεδομένων, ενώ το CNN και ιδιαίτερα ο συνδυασμός ENN→CNN πέτυχαν μεγαλύτερη μείωση των δεδομένων και του χρόνου εκτέλεσης, με μεγαλύτερη όμως επίδραση στην ακρίβεια σε ορισμένες περιπτώσεις. Συνολικά, τα αποτελέσματα αναδεικνύουν τον συμβιβασμό μεταξύ μείωσης δεδομένων, ακρίβειας ταξινόμησης και υπολογιστικού κόστους.

«Evaluation of Data Reduction Techniques for k-NN, Mutual k-NN, and Symmetric k-NN Classification Algorithms»

«Gkouras Ioannis»

Abstract

This thesis evaluates how data reduction affects the performance of nearest neighbor classifiers. We study the standard k-Nearest Neighbors (k-NN) algorithm alongside two variants, Mutual k-NN and Symmetric k-NN. For data reduction, we focus on Condensed Nearest Neighbor (CNN), Edited Nearest Neighbor (ENN), and their combined application (ENN followed by CNN).

Our main goal is to explore whether data reduction techniques can be applied effectively to Mutual and Symmetric k-NN, rather than just standard k-NN. We conducted extensive experiments on 20 benchmark datasets from the UCI and KEEL repositories. All tests were executed using a 5-fold cross-validation setup across four neighborhood values: $k = 1, 3, 5$, and 7 . We evaluated the models in terms of classification accuracy, precision, recall, F1-score, execution time (CPU time), and fallback rate for Mutual k-NN. To validate the differences, we used the Friedman statistical test and examined trade-offs using Skyline Pareto analysis.

The results show that data reduction noticeably cuts down training set size and computational overhead. ENN mostly cleans boundary noise and keeps classification accuracy close to original levels. On the other hand, CNN and the ENN $\rightarrow$ CNN pipeline achieves drastic sample reduction and faster execution times, but this occasionally comes at the cost of lower accuracy. Overall, Symmetric k-NN delivered the highest accuracy across several complex datasets (reaching over 99% on datasets like Monk), whereas k-NN trained on ENN $\rightarrow$ CNN subsets had the lowest execution time. These findings clearly outline the trade-offs between dataset size, predictive performance, and inference speed.

Ευχαριστίες

Θα ήθελα να εκφράσω τις ιδιαίτερες ευχαριστίες μου στον επιβλέποντα καθηγητή μου για την καθοδήγηση, την υποστήριξη και κυρίως για την ευκαιρία που μου έδωσε να ολοκληρώσω την πτυχιακή μου εργασία και τις σπουδές μου.

Περιεχόμενα

Πρόλογος.....	iv
Περίληψη.....	v
Abstract	vi
Ευχαριστίες	vii
Περιεχόμενα	viii
Κατάλογος Πινάκων.....	x
Συντομογραφίες.....	xi
Κεφάλαιο 1ο: Εισαγωγή	1
1.1 Εισαγωγή.....	1
1.2 Κατηγοροποίηση k εγγύτερων γειτόνων	1
1.3 Τεχνικές Μείωσης Δεδομένων	2
1.4 Κίνητρο της εργασίας.....	3
1.5 Συνεισφορά της εργασίας.....	3
1.6 Οργάνωση της εργασίας.....	4
Κεφάλαιο 2ο: Directed k-NN και παραλλαγές του.....	5
2.1 Directed k-NN	5
2.2 Mutual k-NN	6
2.3 Symmetric k-NN	8
Κεφάλαιο 3ο: Τεχνικές Μείωσης Δεδομένων.....	10
3.1 Είδη τεχνικών Μείωσης Δεδομένων	10
3.2 Condensed Nearest Neighbor	10
3.3 Edited Nearest Neighbor	12
3.4 Συνδυασμός ENN - CNN	13
Κεφάλαιο 4ο: Υλοποίηση Αλγορίθμων	16
4.1 Υλοποίηση k-NN.....	16
4.2 Υλοποιήσεις CNN - ENN.....	17
Κεφάλαιο 5ο: Πειραματική Μελέτη	19
5.1 Σύνολα δεδομένων	19
5.2 Εγκαθίδρυση πειραμάτων	24
5.3 Πειραματικές μετρήσεις προεπεξεργασίας.....	26
5.4 Πειραματικές μετρήσεις κατηγοριοποίησης	33
5.5 Στατιστική Ανάλυση	100

Κεφάλαιο 6ο:	Συμπεράσματα και Μελλοντικές επεκτάσεις.....	103
6.1	Συμπεράσματα.....	103
6.2	Μελλοντικές επεκτάσεις.....	103
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		105

Κατάλογος Πινάκων

Πίνακας 4.1: Κυριότερες βιβλιοθήκες Python και η χρήση τους στην υλοποίηση.....	
Πίνακας 5.1: Χαρακτηριστικά των είκοσι συνόλων δεδομένων.....	
Πίνακας 5.2: Αποτελέσματα μείωσης δεδομένων με τη χρήση της τεχνικής ENN.....	
Πίνακας 5.3: Αποτελέσματα μείωσης δεδομένων με τη χρήση της τεχνικής CNN.....	
Πίνακας 5.4: Αποτελέσματα μείωσης δεδομένων με τη συνδυαστική χρήση ENN → CNN.....	
Πίνακας 5.5: Σύγκριση αλγορίθμων (k-NN, Mutual k-NN, Symmetric k-NN) χωρίς μείωση δεδομένων για k = 1.....	
Πίνακας 5.6: Σύγκριση αλγορίθμων (k-NN, Mutual k-NN, Symmetric k-NN) χωρίς μείωση δεδομένων για k = 3.....	
Πίνακας 5.7: Σύγκριση αλγορίθμων (k-NN, Mutual k-NN, Symmetric k-NN) χωρίς μείωση δεδομένων για k = 5.....	
Πίνακας 5.8: Σύγκριση αλγορίθμων (k-NN, Mutual k-NN, Symmetric k-NN) χωρίς μείωση δεδομένων για k = 7.....	
Πίνακας 5.9: Σύγκριση αλγορίθμων μετά από χρήση CNN για k = 1.....	
Πίνακας 5.10: Σύγκριση αλγορίθμων μετά από χρήση CNN για k = 3.....	
Πίνακας 5.11: Σύγκριση αλγορίθμων μετά από χρήση CNN για k = 5.....	
Πίνακας 5.12: Σύγκριση αλγορίθμων μετά από χρήση CNN για k = 7.....	
Πίνακας 5.13: Σύγκριση αλγορίθμων μετά από χρήση ENN για k = 1.....	
Πίνακας 5.14: Σύγκριση αλγορίθμων μετά από χρήση ENN για k = 3.....	
Πίνακας 5.15: Σύγκριση αλγορίθμων μετά από χρήση ENN για k = 5.....	
Πίνακας 5.16: Σύγκριση αλγορίθμων μετά από χρήση ENN για k = 7.....	
Πίνακας 5.17: Σύγκριση αλγορίθμων μετά από συνδυαστική χρήση ENN → CNN για k = 1.....	
Πίνακας 5.18: Σύγκριση αλγορίθμων μετά από συνδυαστική χρήση ENN → CNN για k = 3.....	
Πίνακας 5.19: Σύγκριση αλγορίθμων μετά από συνδυαστική χρήση ENN → CNN για k = 5.....	
Πίνακας 5.20: Σύγκριση αλγορίθμων μετά από συνδυαστική χρήση ENN → CNN για k = 7.....	
Πίνακας 5.21: Friedman Test ως προς την ακρίβεια ταξινόμησης (Accuracy).....	
Πίνακας 5.22: Μέση κατάταξη μεθόδων (Friedman Test) ως προς την Accuracy.....	
Πίνακας 5.23: Friedman Test ως προς τον χρόνο εκτέλεσης (CPU Time).....	
Πίνακας 5.24: Μέση κατάταξη μεθόδων (Friedman Test) ως προς τον CPU Time.....	

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Η ανάπτυξη και η χρήση συστημάτων συλλογής δεδομένων έχει οδηγήσει στη δημιουργία ολοένα μεγαλύτερων και πιο σύνθετων συνόλων δεδομένων. Για τη διαχείριση και αξιοποίηση αυτών των δεδομένων, εμφανίστηκε η ανάγκη ανάπτυξης της Μηχανικής Μάθησης, ενός κλάδου της Τεχνητής Νοημοσύνης. Έναν από τους πιο θεμελιώδεις πυλώνες της Μηχανικής Μάθησης αποτελεί η κατηγοριοποίηση δεδομένων (data classification), δηλαδή η διαδικασία κατά την οποία ένα υπολογιστικό μοντέλο μαθαίνει να αναγνωρίζει μοτίβα σε ήδη υπάρχουσες πληροφορίες, ώστε να μπορέσει να ταξινομήσει νέα, άγνωστα δεδομένα σε συγκεκριμένες προκαθορισμένες κατηγορίες, πραγματοποιώντας με αυτόν τον τρόπο εποπτευόμενη μάθηση (Supervised Learning) [11].

Ανάλογα με τη φύση του προβλήματος και τον αριθμό των πιθανών επιλογών, η κατηγοριοποίηση χωρίζεται σε διάφορα είδη:

- **Δυαδική κατηγοριοποίηση (Binary Classification):** Η απλούστερη και πιο διαδεδομένη μορφή, όπου το αποτέλεσμα μπορεί να είναι μία από δύο επιλογές (ναι/όχι, 0/1). Το κλασικότερο παράδειγμα της συγκεκριμένης κατηγοριοποίησης είναι το φιλτράρισμα ανεπιθύμητων μηνυμάτων ηλεκτρονικού ταχυδρομείου (spam filtering). Το μοντέλο αναλύει τον αποστολέα, το κείμενο και αποφασίζει εάν κατηγοριοποιείται ως spam ή ως εισερχόμενο.
- **Κατηγοριοποίηση Πολλαπλών Κλάσεων (Multi-class Classification):** Η διαφορά αυτού του ταξινομητή είναι ότι κάθε δεδομένο μπορεί να ανήκει αποκλειστικά σε μία κλάση, ωστόσο υπάρχουν παραπάνω από δύο διαθέσιμες κλάσεις. Για παράδειγμα, η ταξινόμηση αιτημάτων ενός κέντρου υποστήριξης πελατών σε κατηγορίες όπως «Τεχνικό πρόβλημα», «Νέα συνδρομή» κ.λπ.
- **Κατηγοριοποίηση Πολλαπλών Ετικετών (Multi-label Classification):** Με τον ταξινομητή αυτό, ένα δεδομένο μπορεί να ανήκει σε παραπάνω από μία κλάσεις, με χαρακτηριστικότερο παράδειγμα ένα ειδησεογραφικό άρθρο που μπορεί να χαρακτηριστεί ταυτόχρονα ως οικονομικό αλλά και ως τεχνολογικό.

Για να υλοποιήσουμε αυτή τη διαδικασία, χρησιμοποιούμε αλγορίθμους ταξινόμησης (classifiers), με τον καθένα να έχει τα δικά του πλεονεκτήματα. Κάποια από τα μοντέλα είναι ο Naive Bayes, ένα από τα απλούστερα μοντέλα που προσφέρει ταχύτητα και υψηλή απόδοση σε δεδομένα κειμένου, και ο k-Nearest Neighbors (k-NN), ο οποίος βασίζεται στην απόσταση και προσφέρει ευελιξία σε πιο σύνθετα δεδομένα, ενώ για ακόμη πιο απαιτητικές περιπτώσεις χρησιμοποιούνται SVM (Support Vector Machines) ή Νευρωνικά Δίκτυα.

1.2 Κατηγοριοποίηση k εγγύτερων γειτόνων

Ο αλγόριθμος k-Nearest Neighbors (k-NN) [1,2] είναι ένας μη παραμετρικός αλγόριθμος κατηγοριοποίησης, επιβλεπόμενης μάθησης (supervised learning) και ευρέως χρησιμοποιούμενος [8] που βασίζεται στην έννοια της γειτνίασης μεταξύ των δεδομένων δηλαδή στην απλή παραδοχή ότι όμοια δεδομένα βρίσκονται συνήθως κοντά το ένα στο άλλο στον χώρο.

Εν αντιθέσει με άλλους αλγορίθμους ο k -NN δεν εκτελεί κάποια φάση εκπαίδευσης ώστε να δημιουργήσει ένα γενικευμένο μοντέλο, για αυτό το λόγο ανήκει στην κατηγορία τεμπέλικης μάθησης (lazy learning), αλλά αποθηκεύει ολόκληρο το σύνολο δεδομένων εκπαίδευσης. Όταν εμφανίζεται ένα νέο, άγνωστο δείγμα ο αλγόριθμος υπολογίζει την απόσταση του νέου σημείου από όλα τα αποθηκευμένα σημεία του συνόλου εκπαίδευσης χρησιμοποιώντας συνήθως την Ευκλείδεια απόσταση είτε άλλες μετρικές όπως τη Manhattan ή τη Minkowski. Αφού υπολογίσει τις αποστάσεις και εντοπίσει τα k κοντινότερα σημεία από το νέο δείγμα τότε το ταξινομεί στην αντίστοιχη κλάση αυτών των k γειτόνων (πλειοψηφική ψηφοφορία).

Η επιλογή της παραμέτρου k είναι ιδιαίτερα σημαντική για τον αλγόριθμο. Με τη χρήση πολύ μικρού k το μοντέλο γίνεται εξαιρετικά ευαίσθητο στο θόρυβο (noise) και στις ακραίες τιμές (outliers) οδηγώντας σε υπερπροσαρμογή (overfitting) ενώ με τη χρήση ενός πολύ μεγάλου k τα όρια των κλάσεων εξομαλύνονται υπερβολικά συμπεριλαμβάνοντας γειτονιές από μακρινές κλάσεις και οδηγώντας σε underfitting.

Πλεονεκτήματα:

Στα θετικά του αλγορίθμου συγκαταλέγεται η απλότητα και η διαφάνεια του. Είναι εύκολος στην κατανόηση και την υλοποίηση και δεν απαιτεί περίπλοκες υποθέσεις σχετικά με την κατανομή των δεδομένων (μη παραμετρικός). Επιπλέον το γεγονός ότι δεν κάνει καμία υπόθεση για την κατανομή των δεδομένων τον καθιστά ευέλικτο σε μη γραμμικά διαχωρίσιμα προβλήματα. Ενώ ακόμη, η προσθήκη νέων δεδομένων είναι άμεση απλά προσθέτοντας τα στο σύνολο εκπαίδευσης αποφεύγοντας έτσι την επανεκπαίδευση του μοντέλου.

Μειονεκτήματα και περιορισμοί:

Το κύριο μειονέκτημα του αλγορίθμου είναι η υπολογιστική ισχύς που απαιτείται κατά την πρόβλεψη ώστε να υπολογίσει τις αποστάσεις από όλα τα σημεία εκπαίδευσης κάτι που τον καθιστά αργό σε μεγάλα σύνολα δεδομένων ενώ ταυτόχρονα, απαιτεί μεγάλη μνήμη για την αποθήκευση ολόκληρου του συνόλου εκπαίδευσης. Ακόμη, χαρακτηριστικά με μεγάλο εύρος τιμών κυριαρχούν στον υπολογισμό των αποστάσεων κάτι που καθιστά την κανονικοποίηση (normalization) απαραίτητο βήμα προς την εφαρμογή του. Επίσης, σε περιπτώσεις που σημεία απέχουν σχεδόν εξίσου μεταξύ τους, η έννοια της απόστασης χάνει την ικανότητα της διάκρισης εμφανίζοντας έτσι την κατάρρα της διαστατικότητας (Curse of Dimensionality).

1.3 Τεχνικές Μείωσης Δεδομένων

Ένας άμεσος τρόπος αντιμετώπισης των αδυναμιών του k -NN είναι οι τεχνικές μείωσης δεδομένων με τη μείωση του αριθμού των δειγμάτων που χρησιμοποιούνται ως σημεία αναφοράς κατά την ταξινόμηση. Έτσι, αντί να συγκρίνεται κάθε νέο σημείο με χιλιάδες παραδείγματα, το μοντέλο διατηρεί μόνο ένα επιλεγμένο υποσύνολο προσπερνώντας τις αδυναμίες της υψηλής απαιτήσης μνήμης και της αργής πρόβλεψης.

Στην παρούσα εργασία το ενδιαφέρον επικεντρώνεται σε δύο κλασσικούς αλλά συμπληρωματικούς αλγορίθμους: τον Condensed Nearest Neighbor (CNN) [3] και τον Edited Nearest Neighbor (ENN) [4] (τεχνικές επιλογής προτύπων [9,10]). Ο αλγόριθμος CNN [3] (Hart, 1968) έχει ως βασικό στόχο τη δημιουργία ενός μικρότερου, πιο συμπυκνωμένου (condensed) συνόλου αντιπροσωπευτικών δειγμάτων και λειτουργεί με τη λογική ότι τα εσωτερικά σημεία μίας κλάσης είναι πλεονάζοντα καθώς δεν επηρεάζουν τη διαμόρφωση των ορίων διαχωρισμού των κλάσεων. Από την άλλη, ο ENN [4] (Wilson,

1972) ακολουθεί διαφορετική κατεύθυνση καθώς επικεντρώνεται στην επεξεργασία (editing) και καθαρισμό (noise) των δεδομένων και λειτουργεί με τη φιλοσοφία ότι κάθε σημείο εκπαίδευσης ταξινομείται με βάση τους k εγγύτερους γειτόνους του. Εάν η κλάση του σημείου δεν συμφωνεί με την πλειοψηφία των γειτόνων του, τότε το σημείο αφαιρείται είτε ως θόρυβος (noise) είτε ως ακραία τιμή (outlier).

Επειδή, όπως προαναφέρθηκε, ο CNN είναι ευαίσθητος στον θόρυβο είναι σύνηθες να εφαρμόζεται πρώτα ο ENN για να καθαρίσει αυτά τα δεδομένα και έπειτα ο CNN για την συμπύκνωση αυτών ώστε να έχουμε το καλύτερο δυνατό αποτέλεσμα από τις τεχνικές μείωσης δεδομένων.

Με τη χρήση των αλγορίθμων μείωσης δεδομένων ο k -NN καταφέρνει να μειώσει την υψηλή του υπολογιστική κατανάλωση καθώς και τον θόρυβο, επιτυγχάνοντας με αυτό τον τρόπο μία πιο αποδοτική και αξιόπιστη διαδικασία αναζήτησης εγγύτερων γειτόνων.

1.4 Κίνητρο της εργασίας

Γνωρίζουμε το πως λειτουργεί ο κλασσικός αλγόριθμος k -NN, την αξιοπιστία του και την απλότητα του καθώς και πως αυτός συμπεριφέρεται με τη χρήση των αλγορίθμων μείωσης δεδομένων. Αυτό όμως που θέλουμε να αναλύσουμε και αποτελεί το κίνητρο της παρούσας εργασίας, είναι με ποιο τρόπο συμπεριφέρονται στους αντίστοιχους αλγορίθμους οι παραλλαγές του κλασσικού k -NN και πιο συγκεκριμένα ο Mutual k -NN [5,6] και ο Symmetric k -NN [6,7]. Ο Mutual k -NN βασίζεται στην αμοιβαία γειτνίαση ενώ ο Symmetric k -NN λαμβάνει υπόψη τόσο τις αμοιβαίες όσο και τις μονόπλευρες σχέσεις γειτνίασης. Επομένως προκύπτει το ενδιαφέρον να μελετήσουμε εάν οι αλγόριθμοι μείωσης δεδομένων συμπεριφέρονται με τον ίδιο τρόπο όταν εφαρμόζονται και στις παραλλαγές του k -NN όπως με τον κλασσικό αλγόριθμο καθώς επίσης και πως συμπεριφέρονται, ο κλασσικός k -NN και οι παραλλαγές του, εάν εφαρμόσουμε και τις δύο τεχνικές μείωσης δεδομένων (ENN και CNN).

1.5 Συνεισφορά της εργασίας

Η κύρια συνεισφορά της παρούσας εργασίας εντοπίζεται αρχικά στην ανάπτυξη και υλοποίηση των αλγορίθμων k -NN, Mutual k -NN και Symmetric k -NN από την αρχή με έμφαση στην αποδοτικότητα του κώδικα με σκοπό την πλήρη κατανόηση της λειτουργίας τους. Έπειτα, πραγματοποιήθηκε εκτενής πειραματική μελέτη σύγκρισης όλων των αλγορίθμων με τη χρήση 20 γνωστών datasets και με τη χρήση τεσσάρων διαφορετικών τιμών k (1, 3, 5, 7). Και τέλος, η πειραματική μελέτη συνεχίστηκε με τη χρήση τεχνικών μείωσης δεδομένων και πιο συγκεκριμένα των ENN και CNN καθώς και συνδυασμός αυτών (πρώτα ENN και στη συνέχεια CNN) και ολοκληρώθηκε αφού ξανά εφαρμόσαμε τους τρεις αρχικούς αλγορίθμους στα νέα μειωμένα δεδομένα και για τις τρεις περιπτώσεις.

1.6 Οργάνωση της εργασίας

Η παρούσα πτυχιακή εργασία οργανώνεται σε έξι κεφάλαια, τα οποία δομούνται ως εξής:

Το Κεφάλαιο 1 περιλαμβάνει μία εισαγωγή στην κατηγοροποίηση δεδομένων, του αλγορίθμου k -NN, τη λειτουργία και τα μειονεκτήματά του. Επιπλέον, αναφέρονται οι τεχνικές μείωσης δεδομένων και διατυπώνονται τα κίνητρα και η συνεισφορά της εργασίας.

Κεφάλαιο 1: Εισαγωγή

Στο Κεφάλαιο 2 παρουσιάζεται ο κλασσικός αλγόριθμος k-NN καθώς και οι παραλλαγές αυτού δηλαδή ο Mutual k-NN και ο Symmetric k-NN μαζί με τις υλοποιήσεις τους.

Στο Κεφάλαιο 3 περιγράφονται οι τεχνικές μείωσης δεδομένων καθώς και συνδυασμός εφαρμογής τους.

Στο Κεφάλαιο 4 παρουσιάζεται η υλοποίηση των τεχνικών μείωσης δεδομένων.

Το Κεφάλαιο 5 παρουσιάζει τις πρωτότυπες εφαρμογές της εργασίας καθώς αναλύει την πειραματική μελέτη και την στατιστική ανάλυση των αποτελεσμάτων.

Στο Κεφάλαιο 6 περιγράφονται τα συμπεράσματα της εργασίας καθώς και πιθανές μελλοντικές επεκτάσεις αυτής.

Κεφάλαιο 2ο: Directed k-NN και παραλλαγές του

Οι αλγόριθμοι που βασίζονται στους πλησιέστερους γείτονες αποτελούν μία απλή και ευρέως χρησιμοποιούμενη προσέγγιση για προβλήματα κατηγοριοποίησης. Η βασική τους ιδέα είναι ότι παρατηρήσεις που βρίσκονται κοντά μεταξύ τους στον χώρο των χαρακτηριστικών είναι πιθανό να ανήκουν στην ίδια κλάση.

Στο παρόν κεφάλαιο παρουσιάζονται οι τρεις αλγόριθμοι κατηγοριοποίησης που χρησιμοποιήθηκαν στην πειραματική μελέτη της εργασίας. Αρχικά παρουσιάζεται ο κλασικός k-Nearest Neighbors (k-NN) και στη συνέχεια οι δύο παραλλαγές του, Mutual k-NN και Symmetric k-NN.

2.1 Directed k-NN

Ο k-Nearest Neighbors (k-NN) είναι ένας αλγόριθμος επιβλεπόμενης μάθησης που μπορεί να χρησιμοποιηθεί για την κατηγοριοποίηση νέων παρατηρήσεων. Η λειτουργία του βασίζεται στην απόσταση μεταξύ των παρατηρήσεων και στην υπόθεση ότι παρατηρήσεις που βρίσκονται κοντά μεταξύ τους είναι πιθανό να έχουν παρόμοια χαρακτηριστικά και να ανήκουν στην ίδια κλάση.

Σε αντίθεση με άλλους αλγορίθμους μηχανικής μάθησης, ο k-NN δεν δημιουργεί κατά τη φάση της εκπαίδευσης κάποιο συγκεκριμένο μοντέλο. Αντίθετα, διατηρεί τις παρατηρήσεις του συνόλου εκπαίδευσης και τις χρησιμοποιεί άμεσα όταν χρειάζεται να ταξινομήσει μία νέα παρατήρηση.

Έστω μία νέα παρατήρηση xx , της οποίας η κλάση είναι άγνωστη. Αρχικά υπολογίζεται η απόσταση της xx από τις παρατηρήσεις του συνόλου εκπαίδευσης. Στη συνέχεια, οι αποστάσεις ταξινομούνται και επιλέγονται οι k παρατηρήσεις με τη μικρότερη απόσταση από την xx . Οι παρατηρήσεις αυτές αποτελούν τους k πλησιέστερους γείτονες της νέας παρατήρησης.

Αν συμβολίσουμε το σύνολο των k πλησιέστερων γειτόνων της x ως:

$$N_k(x)$$

τότε μόνο οι παρατηρήσεις που ανήκουν στο $N_k(x)$ συμμετέχουν στην ψηφοφορία για την κλάση της xx . Κάθε γείτονας ψηφίζει σύμφωνα με τη δική του κλάση και η νέα παρατήρηση ταξινομείται στην κλάση που συγκεντρώνει τις περισσότερες ψήφους.

Αλγόριθμος 1 – Directed k-NN

```
Require: Εκπαιδευτικό σύνολο  $D_{train} = \{(x_i, y_i)\}_{i=1}^n$ ,  
         σύνολο ελέγχου  $D_{test}$ , αριθμός γειτόνων  $k$   
Ensure: Προβλεπόμενη κατηγορία  $\hat{y}$  για κάθε δείγμα του  $D_{test}$   
1: for  $i = 1$  to  $n$  do  
2:   Υπολόγισε τις αποστάσεις  $d(x_i, x_j)$  για όλα τα  $j \neq i$   
3:   Ταξινόμησε τους γείτονες κατά αύξουσα απόσταση  
4:   Αποθήκευσε τις απαιτούμενες πληροφορίες γειτονιάς  
5: end for  
6: Φάση Ταξινόμησης:  
7: for κάθε δείγμα  $x \in D_{test}$  do  
8:   Βρες τους  $k$  πλησιέστερους γείτονες  $N_k(x)$   
9:   Συγκέντρωσε τις κατηγορίες των  $k$  γειτόνων  
10:   $\hat{y}(x) \leftarrow \text{MajorityVote}(\{y_i : x_i \in N_k(x)\})$   
11:  Αποθήκευσε την προβλεπόμενη κατηγορία  $\hat{y}(x)$   
12: end for  
13: return  $\hat{Y}_{pred}$ 
```

2.2 Mutual k-NN

Ο Mutual k-Nearest Neighbors (Mutual k-NN) [5,6] αποτελεί παραλλαγή του κλασικού k-NN, στην οποία χρησιμοποιείται μία αυστηρότερη συνθήκη για τον καθορισμό της γειτονιάς. Στον κλασικό k-NN, για να συμμετέχει μία παρατήρηση y στην ταξινόμηση μιας νέας παρατήρησης x , αρκεί η y να βρίσκεται στους k πλησιέστερους γείτονες της x :

$$Y \in Nk(x)$$

Στον Mutual k-NN αυτό δεν είναι αρκετό. Απαιτείται ταυτόχρονα και η αντίστροφη σχέση. Δηλαδή, η νέα παρατήρηση x πρέπει επίσης να βρίσκεται στους k πλησιέστερους γείτονες της y .

Επομένως, οι δύο παρατηρήσεις θεωρούνται Mutual Neighbors όταν:

$$y \in Nk(x) \text{ και } x \in Nk(y)$$

Με απλά λόγια, η σχέση γειτονιάς πρέπει να είναι αμοιβαία.

Κατά την ταξινόμηση μιας νέας παρατήρησης x , αρχικά εντοπίζονται οι k πλησιέστεροι γείτονές της. Στη συνέχεια, για κάθε έναν από αυτούς ελέγχεται αν και η x βρίσκεται αντίστοιχα στη δική του γειτονιά. Μόνο οι παρατηρήσεις για τις οποίες ικανοποιούνται και οι δύο συνθήκες συμμετέχουν στην τελική ψηφοφορία.

Για να γίνει πιο κατανοητή η λειτουργία του Mutual k-NN, θεωρούμε και πάλι μία νέα παρατήρηση και τιμή $=3$. Αρχικά εντοπίζονται οι τρεις πλησιέστεροι γείτονές της, όπως ακριβώς και στον κλασικό k-NN.

Στη συνέχεια, όμως, πραγματοποιείται ένας επιπλέον έλεγχος. Για καθέναν από τους τρεις γείτονες εξετάζεται αν και η νέα παρατήρηση βρίσκεται μέσα στους δικούς του τρεις πλησιέστερους γείτονες. Έστω ότι αυτό ισχύει μόνο για τον πρώτο και τον τρίτο γείτονα, ενώ για τον δεύτερο δεν ισχύει. Στην περίπτωση αυτή, μόνο ο πρώτος και ο τρίτος θεωρούνται αμοιβαίοι γείτονες και μόνο αυτοί συμμετέχουν στην ψηφοφορία για την κλάση της νέας παρατήρησης.

Το παράδειγμα δείχνει ότι στον Mutual k-NN δεν είναι αρκετό ένα σημείο να βρίσκεται κοντά στη νέα παρατήρηση. Η σχέση πρέπει να είναι αμοιβαία. Για τον λόγο αυτό, ο αριθμός των παρατηρήσεων που τελικά συμμετέχουν στην ψηφοφορία μπορεί να είναι μικρότερος από την τιμή του k .

Υπάρχει επίσης η περίπτωση κανένας από τους αρχικούς γείτονες να μην έχει αμοιβαία σχέση με τη νέα παρατήρηση. Στην υλοποίηση που χρησιμοποιήθηκε στην παρούσα εργασία, σε αυτή την περίπτωση ενεργοποιείται ο μηχανισμός fallback και η νέα παρατήρηση ταξινομείται σύμφωνα με την κλάση του πλησιέστερου γείτονά της.

Αλγόριθμος 2 – Mutual k-NN

Require: Εκπαιδευτικό σύνολο $D_{\text{train}} = \{(x_i, y_i)\}_{i=1}^n$,
 σύνολο ελέγχου D_{test} , αριθμός γειτόνων k

Ensure: Προβλεπόμενη κατηγορία $\hat{y}$ για κάθε δείγμα του D_{test}

```

1: for  $i = 1$  to  $n$  do
2:   Υπολόγισε τις αποστάσεις  $d(x_i, x_j)$  από όλα τα  $x_j, j \neq i$ 
3:   Ταξινόμησε τις αποστάσεις κατά αύξουσα σειρά
4:   Βρες τον  $k$ -οστό πλησιέστερο γείτονα  $x_i^{(k)}$ 
5:    $r_i \leftarrow d(x_i, x_i^{(k)})$ 
6:   Αποθήκευσε την ακτίνα  $r_i$ 
7: end for
8: Φάση Ταξινόμησης:
9: for κάθε δείγμα  $x \in D_{\text{test}}$  do
10:  Βρες τους  $k$  πλησιέστερους γείτονες του  $x$ 
11:  Δημιούργησε κενό σύνολο mutual γειτόνων  $M(x)$ 
12:  for κάθε υποψήφιο γείτονα  $x_i$  do
13:    Υπολόγισε  $d(x, x_i)$ 
14:    if  $d(x, x_i) \leq r_i$  then
15:      Πρόσθεσε το  $x_i$  στο  $M(x)$ 
16:    end if
17:  end for
18:  if  $M(x) \neq \emptyset$  then
19:     $\hat{y}(x) \leftarrow \text{MajorityVote}(\{y_i : x_i \in M(x)\})$ 
20:  else
21:    Βρες τον πλησιέστερο γείτονα του  $x$ 
22:     $\hat{y}(x) \leftarrow$  κατηγορία του πλησιέστερου γείτονα
23:  end if
24:  Αποθήκευσε την προβλεπόμενη κατηγορία  $\hat{y}(x)$ 
25: end for
26: return  $\hat{Y}_{\text{pred}}$ 

```

2.3 Symmetric k-NN

Ο Symmetric k-Nearest Neighbors (Symmetric k-NN) [6,7] αποτελεί μία παραλλαγή του κλασικού k-NN, στην οποία διαφοροποιείται ο τρόπος με τον οποίο καθορίζεται η γειτονιά που χρησιμοποιείται για την ταξινόμηση μιας νέας παρατήρησης. Στον κλασικό k-NN λαμβάνονται υπόψη μόνο οι k πλησιέστεροι γείτονες της νέας παρατήρησης. Αντίθετα, στον Symmetric k-NN εξετάζεται η σχέση γειτονίας και προς την αντίθετη κατεύθυνση.

Πιο συγκεκριμένα, όταν μία νέα παρατήρηση πρέπει να ταξινομηθεί, αρχικά υπολογίζονται οι αποστάσεις της από τις παρατηρήσεις του συνόλου εκπαίδευσης και εντοπίζονται οι k πλησιέστεροι γείτονές της. Οι παρατηρήσεις αυτές συμμετέχουν στην ψηφοφορία για τον προσδιορισμό της κλάσης της νέας παρατήρησης, όπως ακριβώς συμβαίνει και στον κλασικό k-NN.

Στον Symmetric k-NN, όμως, η διαδικασία δεν σταματά στους k αυτούς γείτονες. Στη συνέχεια εξετάζονται και οι παρατηρήσεις του συνόλου εκπαίδευσης, προκειμένου να εντοπιστούν εκείνες που έχουν τη νέα παρατήρηση μέσα στους δικούς τους k πλησιέστερους γείτονες. Οι παρατηρήσεις αυτές προστίθενται επίσης στη γειτονιά που χρησιμοποιείται για την ταξινόμηση, ακόμη και αν δεν ανήκουν στους k πλησιέστερους γείτονες της νέας παρατήρησης. Επομένως, η συμμετρική γειτονιά σχηματίζεται από την **ένωση δύο συνόλων παρατηρήσεων**: των k πλησιέστερων γειτόνων της νέας παρατήρησης και των παρατηρήσεων που έχουν τη νέα παρατήρηση μέσα στους δικούς τους k πλησιέστερους γείτονες. Όλες οι παρατηρήσεις που ανήκουν στην τελική αυτή γειτονιά συμμετέχουν στην ψηφοφορία και η νέα παρατήρηση ταξινομείται στην κλάση που συγκεντρώνει τις περισσότερες ψήφους.

Με τον τρόπο αυτό, ο αριθμός των παρατηρήσεων που συμμετέχουν στην ψηφοφορία δεν περιορίζεται απαραίτητα στην τιμή του k. Η συμμετρική γειτονιά μπορεί να περιλαμβάνει περισσότερες από k παρατηρήσεις, καθώς είναι δυνατό να προστεθούν επιπλέον σημεία μέσω της αντίστροφης σχέσης γειτονίας.

Παράδειγμα λειτουργίας:

Για την καλύτερη κατανόηση της λειτουργίας του Symmetric k-NN, θεωρούμε ότι μία νέα παρατήρηση εισέρχεται στο σύστημα και πρέπει να ταξινομηθεί, χρησιμοποιώντας τιμή $k=3$. Αρχικά εντοπίζονται οι τρεις πλησιέστεροι γείτονες της νέας παρατήρησης. Οι τρεις αυτοί γείτονες συμμετέχουν στην ψηφοφορία για την κλάση της, όπως ακριβώς θα συνέβαινε και στον κλασικό k-NN. Στη συνέχεια, όμως, πραγματοποιείται ένας επιπλέον έλεγχος προς την αντίθετη κατεύθυνση. Εξετάζονται οι παρατηρήσεις του συνόλου εκπαίδευσης, ώστε να διαπιστωθεί αν υπάρχουν σημεία που έχουν τη νέα παρατήρηση μέσα στους δικούς τους τρεις πλησιέστερους γείτονες. Για παράδειγμα, μπορεί να υπάρξει μία τέταρτη παρατήρηση η οποία δεν βρίσκεται ανάμεσα στους τρεις πλησιέστερους γείτονες της νέας παρατήρησης. Παρ' όλα αυτά, η νέα παρατήρηση μπορεί να βρίσκεται ανάμεσα στους τρεις πλησιέστερους γείτονες αυτής της τέταρτης παρατήρησης. Στον Symmetric k-NN αυτό είναι αρκετό ώστε και η τέταρτη παρατήρηση να προστεθεί στη συμμετρική γειτονιά και να συμμετέχει στην ψηφοφορία.

Στο συγκεκριμένο παράδειγμα, λοιπόν, παρόλο που έχει επιλεγεί $k=3$, στην τελική ψηφοφορία συμμετέχουν τέσσερις παρατηρήσεις: οι τρεις πλησιέστεροι γείτονες της νέας παρατήρησης και η επιπλέον παρατήρηση που έχει τη νέα παρατήρηση μέσα στη δική της τριάδα πλησιέστερων γειτόνων.

Αν από τις τέσσερις αυτές παρατηρήσεις οι τρεις ανήκουν στην Κλάση 1 και η μία στην Κλάση 2, τότε η νέα παρατήρηση ταξινομείται στην Κλάση 1, καθώς αυτή συγκεντρώνει την πλειοψηφία των ψήφων. Το παράδειγμα αναδεικνύει τη βασική διαφορά του Symmetric k-NN από τον κλασικό k-NN. Η ταξινόμηση δεν βασίζεται αποκλειστικά στους γείτονες που επιλέγει η νέα παρατήρηση, αλλά λαμβάνει υπόψη και τις παρατηρήσεις που, αντίστροφα, θεωρούν τη νέα παρατήρηση ως έναν από τους δικούς τους πλησιέστερους γείτονες.

Παράλληλα, ο Symmetric k-NN διαφοροποιείται και από τον Mutual k-NN. Στον Mutual k-NN απαιτείται η σχέση γειτονίας να ισχύει ταυτόχρονα και προς τις δύο κατευθύνσεις. Δηλαδή, ένα σημείο πρέπει να βρίσκεται στους kk πλησιέστερους γείτονες της νέας παρατήρησης και ταυτόχρονα η νέα παρατήρηση να βρίσκεται στους kk πλησιέστερους γείτονες του συγκεκριμένου σημείου. Αντίθετα, στον Symmetric k-NN αρκεί να ισχύει μία από τις δύο σχέσεις. Ένα σημείο συμμετέχει στην ψηφοφορία είτε επειδή είναι ένας από τους kk πλησιέστερους γείτονες της νέας παρατήρησης είτε επειδή έχει τη νέα παρατήρηση μέσα στους δικούς του kk πλησιέστερους γείτονες.

Αλγόριθμος 3 – Symmetric k-NN

```

Require: Εκπαιδευτικό σύνολο  $D_{train} = \{(x_i, y_i)\}_{i=1}^n$ ,
        σύνολο ελέγχου  $D_{test}$ , αριθμός γειτόνων  $k$ 
Ensure: Προβλεπόμενη κατηγορία  $\hat{y}$  για κάθε δείγμα του  $D_{test}$ 
1: for  $i = 1$  to  $n$  do
2:   Υπολόγισε τις αποστάσεις  $d(x_i, x_j)$  από όλα τα  $x_j, j \neq i$ 
3:   Βρες τους  $k$  πλησιέστερους γείτονες του  $x_i$ 
4:    $r_i \leftarrow d(x_i, x_i^{(k)})$ 
5:   Αποθήκευσε την ακτίνα  $r_i$ 
6: end for
7: Φάση Ταξινόμησης:
8: for κάθε δείγμα  $x \in D_{test}$  do
9:   Βρες τους  $k$  πλησιέστερους training γείτονες και αποθήκευσέ τους στο  $N_k(x)$ 
10:  Δημιούργησε κενό σύνολο αντίστροφων γειτόνων  $R(x)$ 
11:  for κάθε training δείγμα  $x_i$  do
12:    Υπολόγισε  $d(x, x_i)$ 
13:    if  $d(x, x_i) \leq r_i$  then
14:      Πρόσθεσε το  $x_i$  στο  $R(x)$ 
15:    end if
16:  end for
17:   $N_{sym}(x) \leftarrow N_k(x) \cup R(x)$ 
18:  Αφαίρεσε τυχόν διπλότυπα από το  $N_{sym}(x)$ 
19:  Συγκέντρωσε τις κατηγορίες των δειγμάτων του  $N_{sym}(x)$ 
20:   $\hat{y}(x) \leftarrow \text{MajorityVote}(\{y_i : x_i \in N_{sym}(x)\})$ 
21:  Αποθήκευσε την προβλεπόμενη κατηγορία  $\hat{y}(x)$ 
22: end for
23: return  $\hat{Y}_{pred}$ 

```

Κεφάλαιο 3ο: Τεχνικές Μείωσης Δεδομένων

3.1 Είδη τεχνικών Μείωσης Δεδομένων

Οι τεχνικές μείωσης δεδομένων (Data Reduction Techniques) χρησιμοποιούνται με σκοπό τον περιορισμό του μεγέθους ενός συνόλου δεδομένων, διατηρώντας παράλληλα όσο το δυνατόν περισσότερο την πληροφορία που είναι απαραίτητη για τη διαδικασία της κατηγοριοποίησης. Η εφαρμογή τους μπορεί να οδηγήσει σε μικρότερα σύνολα εκπαίδευσης και, κατά συνέπεια, σε μείωση του υπολογιστικού κόστους που απαιτείται από έναν αλγόριθμο κατηγοριοποίησης.

Η μείωση του συνόλου εκπαίδευσης παρουσιάζει ιδιαίτερο ενδιαφέρον στους αλγορίθμους που βασίζονται στους πλησιέστερους γείτονες. Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, κατά την ταξινόμηση μιας νέας παρατήρησης απαιτείται η αναζήτηση γειτονικών παρατηρήσεων μέσα στο σύνολο εκπαίδευσης. Επομένως, όσο μεγαλύτερο είναι το σύνολο αυτό, τόσο περισσότερες παρατηρήσεις είναι διαθέσιμες και πρέπει να εξεταστούν κατά τη διαδικασία αναζήτησης των πλησιέστερων γειτόνων.

Η βασική ιδέα των τεχνικών μείωσης δεδομένων είναι να δημιουργηθεί ένα μικρότερο σύνολο εκπαίδευσης, το οποίο θα μπορεί να χρησιμοποιηθεί στη θέση του αρχικού. Ωστόσο, η μείωση του αριθμού των παρατηρήσεων δεν αποτελεί από μόνη της ένδειξη ότι μία τεχνική είναι αποτελεσματική. Είναι απαραίτητο το μειωμένο σύνολο να εξακολουθεί να περιέχει επαρκή πληροφορία, ώστε ο classifier να μπορεί να πραγματοποιεί σωστά την κατηγοριοποίηση νέων παρατηρήσεων.

Για παράδειγμα, μία τεχνική θα μπορούσε να αφαιρέσει ένα πολύ μεγάλο μέρος του αρχικού συνόλου και να μειώσει σημαντικά τον απαιτούμενο χρόνο εκτέλεσης. Αν όμως η αφαίρεση αυτή προκαλούσε σημαντική μείωση της ακρίβειας ταξινόμησης, τότε το όφελος της μείωσης θα συνοδευόταν από αντίστοιχο κόστος στην ποιότητα των αποτελεσμάτων. Για τον λόγο αυτό, η αξιολόγηση των τεχνικών μείωσης πρέπει να λαμβάνει υπόψη τόσο τον βαθμό μείωσης των δεδομένων όσο και την επίδραση που αυτή έχει στην απόδοση του classifier.

Οι τεχνικές μείωσης δεδομένων μπορούν να ακολουθούν διαφορετικές προσεγγίσεις. Ορισμένες επικεντρώνονται στην αφαίρεση παρατηρήσεων που θεωρούνται θορυβώδεις ή δεν συμφωνούν με τη γειτονιά τους, ενώ άλλες επιδιώκουν να επιλέξουν ένα μικρότερο σύνολο αντιπροσωπευτικών παρατηρήσεων από το αρχικό dataset.

Στην παρούσα εργασία εξετάζονται δύο τέτοιες τεχνικές: η Condensed Nearest Neighbor (CNN) και η Edited Nearest Neighbor (ENN). Οι δύο τεχνικές ακολουθούν διαφορετική λογική μείωσης. Η CNN επιδιώκει τη δημιουργία ενός μικρότερου συνόλου αντιπροσωπευτικών παρατηρήσεων, ενώ η ENN επικεντρώνεται στην αφαίρεση παρατηρήσεων που δεν συμφωνούν με την τοπική τους γειτονιά. Επιπλέον, εξετάζεται και η διαδοχική εφαρμογή των δύο τεχνικών με τη μορφή ENN→CNN.

3.2 Condensed Nearest Neighbor

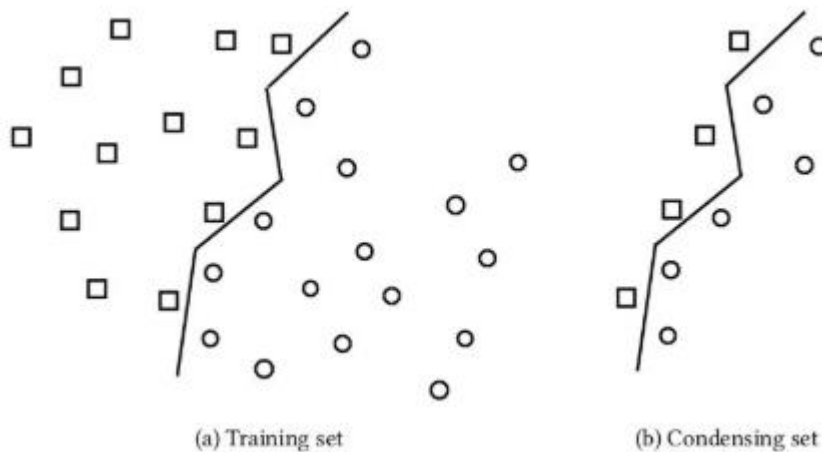
Η **Condensed Nearest Neighbor (CNN)** [4] είναι μία τεχνική μείωσης δεδομένων που έχει ως στόχο τη δημιουργία ενός μικρότερου συνόλου εκπαίδευσης, επιλέγοντας από το αρχικό dataset παρατηρήσεις που είναι απαραίτητες για την κατηγοριοποίηση.

Η βασική ιδέα της CNN είναι ότι δεν είναι απαραίτητο να διατηρούνται όλες οι παρατηρήσεις του αρχικού συνόλου εκπαίδευσης. Σε πολλές περιπτώσεις υπάρχουν παρατηρήσεις οι οποίες μπορούν ήδη να ταξινομηθούν σωστά χρησιμοποιώντας άλλες παρατηρήσεις του dataset. Η διατήρησή τους, επομένως, μπορεί να μην προσφέρει επιπλέον πληροφορία στη διαδικασία της ταξινόμησης.

Για τον λόγο αυτό, η CNN δημιουργεί σταδιακά ένα μικρότερο σύνολο αντιπροσωπευτικών παρατηρήσεων, οι οποίες αναφέρονται ως prototypes. Η διαδικασία ξεκινά με την επιλογή μιας αρχικής παρατήρησης, η οποία τοποθετείται στο σύνολο των prototypes.

Στη συνέχεια εξετάζονται διαδοχικά οι υπόλοιπες παρατηρήσεις του αρχικού συνόλου. Κάθε παρατήρηση ταξινομείται χρησιμοποιώντας το σύνολο των prototypes που έχει δημιουργηθεί μέχρι εκείνη τη στιγμή. Αν η παρατήρηση ταξινομηθεί σωστά, δεν χρειάζεται να προστεθεί στο σύνολο των prototypes. Αν όμως ταξινομηθεί λανθασμένα, τότε θεωρείται ότι το υπάρχον σύνολο δεν είναι επαρκές για τη σωστή αναπαράστασή της και η συγκεκριμένη παρατήρηση προστίθεται στα prototypes.

Με την προσθήκη της, το σύνολο των prototypes εμπλουτίζεται και χρησιμοποιείται για την ταξινόμηση των επόμενων παρατηρήσεων. Η διαδικασία επαναλαμβάνεται μέχρι να ολοκληρωθεί ο έλεγχος των παρατηρήσεων και να μην απαιτείται η προσθήκη νέων prototypes.



Παράδειγμα λειτουργίας:

Για παράδειγμα, θεωρούμε ένα σύνολο εκπαίδευσης που αποτελείται από παρατηρήσεις δύο διαφορετικών κλάσεων. Αρχικά επιλέγεται μία παρατήρηση και τοποθετείται στο σύνολο των prototypes. Στη συνέχεια εξετάζεται μία δεύτερη παρατήρηση. Αν ο πλησιέστερος prototype την ταξινομεί σωστά, η παρατήρηση αυτή δεν προστίθεται στο μειωμένο σύνολο. Αντίθετα, αν ταξινομείται λανθασμένα, προστίθεται ως νέο prototype. Η ίδια διαδικασία συνεχίζεται και για τις υπόλοιπες παρατηρήσεις. Με αυτόν τον τρόπο, παρατηρήσεις οι οποίες μπορούν ήδη να ταξινομηθούν σωστά από τα υπάρχοντα prototypes δεν χρειάζεται να διατηρηθούν, ενώ εκείνες που είναι απαραίτητες για τη σωστή αναπαράσταση διαφορετικών περιοχών του dataset προστίθενται στο μειωμένο σύνολο. Στο τέλος της διαδικασίας, αντί να χρησιμοποιείται ολόκληρο το αρχικό training set, χρησιμοποιείται το μικρότερο σύνολο των prototypes που προέκυψε από την CNN. Για να υπολογιστεί το ποσοστό μείωσης που επιτυγχάνει η CNN χρησιμοποιείται ο Reduction Rate, ο οποίος δίνεται από τον τύπο:

$$Reduction Rate = (1 - \frac{N_{cnn}}{N_{original}}) \times 100\%$$

όπου $N_{original}$ είναι ο αριθμός των παρατηρήσεων του αρχικού συνόλου εκπαίδευσης και N_{cnn} ο αριθμός των prototypes που παραμένουν μετά την εφαρμογή της CNN.

Για παράδειγμα, αν το αρχικό σύνολο αποτελείται από 1.000 παρατηρήσεις και μετά την εφαρμογή της CNN παραμένουν 200 prototypes, ο Reduction Rate είναι 80%. Αυτό σημαίνει ότι αφαιρέθηκε το 80% των αρχικών παρατηρήσεων και διατηρήθηκε το υπόλοιπο 20%.

Η CNN μπορεί επομένως να οδηγήσει σε σημαντική μείωση του αριθμού των παρατηρήσεων. Ωστόσο, η έντονη συμπίκνωση του dataset μπορεί σε ορισμένες περιπτώσεις να επηρεάσει την ποιότητα της ταξινόμησης, καθώς το μειωμένο σύνολο περιέχει λιγότερες παρατηρήσεις από το αρχικό. Για τον λόγο αυτό, στην πειραματική μελέτη εξετάζεται όχι μόνο το ποσοστό μείωσης που επιτυγχάνει η CNN, αλλά και η επίδρασή της στην απόδοση των classifiers. Στην παρούσα εργασία, η συγκεκριμένη μέτρηση είναι ιδιαίτερα σημαντική καθώς αποτελεί βασικό κριτήριο της αξιολόγησης της αποτελεσματικότητας των τεχνικών μείωσης δεδομένων.

3.3 Edited Nearest Neighbor

Η **Edited Nearest Neighbor (ENN)** [4] ακολουθεί διαφορετική προσέγγιση από την CNN. Αντί να προσπαθεί να δημιουργήσει ένα μικρό σύνολο αντιπροσωπευτικών prototypes, εξετάζει τις παρατηρήσεις του υπάρχοντος συνόλου εκπαίδευσης και αφαιρεί εκείνες που δεν συμφωνούν με την τοπική τους γειτονιά.

Για κάθε παρατήρηση του training set εντοπίζονται οι πλησιέστεροι γείτονές της. Η ίδια η παρατήρηση που εξετάζεται δεν λαμβάνεται υπόψη ως γείτονα του εαυτού της. Στη συνέχεια πραγματοποιείται πρόβλεψη της κλάσης της με βάση την πλειοψηφία των κλάσεων των γειτόνων της.

Η προβλεπόμενη κλάση συγκρίνεται με την πραγματική κλάση της παρατήρησης. Αν οι δύο κλάσεις συμφωνούν, η παρατήρηση παραμένει στο σύνολο εκπαίδευσης. Αντίθετα, αν η πλειοψηφία των γειτονικών παρατηρήσεων ανήκει σε διαφορετική κλάση, η συγκεκριμένη παρατήρηση χαρακτηρίζεται για αφαίρεση.

Με αυτή τη διαδικασία, η ENN λειτουργεί ως μία τεχνική **editing** του συνόλου δεδομένων. Σκοπός της δεν είναι απαραίτητα να επιτύχει τη μεγαλύτερη δυνατή μείωση, αλλά να αφαιρέσει παρατηρήσεις οι οποίες δεν είναι συνεπείς με τη γειτονική περιοχή στην οποία βρίσκονται.

Τέτοιες παρατηρήσεις μπορεί να σχετίζονται με θόρυβο στα δεδομένα ή να βρίσκονται σε περιοχές κοντά στα όρια μεταξύ διαφορετικών κλάσεων. Η αφαίρεσή τους έχει ως αποτέλεσμα τη δημιουργία ενός επεξεργασμένου συνόλου εκπαίδευσης, στο οποίο έχουν διατηρηθεί οι παρατηρήσεις που συμφωνούν με την τοπική τους γειτονιά.

Παράδειγμα λειτουργίας

Για παράδειγμα, θεωρούμε μία παρατήρηση που ανήκει στην Κλάση 1 και βρίσκεται σε μία περιοχή όπου οι περισσότεροι από τους πλησιέστερους γείτονές της ανήκουν στην Κλάση 2.

Η ENN εξετάζει τους πλησιέστερους γείτονες της συγκεκριμένης παρατήρησης και πραγματοποιεί μία πρόβλεψη για την κλάση της. Εφόσον η πλειοψηφία των γειτόνων ανήκει στην Κλάση 2, η πρόβλεψη είναι Κλάση 2. Επειδή όμως η πραγματική κλάση της παρατήρησης είναι Κλάση 1, υπάρχει διαφωνία

μεταξύ της πραγματικής κλάσης και της τοπικής γειτονιάς. Η παρατήρηση αυτή χαρακτηρίζεται επομένως για αφαίρεση.

Αντίθετα, μία παρατήρηση της Κλάσης 1 που περιβάλλεται κυρίως από άλλες παρατηρήσεις της ίδιας κλάσης ταξινομείται σωστά από τη γειτονιά της και παραμένει στο σύνολο.

Μετά την εξέταση όλων των παρατηρήσεων δημιουργείται το **edited set**, το οποίο αποτελείται από τις παρατηρήσεις που δεν χαρακτηρίστηκαν για αφαίρεση.

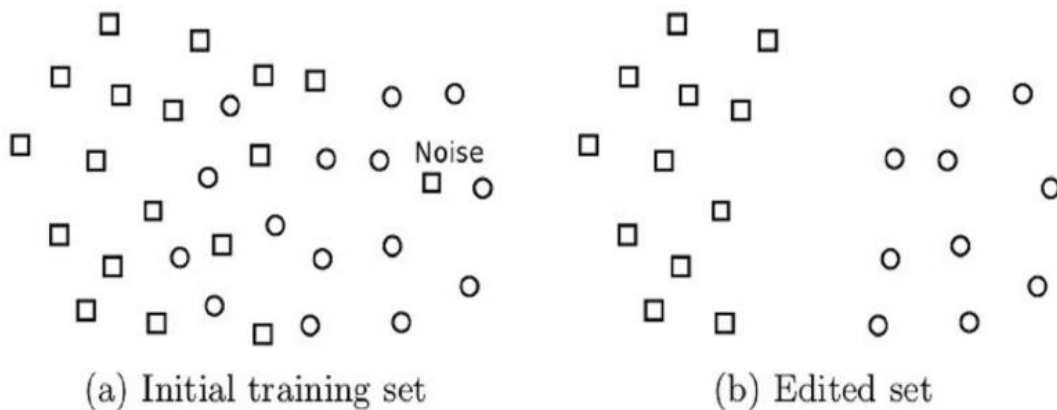
Το ποσοστό μείωσης που επιτυγχάνεται μέσω της ENN υπολογίζεται με τον Reduction Rate:

$$Reduction Rate = (1 - \frac{N_{enn}}{N_{original}}) \times 100\%$$

όπου $N_{original}$ είναι ο αριθμός των παρατηρήσεων του αρχικού συνόλου εκπαίδευσης και N_{ENN} ο αριθμός των παρατηρήσεων που παραμένουν μετά την εφαρμογή της ENN.

Για παράδειγμα, αν το αρχικό training set αποτελείται από 1.000 παρατηρήσεις και μετά την εφαρμογή της ENN παραμένουν 800, τότε ο Reduction Rate είναι 20%. Επομένως, η ENN αφαίρεσε το 20% των αρχικών παρατηρήσεων και διατήρησε το 80%.

Σε αντίθεση με την CNN, λοιπόν, η ENN δεν επιδιώκει πρωτίστως τη δημιουργία ενός πολύ μικρού συνόλου prototypes. Η μείωση προκύπτει ως αποτέλεσμα της αφαίρεσης των παρατηρήσεων που δεν συμφωνούν με τη γειτονιά τους.



3.4 Συνδυασμός ENN - CNN

Οι τεχνικές ENN και CNN μπορούν να χρησιμοποιηθούν ανεξάρτητα, αλλά στην παρούσα εργασία εξετάστηκε και η διαδοχική εφαρμογή τους. Συγκεκριμένα, εφαρμόστηκε αρχικά η τεχνική ENN και στη συνέχεια η CNN, δηλαδή ο συνδυασμός ENN→CNN.

Η επιλογή αυτής της σειράς βασίζεται στον διαφορετικό ρόλο που έχει κάθε τεχνική. Όπως αναφέρθηκε προηγουμένως, η ENN λειτουργεί ως τεχνική editing. Εξετάζει το αρχικό dataset και αφαιρεί

παρατηρήσεις που δεν συμφωνούν με την τοπική τους γειτονιά. Με αυτόν τον τρόπο παράγεται ένα επεξεργασμένο σύνολο δεδομένων.

Στο δεύτερο στάδιο, η CNN δεν εφαρμόζεται πλέον στο αρχικό dataset, αλλά στο σύνολο που έχει προκύψει μετά την εφαρμογή της ENN. Πάνω σε αυτό το σύνολο πραγματοποιείται η διαδικασία επιλογής των prototypes, με στόχο την περαιτέρω μείωση του αριθμού των παρατηρήσεων.

Η λογική του συνδυασμού μπορεί επομένως να περιγραφεί ως **editing και στη συνέχεια condensation**. Αρχικά η ENN επεξεργάζεται το dataset και στη συνέχεια η CNN επιχειρεί να δημιουργήσει ένα μικρότερο σύνολο αντιπροσωπευτικών παρατηρήσεων από το αποτέλεσμα του πρώτου σταδίου.

Παράδειγμα λειτουργίας

Για παράδειγμα, θεωρούμε ένα αρχικό σύνολο εκπαίδευσης το οποίο περιλαμβάνει μεγάλο αριθμό παρατηρήσεων, μεταξύ των οποίων υπάρχουν και ορισμένα σημεία που δεν συμφωνούν με την τοπική τους γειτονιά. Στο πρώτο στάδιο εφαρμόζεται η ENN. Οι παρατηρήσεις εξετάζονται με βάση τους πλησιέστερους γείτονές τους και όσες δεν συμφωνούν με την πλειοψηφία της γειτονιάς τους αφαιρούνται. Το αποτέλεσμα είναι ένα μικρότερο και επεξεργασμένο σύνολο δεδομένων.

Στη συνέχεια εφαρμόζεται η CNN πάνω στο σύνολο που προέκυψε από την ENN. Η CNN εξετάζει ποιες από τις παρατηρήσεις που έχουν απομείνει είναι απαραίτητες ως prototypes και διατηρεί ένα ακόμη μικρότερο υποσύνολο. Με τον τρόπο αυτό, οι δύο τεχνικές λειτουργούν διαδοχικά και με διαφορετικό σκοπό: η ENN πραγματοποιεί πρώτα το editing του dataset και η CNN στη συνέχεια πραγματοποιεί τη συμπίκνωσή του.

Για την περίπτωση ENN→CNN, ο Reduction Rate του δεύτερου σταδίου υπολογίζεται σε σχέση με το σύνολο που έχει ήδη προκύψει μετά την εφαρμογή της ENN και δίνεται από τον τύπο:

$$Reduction Rate = (1 - \frac{N_{enn \rightarrow cnn}}{N_{enn}}) \times 100\%$$

όπου N_{ENN} είναι ο αριθμός των παρατηρήσεων που παραμένουν μετά την εφαρμογή της ENN και $N_{ENN \rightarrow CNN}$ ο τελικός αριθμός των prototypes που παραμένουν μετά την εφαρμογή της CNN στο edited set. Επομένως, ο συγκεκριμένος Reduction Rate εκφράζει το ποσοστό μείωσης που επιτυγχάνει η CNN **πάνω στο ήδη επεξεργασμένο σύνολο της ENN** και όχι τη συνολική μείωση σε σχέση με το αρχικό dataset.

Για παράδειγμα, αν μετά την εφαρμογή της ENN έχουν παραμείνει 800 παρατηρήσεις και μετά την εφαρμογή της CNN σε αυτές παραμένουν 80 prototypes, ο Reduction Rate του δεύτερου σταδίου είναι 90%. Αυτό σημαίνει ότι η CNN αφαίρεσε το 90% των παρατηρήσεων που είχε παραλάβει από την ENN.

Αντίθετα, όταν ζητείται η **συνολική μείωση** που έχει επιτευχθεί μετά και τα δύο στάδια σε σχέση με το αρχικό σύνολο δεδομένων, μπορεί να χρησιμοποιηθεί ο συνολικός Reduction Rate:

$$Reduction Rate = (1 - \frac{N_{enn \rightarrow cnn}}{N_{original}}) \times 100\%$$

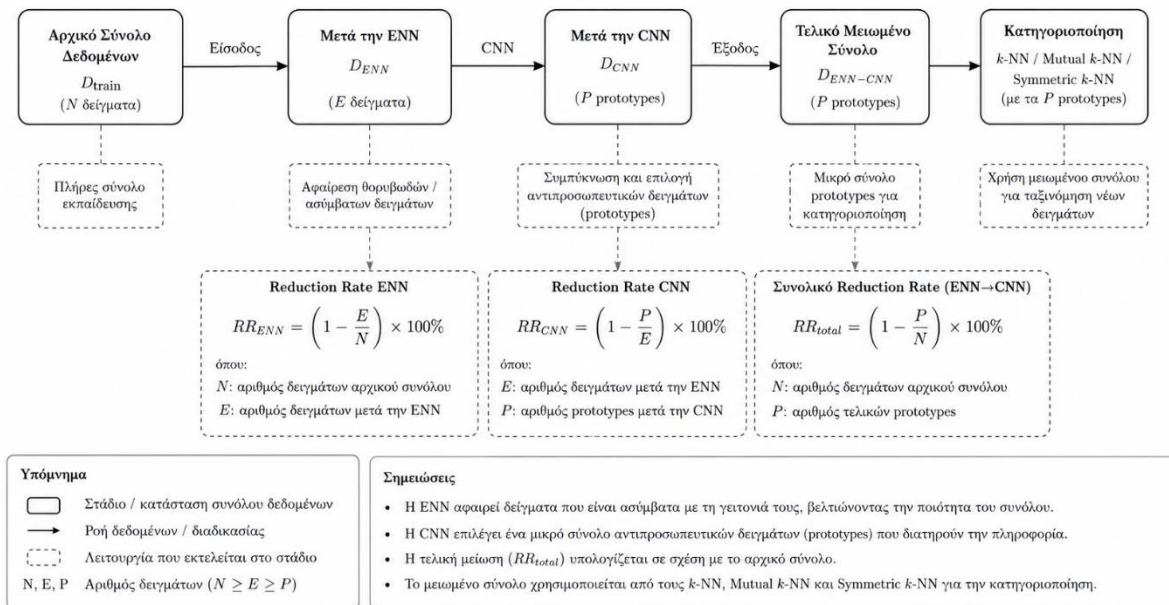
όπου $N_{original}$ είναι ο αριθμός των παρατηρήσεων του αρχικού training set και $N_{ENN \rightarrow CNN}$ ο αριθμός των παρατηρήσεων που παραμένουν μετά την ολοκλήρωση και των δύο σταδίων.

Η διάκριση μεταξύ των δύο παραπάνω ποσοστών είναι σημαντική. Ο πρώτος δείκτης περιγράφει τη μείωση που πραγματοποιεί η CNN σε σχέση με το αποτέλεσμα της ENN, ενώ ο δεύτερος περιγράφει τη συνολική μείωση από το αρχικό dataset μέχρι το τελικό σύνολο που προκύπτει από τον συνδυασμό ENN→CNN.

Ο συνδυασμός των δύο τεχνικών μπορεί να οδηγήσει σε σημαντική μείωση του αριθμού των παρατηρήσεων που χρησιμοποιούνται τελικά για την ταξινόμηση. Ωστόσο, μεγαλύτερη μείωση δεν συνεπάγεται απαραίτητα και καλύτερη απόδοση του classifier. Για τον λόγο αυτό, στην πειραματική μελέτη αξιολογείται τόσο ο βαθμός μείωσης όσο και η επίδρασή του στην ποιότητα της ταξινόμησης και στο απαιτούμενο υπολογιστικό κόστος.

Συνοψίζοντας, οι δύο τεχνικές που εξετάζονται ακολουθούν διαφορετική προσέγγιση στη μείωση των δεδομένων. Η ENN επικεντρώνεται στην επεξεργασία του συνόλου μέσω της αφαίρεσης παρατηρήσεων που δεν συμφωνούν με τη γειτονιά τους, ενώ η CNN επιδιώκει τη δημιουργία ενός μικρότερου συνόλου αντιπροσωπευτικών prototypes. Ο συνδυασμός ENN→CNN αξιοποιεί διαδοχικά τις δύο αυτές λειτουργίες, εφαρμόζοντας πρώτα editing και στη συνέχεια condensation. Στα επόμενα κεφάλαια παρουσιάζεται η υλοποίηση των τεχνικών και αξιολογείται πειραματικά η επίδρασή τους στους k-NN, Mutual k-NN και Symmetric k-NN.

Σχήμα 3.1: Διάγραμμα ροής της διαδικασίας μείωσης δεδομένων με ENN και CNN.



Κεφάλαιο 4ο: Υλοποίηση Αλγορίθμων

4.1 Υλοποίηση k-NN κατηγοριοποιητών

Στην παρούσα εργασία χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python, η οποία αποτελεί μία σύγχρονη γλώσσα με απλή σύνταξη, ευκολία χρήσης και εκτεταμένα εργαλεία για τη διαχείριση και ανάλυση δεδομένων. Υλοποιήθηκαν οι τρεις κατηγοριοποιητές, ο k-NN, ο Mutual k-NN, ο Symmetric k-NN όπως έχει ήδη αναφερθεί. Η υλοποίηση σχεδιάστηκε έτσι ώστε οι αλγόριθμοι να εκτελούνται αυτοματοποιημένα για όλα τα σύνολα δεδομένων και για τις τιμές $k = \{1, 3, 5, 7\}$.

Για την επεξεργασία δεδομένων έγινε χρήση βιβλιοθηκών της Python με τις κυριότερες να βρίσκονται στον παρακάτω πίνακα:

Πίνακας 4.1: Κυριότερες βιβλιοθήκες Python και η χρήση τους στην υλοποίηση

Βιβλιοθήκη	Χρήση
Pandas	Επεξεργασία δεδομένων
NumPy	Αριθμητικοί υπολογισμοί
Scikit-learn	k-NN, αναζήτηση γειτόνων και μετρικές
SciPy	Majority voting στον Symmetric k-NN
time	Μέτρηση χρόνου CPU

Τα datasets εισάγονται από αρχεία CSV. Η τελευταία στήλη χρησιμοποιείται ως ετικέτα κατηγορίας, ενώ οι υπόλοιπες στήλες αποτελούν τα χαρακτηριστικά των δειγμάτων.

Ο Directed k-NN αποτελεί την κλασική μορφή του κατηγοριοποιητή εγγύτερων γειτόνων και για την υλοποίηση του χρησιμοποιήθηκε η κλάση KNeighborsClassifier της βιβλιοθήκης Scikit-learn [18].

```
knn = KNeighborsClassifier(
    n_neighbors=k
)

knn.fit(X_train, y_train)

y_pred = knn.predict(X_test)
```

Στον Mutual k-NN το κριτήριο γειτνίασης εμπλουτίζεται με το εάν η σχέση μεταξύ των δύο δειγμάτων είναι αμοιβαία. Εντοπίζονται οι πλησιέστεροι γείτονες και εν συνεχεία ελέγχεται η αντίστοιχη συνθήκη mutual.

```
if d_test_to_tr <= max_dists_train[neighbor_idx]:
    mutuals.append(neighbor_idx)
```

Στη συνέχεια οι ετικέτες των mutual γειτόνων χρησιμοποιούνται για την τελική πρόβλεψη μέσω πλειοψηφικής ψηφοφορίας. Σε περίπτωση μη έβρεσης αμοιβαίου γείτονα, η υλοποίηση διαθέτει μηχανισμό fallback ώστε να παραχθεί πρόβλεψη για το συγκεκριμένο δείγμα.

Στην περίπτωση του Symmetric η γειτονιά κατασκευάζεται με διαφορετικό τρόπο. Αρχικά εντοπίζονται οι k πλησιέστεροι γείτονες κάθε δείγματος ελέγχου.

```
nn_tr = NearestNeighbors(n_neighbors=k).fit(X_tr)
_, indices_ts = nn_tr.kneighbors(X_ts)
```

Στη συνέχεια υπολογίζεται η απόσταση του k-οστού γείτονα για κάθε δείγμα εκπαίδευσης. Με βάση αυτή την απόσταση προσδιορίζεται το αντίστοιχο symmetric σύνολο γειτόνων. Τα σύνολα γειτόνων συνδυάζονται και οι ετικέτες τους χρησιμοποιούνται για την τελική πλειοψηφική ψηφοφορία:

```
all_voters = np.unique(
    np.concatenate([k_neighbors, j_neighbors])
)
```

Επιπροσθέτως για κάθε αλγόριθμο υπολογίζονται οι βασικές μετρικές αξιολόγησης:

- Accuracy
- Precision
- Recall
- F1-score
- CPU Time
- Fallback (Mutual k-NN)

Τα αποτελέσματα των επιμέρους εκτελέσεων αποθηκεύονται σε αρχείο Excel και υπολογίζεται ο μέσος όρος αυτών. Με αυτό τον τρόπο προκύπτει ένα ενιαίο σύνολο αποτελεσμάτων για κάθε συνδυασμό dataset – αλγόριθμου – τιμής k.

4.2 Υλοποιήσεις ENN - CNN

Για τη χρήση τεχνικών μείωσης δεδομένων, στην παρούσα εργασία, χρησιμοποιήθηκαν οι έτοιμες υλοποιήσεις των αλγορίθμων Editing Nearest Neighbors (ENN) και Condensed Nearest Neighbors (CNN) σε γλώσσα προγραμματισμού C++.

CNN

```
//Harts Condensing Nearest Neighbor Rule
do{
    fl=0;
    g++;
    for (i=0; i<dataNumV; i++){
        if (trainData[i].SX == 'X'){ //for each item belong to X
            nn=NearestNeighbor(trainData[i], trainData, dataNumV);
            if (nn!=-1){
                if (trainData[nn].classAttr != trainData[i].classAttr){
                    trainData[i].SX='S';
                    fl=1;
                }
            }
        }
    }
    c=0;
    for (i=0; i<dataNumV; i++){
        if (trainData[i].SX == 'X'){
            c++;
        }
    }
}
while ((fl != 0) && (c != 0));
```

Στο παραπάνω τμήμα εφαρμόζεται η βασική διαδικασία του CNN. Αρχικά, η πρώτη παρατήρηση τοποθετείται στο σύνολο των prototypes (S). Στη συνέχεια εξετάζονται οι υπόλοιπες παρατηρήσεις και εντοπίζεται ο πλησιέστερος γείτονάς τους από το σύνολο των prototypes. Εάν η κλάση του πλησιέστερου prototype είναι διαφορετική από την πραγματική κλάση της παρατήρησης, τότε η παρατήρηση προστίθεται στο σύνολο S. Η διαδικασία επαναλαμβάνεται μέχρι να μην απαιτείται η προσθήκη νέων prototypes.

ENN

```
//wilson editing
for (i=0; i<dataNumV; i++){
    prediction=conventionalKNNSearch(trainData, trainData[i], dataNumV, i);
    if (trainData[i].classAttr != prediction){
        trainData[i].remove=1;
    }
}
```

Στο συγκεκριμένο τμήμα εξετάζεται κάθε παρατήρηση του συνόλου δεδομένων και υπολογίζεται η προβλεπόμενη κλάση της μέσω των k πλησιέστερων γειτόνων. Εάν η προβλεπόμενη κλάση διαφέρει από την πραγματική κλάση της παρατήρησης, αυτή σημειώνεται προς αφαίρεση (remove = 1). Στη συνέχεια, κατά τη δημιουργία του νέου συνόλου δεδομένων, διατηρούνται μόνο οι παρατηρήσεις που δεν έχουν σημειωθεί για αφαίρεση.

Κεφάλαιο 5ο: Πειραματική Μελέτη

5.1 Σύνολα δεδομένων

Στην πειραματική μελέτη χρησιμοποιήθηκαν δεδομένα από το UCI Machine Learning Repository [17] και το KEEL Repository [16]. Είκοσι σύνολα δεδομένων ταξινόμησης τα οποία παρουσιάζουν διαφορετικό πλήθος χαρακτηριστικών και κλάσεων. Με τον τρόπο αυτό επιδιώκεται η αξιολόγηση των εξεταζόμενων μεθόδων σε προβλήματα διαφορετικής κλίμακας και πολυπλοκότητας. Για κάθε σύνολο δεδομένων χρησιμοποιήθηκαν πέντε ζεύγη αρχείων εκπαίδευσης και ελέγχου (data-tr1–data-tr5 και data-ts1–data-ts5). Τα δεδομένα περιλαμβάνουν αριθμητικά χαρακτηριστικά, ενώ η τελευταία στήλη κάθε αρχείου αντιστοιχεί στην κλάση της εκάστοτε παρατήρησης.

Balance (bl)

Το Balance Scale (bl), είναι ένα μικρό σύνολο δεδομένων ταξινόμησης με 625 παρατηρήσεις, 4 χαρακτηριστικά και 3 κλάσεις. Τα χαρακτηριστικά περιγράφουν παραμέτρους που σχετίζονται με τα δύο σκέλη μιας ζυγαριάς και το πρόβλημα αφορά την πρόβλεψη της κατάστασης ισορροπίας. Οι τρεις κλάσεις αντιστοιχούν σε ισορροπία προς τα αριστερά, ισορροπία προς τα δεξιά και πραγματική ισορροπία. Πρόκειται για πρόβλημα πολλαπλών κλάσεων, στο οποίο υπάρχει έντονη διαφορά ως προς το πλήθος των παρατηρήσεων των τριών κλάσεων.

Banana (bn)

Το Banana (bn) αποτελεί τεχνητό δυαδικό σύνολο δεδομένων με 5.300 παρατηρήσεις, 2 χαρακτηριστικά και 2 κλάσεις. Τα δύο χαρακτηριστικά αντιστοιχούν στις δύο διαστάσεις του χώρου όπου τοποθετούνται οι παρατηρήσεις. Τα δεδομένα σχηματίζουν δύο περιοχές με χαρακτηριστική μορφή μπανάνας και ο στόχος της ταξινόμησης είναι ο διαχωρισμός των δύο κλάσεων. Το συγκεκριμένο σύνολο είναι ιδιαίτερα χρήσιμο για την αξιολόγηση μεθόδων που βασίζονται σε γειτονικές σχέσεις, καθώς η γεωμετρική κατανομή των δεδομένων παίζει σημαντικό ρόλο στην ταξινόμηση.

Ecoli (ec1)

Το Ecoli (ec1) περιλαμβάνει 336 παρατηρήσεις, 7 χαρακτηριστικά και 8 κλάσεις. Το σύνολο αφορά την ταξινόμηση πρωτεϊνών με βάση χαρακτηριστικά που σχετίζονται με τη θέση τους μέσα στο κύτταρο. Οι κλάσεις αντιστοιχούν σε διαφορετικές υποκυτταρικές θέσεις των πρωτεϊνών. Το πρόβλημα αποτελεί χαρακτηριστικό παράδειγμα πολυταξικής ταξινόμησης, καθώς χρησιμοποιούνται οκτώ διαφορετικές κατηγορίες.

Eye State (eeg)

Το EEG Eye State περιλαμβάνει 14 χαρακτηριστικά και 2 κλάσεις. Το πλήρες σύνολο αποτελείται από 14.980 παρατηρήσεις, οι οποίες προέρχονται από καταγραφή ηλεκτροεγκεφαλογράφηματος μέσω συσκευής με 14 αισθητήρες. Η μεταβλητή στόχος περιγράφει την κατάσταση των ματιών του εξεταζόμενου, δηλαδή αν τα μάτια είναι ανοιχτά ή κλειστά.

KDD Cup

Το KDD Cup, αποτελεί ένα από τα μεγαλύτερα σύνολα δεδομένων που χρησιμοποιήθηκαν. Περιλαμβάνει 23 κλάσεις και χρησιμοποιείται για την ταξινόμηση δικτυακών συνδέσεων. Οι κλάσεις περιγράφουν διαφορετικούς τύπους δικτυακής συμπεριφοράς, συμπεριλαμβανομένης της φυσιολογικής κίνησης και διαφορετικών τύπων επιθέσεων. Η αρχική έκδοση του συνόλου περιλαμβάνει εκατοντάδες χιλιάδες παρατηρήσεις και 41 χαρακτηριστικά, ενώ στα αρχεία της παρούσας πειραματικής διαδικασίας χρησιμοποιούνται 36 αριθμητικά χαρακτηριστικά.

Letter Recognition (lir)

Το Letter Recognition (lir), αποτελεί σύνολο δεδομένων για την αναγνώριση γραμμάτων. Περιλαμβάνει 20.000 παρατηρήσεις, 16 χαρακτηριστικά και 26 κλάσεις, οι οποίες αντιστοιχούν στα γράμματα του αγγλικού αλφαβήτου από το Α έως το Ζ. Κάθε παρατήρηση περιγράφει χαρακτηριστικά ενός γράμματος και ο στόχος είναι η αναγνώριση της αντίστοιχης κατηγορίας.

Landsat Satellite (ls)

Το Landsat Satellite (ls), περιλαμβάνει 6.435 παρατηρήσεις, 36 χαρακτηριστικά και 6 κλάσεις. Κάθε παρατήρηση αντιστοιχεί σε περιοχή ή pixel δορυφορικής εικόνας και τα χαρακτηριστικά περιγράφουν φασματικές πληροφορίες. Οι κλάσεις αντιστοιχούν σε διαφορετικούς τύπους επιφάνειας ή κάλυψης εδάφους. Το σύνολο αποτελεί χαρακτηριστικό παράδειγμα πολυταξικής ταξινόμησης δεδομένων υψηλότερης διάστασης.

Magic Gamma Telescope (mgt)

Το Magic Gamma Telescope (mgt), αποτελεί δυαδικό σύνολο δεδομένων με 19.020 παρατηρήσεις, 10 χαρακτηριστικά και 2 κλάσεις. Οι παρατηρήσεις προέρχονται από προσομοιώσεις σωματιδίων που καταγράφονται από τηλεσκόπιο και ο στόχος είναι η διάκριση μεταξύ gamma και hadron γεγονότων. Το σύνολο παρουσιάζει σημαντικά μεγαλύτερο πλήθος παρατηρήσεων σε σχέση με τα μικρότερα datasets της μελέτης και επομένως είναι χρήσιμο για την αξιολόγηση του υπολογιστικού κόστους των αλγορίθμων.

MONK's Problems (mnk)

Το MONK's Problems(mnk) , αποτελεί τεχνητό πρόβλημα δυαδικής ταξινόμησης. Τα δεδομένα περιγράφουν αντικείμενα μέσω 6 χαρακτηριστικών διακριτών τιμών και η έξοδος αποτελείται από 2 κλάσεις. Το αρχικό σύνολο του χώρου των MONK's προβλημάτων περιλαμβάνει 432 δυνατούς συνδυασμούς χαρακτηριστικών. Τα προβλήματα δημιουργήθηκαν για την αξιολόγηση αλγορίθμων μηχανικής μάθησης ως προς την ικανότητά τους να μαθαίνουν λογικές έννοιες από παραδείγματα.

Pen-Based Recognition of Handwritten Digits (pd)

Το Pen-Based Recognition of Handwritten Digits (pd), με συμβολισμό pd, αφορά την αναγνώριση χειρόγραφων ψηφίων. Το σύνολο αποτελείται από 10.992 παρατηρήσεις, 16 χαρακτηριστικά και 10 κλάσεις, οι οποίες αντιστοιχούν στα ψηφία 0 έως 9. Κάθε παρατήρηση περιγράφει τη μορφή ενός χειρόγραφου ψηφίου μέσω αριθμητικών χαρακτηριστικών. Το συγκεκριμένο σύνολο αποτελεί χαρακτηριστικό παράδειγμα πολυταξικής ταξινόμησης.

Phoneme (ph)

Το Phoneme (ph), αποτελεί δυαδικό σύνολο δεδομένων με 5.404 παρατηρήσεις, 5 χαρακτηριστικά και 2 κλάσεις. Στόχος του είναι η διάκριση μεταξύ δύο κατηγοριών φωνημάτων με βάση χαρακτηριστικά που προκύπτουν από την επεξεργασία φωνητικού σήματος. Λόγω του σχετικά μεγάλου αριθμού παρατηρήσεων σε σχέση με τον μικρό αριθμό χαρακτηριστικών, το σύνολο είναι κατάλληλο για τη μελέτη της υπολογιστικής συμπεριφοράς των αλγορίθμων.

Pima Indians Diabetes (pm)

Το Pima Indians Diabetes (pm), περιλαμβάνει 768 παρατηρήσεις, 8 χαρακτηριστικά και 2 κλάσεις. Τα χαρακτηριστικά αφορούν ιατρικές και φυσιολογικές μετρήσεις και ο στόχος είναι η πρόβλεψη της παρουσίας ή απουσίας διαβήτη. Το σύνολο αποτελεί κλασικό παράδειγμα δυαδικής ταξινόμησης και χρησιμοποιείται ευρέως για την αξιολόγηση αλγορίθμων μηχανικής μάθησης.

Ringnorm (ring)

Το Ringnorm (ring), είναι τεχνητό δυαδικό σύνολο δεδομένων με 7.400 παρατηρήσεις, 20 χαρακτηριστικά και 2 κλάσεις. Οι παρατηρήσεις δημιουργούνται με τρόπο ώστε οι δύο κλάσεις να παρουσιάζουν διαφορετική γεωμετρική κατανομή στον χώρο των χαρακτηριστικών. Η σχετικά μεγάλη διάσταση του συνόλου το καθιστά χρήσιμο για την αξιολόγηση της συμπεριφοράς των αλγορίθμων όταν αυξάνεται το πλήθος των χαρακτηριστικών.

Segment (sg)

Το Image Segmentation (sg), περιλαμβάνει 2.310 παρατηρήσεις, 19 χαρακτηριστικά και 7 κλάσεις. Κάθε παρατήρηση αντιστοιχεί σε τμήμα εικόνας και τα χαρακτηριστικά περιγράφουν ιδιότητες όπως χρωματικά και χωρικά χαρακτηριστικά της περιοχής. Στόχος είναι η ταξινόμηση κάθε τμήματος σε μία από τις επτά κατηγορίες.

Statlog Shuttle (sh)

Το Statlog Shuttle (sh), αποτελεί μεγάλο σύνολο δεδομένων ταξινόμησης με 58.000 παρατηρήσεις, 9 χαρακτηριστικά και 7 κλάσεις. Τα δεδομένα χρησιμοποιούνται για την αναγνώριση της κατάλληλης λειτουργίας ελέγχου ενός διαστημικού λεωφορείου. Οι επτά κλάσεις αντιστοιχούν σε διαφορετικές καταστάσεις ελέγχου, όπως *Rad Flow*, *Fpn Close*, *Fpn Open*, *High*, *Bypass*, *Bpn Close* και *Bpn Open*.

Twonorm (tn)

Το Twonorm (tn), είναι τεχνητό δυαδικό σύνολο δεδομένων με 7.400 παρατηρήσεις, 20 χαρακτηριστικά και 2 κλάσεις. Τα δεδομένα έχουν σχεδιαστεί για την αξιολόγηση αλγορίθμων ταξινόμησης σε πρόβλημα δύο κλάσεων και παρουσιάζουν διαφορετικές κατανομές στον χώρο των χαρακτηριστικών. Μαζί με το Ringnorm αποτελεί ένα από τα μεγαλύτερα τεχνητά σύνολα δεδομένων που χρησιμοποιούνται στην πειραματική μελέτη.

Texture (txr)

Το Texture (txr), αποτελεί σύνολο δεδομένων ταξινόμησης υφών με 5.500 παρατηρήσεις, 40 χαρακτηριστικά και 11 κλάσεις. Τα χαρακτηριστικά αποτελούν αριθμητικούς περιγραφείς της υφής και κάθε παρατήρηση αντιστοιχεί σε περιοχή εικόνας. Οι 11 κλάσεις αντιστοιχούν σε διαφορετικούς τύπους υφής. Κάθε κλάση περιλαμβάνει 500 παρατηρήσεις στο πλήρες σύνολο δεδομένων.

Waveform (wf)

Το Waveform (wf), αποτελεί τεχνητό σύνολο δεδομένων με 5.000 παρατηρήσεις, 21 χαρακτηριστικά και 3 κλάσεις. Οι παρατηρήσεις προκύπτουν από συνδυασμούς διαφορετικών κυματομορφών και ο στόχος είναι η αναγνώριση της κατηγορίας στην οποία ανήκει κάθε παρατήρηση. Το σύνολο επιτρέπει την αξιολόγηση αλγορίθμων σε πρόβλημα πολλαπλών κλάσεων με σχετικά μεγάλο αριθμό χαρακτηριστικών.

Wine Quality (wine)

Το Wine Quality (wine), χρησιμοποιείται για την πρόβλεψη της ποιότητας λευκού κρασιού. Το πλήρες σύνολο περιλαμβάνει 4.898 παρατηρήσεις, 11 χαρακτηριστικά και 7 κλάσεις, οι οποίες αντιστοιχούν σε διαφορετικές βαθμολογίες ποιότητας. Τα χαρακτηριστικά προέρχονται από φυσικοχημικές μετρήσεις, όπως η οξύτητα, η περιεκτικότητα σε ζάχαρη, η πυκνότητα, το pH, τα θειώδη και η περιεκτικότητα σε αλκοόλη.

Yeast (ys)

Το Yeast (ys), περιλαμβάνει 1.484 παρατηρήσεις, 8 χαρακτηριστικά και 10 κλάσεις. Το σύνολο αφορά την πρόβλεψη της υποκυτταρικής θέσης πρωτεϊνών ζύμης. Τα χαρακτηριστικά περιγράφουν ιδιότητες των πρωτεϊνών και η τελευταία μεταβλητή χρησιμοποιείται για την ταξινόμησή τους στις διαφορετικές υποκυτταρικές κατηγορίες. Το σύνολο αποτελεί επομένως ένα πολυταξικό πρόβλημα με δέκα διαφορετικές κλάσεις.

Στον παρακάτω πίνακα παρουσιάζεται συγκεντρωτικά ο αριθμός παρατηρήσεων, χαρακτηριστικών και κλάσεων για κάθε dataset.

Πίνακας 5.1: Χαρακτηριστικά των συνόλων δεδομένων

Dataset	Παρατηρήσεις	Χαρακτηριστικά	Κλάσεις
Balance (bl)	625	4	3
Banana (bl)	5.300	2	2
Ecoli (ec1)	336	7	8
EEG Eye State (eeg)	14.980	14	2
KDD Cup (kdd)	494.020	36	23
Letter Recognition (lir)	20.000	16	26
Landsat Satellite (ls)	6.435	36	6
Magic Gamma Telescope (mgt)	19.020	10	2
MONK's Problems (mnk)	432	6	2
Pen-Based Digits	10.992	16	10
Phoneme (ph)	5.404	5	2
Pima Diabetes (pm)	768	8	2
Ringnorm (ring)	7.400	20	2
Segment (sg)	2.310	19	7
Shuttle (sh)	58.000	9	7
Twonorm (tn)	7.400	20	2
Texture (txr)	5.500	40	11
Waveform (wf)	5.000	21	3
Wine Quality (wine)	4.898	11	7
Yeast (ys)	1.484	8	10

5.2 Εγκαθίδρυση πειραμάτων

Η υλοποίηση και εκτέλεση του πειραματικού μέρους της εργασίας πραγματοποιήθηκε σε υπολογιστικό σύστημα με τις ακόλουθες προδιαγραφές:

- **Επεξεργαστής:** Intel® Core™ i7-6800K στα 3.40GHz (6 πυρήνες, 12 νήματα)
- **Μνήμη:** RAM: 16.0 GB
- **Λειτουργικό Σύστημα:** Windows 10 Pro, Version 22H2
- **Γλώσσα Προγραμματισμού:** Python 3.14
- **Περιβάλλον Εκτέλεσης:** Visual Studio Code
-

Ο επεξεργαστής Intel® Core™ i7-6800K, ο οποίος βασίζεται στην αρχιτεκτονική Broadwell, παρέχει υψηλή υπολογιστική απόδοση για εφαρμογές μηχανικής μάθησης, ενώ τα 16GB RAM εξασφαλίζουν επαρκή μνήμη για την επεξεργασία των datasets που χρησιμοποιήθηκαν.

Για την αξιολόγηση της απόδοσης των αλγορίθμων k-NN και των παραλλαγών τους, χρησιμοποιήθηκαν πέντε βασικές μετρικές που καλύπτουν τόσο την ταχύτητα όσο και την ποιότητα αξιολόγησης [12]. Οι μετρικές αυτές είναι ο χρόνος εκτέλεσης (CPU Time), η ακρίβεια (Accuracy), η μέση ακρίβεια ανά κλάση (Average Precision), η μέση ανάκληση ανά κλάση (Average Recall) και το F1-Score. Οι μετρικές παρουσιάζονται αναλυτικά παρακάτω.

Χρόνος Εκτέλεσης (CPU Time)

Ο χρόνος εκτέλεσης εκφράζει τον χρόνο που απαιτείται από τον αλγόριθμο για την ολοκλήρωση της διαδικασίας ταξινόμησης. Η συγκεκριμένη μετρική χρησιμοποιείται για την αξιολόγηση του υπολογιστικού κόστους κάθε αλγορίθμου. Μικρότερη τιμή CPU Time υποδηλώνει μικρότερο χρόνο εκτέλεσης και, επομένως, καλύτερη υπολογιστική αποδοτικότητα.

Ακρίβεια (Accuracy)

Η ακρίβεια εκφράζει το ποσοστό των παρατηρήσεων που ταξινομήθηκαν σωστά σε σχέση με το συνολικό πλήθος των παρατηρήσεων. Υπολογίζεται από τη σχέση:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

όπου TP και TN αντιστοιχούν στις σωστές θετικές και αρνητικές προβλέψεις, ενώ FP και FN στις λανθασμένες θετικές και αρνητικές προβλέψεις, αντίστοιχα. Μεγαλύτερη τιμή Accuracy υποδηλώνει καλύτερη συνολική επίδοση ταξινόμησης.

Μέση Ακρίβεια ανά Κλάση (Precision)

Η Precision μετρά το ποσοστό των σωστών θετικών προβλέψεων μεταξύ όλων των παρατηρήσεων που προβλέφθηκαν ως θετικές για μία συγκεκριμένη κλάση. Υπολογίζεται από τη σχέση::

$$Precision = \frac{TP}{TP + FP}$$

Στην περίπτωση προβλημάτων πολλαπλών κλάσεων, υπολογίζεται η Precision για κάθε κλάση ξεχωριστά και στη συνέχεια ο μέσος όρος τους, ώστε να προκύψει η Average Precision.

Μέση Ανάκληση (Average Recall)

Η Recall εκφράζει το ποσοστό των σωστά ταξινομημένων παρατηρήσεων μιας κλάσης σε σχέση με το σύνολο των παρατηρήσεων που πραγματικά ανήκουν στη συγκεκριμένη κλάση. Υπολογίζεται από τη σχέση:

$$Recall = \frac{TP}{TP + FN}$$

Για προβλήματα πολλαπλών κλάσεων, υπολογίζεται αντίστοιχα η Recall για κάθε κλάση και στη συνέχεια ο μέσος όρος τους, ώστε να προκύψει η Average Recall.

F1-Score

Το F1-Score αποτελεί συνδυαστική μετρική των Precision και Recall και υπολογίζεται ως ο αρμονικός μέσος τους:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Στα προβλήματα πολλαπλών κλάσεων υπολογίζεται το F1-Score για κάθε κλάση και στη συνέχεια χρησιμοποιείται ο μέσος όρος των επιμέρους τιμών. Υψηλότερη τιμή F1-Score υποδηλώνει καλύτερη ισορροπία μεταξύ Precision και Recall.

Fallback

Το Fallback χρησιμοποιείται στην υλοποίηση του Mutual k-NN στην περίπτωση κατά την οποία για μία εξεταζόμενη παρατήρηση δεν εντοπίζονται αμοιβαίοι γείτονες μεταξύ των k πλησιέστερων παρατηρήσεων. Σε αυτή την περίπτωση εφαρμόζεται εναλλακτική διαδικασία ταξινόμησης, η οποία βασίζεται στον πλησιέστερο γείτονα (1-NN), ώστε να είναι δυνατή η ολοκλήρωση της ταξινόμησης. Το ποσοστό Fallback εκφράζει το ποσοστό των παρατηρήσεων για τις οποίες χρειάστηκε να χρησιμοποιηθεί αυτή η εναλλακτική διαδικασία. Μικρότερη τιμή Fallback σημαίνει ότι ο αλγόριθμος μπόρεσε να πραγματοποιήσει την ταξινόμηση μέσω της κανονικής διαδικασίας Mutual k-NN σε μεγαλύτερο ποσοστό των παρατηρήσεων.

Συνολικά, οι παραπάνω μετρικές χρησιμοποιήθηκαν για κάθε σύνολο δεδομένων και για τις διαφορετικές τιμές του k , με σκοπό τη σύγκριση των αλγορίθμων ως προς την ποιότητα της ταξινόμησης, το υπολογιστικό κόστος και, στην περίπτωση του Mutual k -NN, τη συχνότητα χρήσης της εναλλακτικής διαδικασίας Fallback.

5.3 Πειραματικές μετρήσεις προεπεξεργασίας

Στην παρούσα ενότητα παρουσιάζονται τα αποτελέσματα που προέκυψαν από την εφαρμογή των τεχνικών μείωσης δεδομένων ENN, CNN και του συνδυασμού ENN→CNN στα είκοσι σύνολα δεδομένων της πειραματικής μελέτης. Στόχος είναι να αξιολογηθεί η αποτελεσματικότητα κάθε τεχνικής ως προς τη μείωση του αριθμού των διαθέσιμων παρατηρήσεων, αλλά και ως προς το υπολογιστικό κόστος που απαιτείται για την πραγματοποίηση της μείωσης.

Για κάθε σύνολο δεδομένων πραγματοποιήθηκε πενταπλή διαδικασία εκτέλεσης (5 fold), σύμφωνα με τη διαδικασία διασταυρωμένης επικύρωσης (cross validation) [13] που περιγράφεται στην Ενότητα 5.2. Για την αξιολόγηση των τεχνικών μείωσης χρησιμοποιήθηκαν ως βασικές μετρήσεις το μέσο πλήθος παρατηρήσεων που παρέμειναν μετά τη μείωση, το ποσοστό μείωσης και ο αριθμός υπολογισμών που απαιτήθηκε. Το ποσοστό μείωσης εκφράζει το ποσοστό των αρχικών παρατηρήσεων που αφαιρέθηκαν από το σύνολο δεδομένων.

Μείωση δεδομένων με ENN

Η πρώτη τεχνική που εξετάστηκε είναι η ENN (Edited Nearest Neighbour). Η τεχνική εξετάζει τις παρατηρήσεις του συνόλου εκπαίδευσης και απομακρύνει εκείνες που δεν ταξινομούνται στην ίδια κλάση με την πραγματική τους κλάση, σύμφωνα με τη διαδικασία που περιγράφηκε στο Κεφάλαιο 3. Στην υλοποίηση που χρησιμοποιήθηκε, η διαδικασία πραγματοποιήθηκε για τα πέντε folds κάθε συνόλου δεδομένων και στη συνέχεια υπολογίστηκε ο μέσος αριθμός παρατηρήσεων που παρέμειναν, το μέσο ποσοστό μείωσης και οι απαιτούμενοι υπολογισμοί.

Πίνακας 5.2: Αποτελέσματα της μείωσης δεδομένων με τη χρήση της τεχνικής ENN

Dataset	Μέσο πλήθος παρατηρήσεων	Computation	Reduction Rate (%)
Balance (bl)	411,8	124.251	17,640
Banana (bl)	3.745,2	8.986.680	11,670
Ecoli (ecl)	208,2	35.778	22,313
EEG Eye State (eeg)	10.234,2	71.802.136	14,601
KDD Cup (kdd)	112.854,203	6.405.252.096	0,291
Letter Recognition (lir)	15.302,8	127.992.000	4,358
Landsat Satellite (ls)	4.679,2	13.248.378	9,106

Dataset	Μέσο πλήθος παρατηρήσεων	Computation	Reduction Rate (%)
Magic Gamma Telescope (mgt)	12.624,8	115.755.720	17,029
MONK's Problems (mnk)	338,8	59.685	2,081
Pen-Based Digits	8.732,8	38.654.028	0,685
Phoneme (ph)	3.816,2	9.342.003	11,723
Pima Diabetes (pm)	448,6	188.191	26,938
Ringnorm (ring)	4.223,4	17.520.240	28,659
Segment (sg)	1.513,0	1.706.628	18,128
Shuttle (sh)	46.353,199	1.076.410.368	0,099
Twonorm (tn)	5.707,6	17.520.240	3,588
Texture (txr)	4.341,8	9.677.800	1,323
Waveform (wf)	3.220,2	7.998.000	19,495
Wine Quality (wine)	2.070,6	7.677.321	47,165
Yeast (ys)	646,4	703.891	45,543

Από τα παραπάνω αποτελέσματα παρατηρείται ότι η ENN παρουσιάζει αρκετά διαφορετική συμπεριφορά ανάλογα με τα χαρακτηριστικά του κάθε συνόλου δεδομένων. Το ποσοστό μείωσης δεν είναι σταθερό, αλλά μεταβάλλεται σημαντικά μεταξύ των διαφορετικών datasets. Αυτό είναι αναμενόμενο, καθώς η ENN βασίζεται στη σχέση μεταξύ των παρατηρήσεων και των γειτόνων τους και επομένως η αποτελεσματικότητά της επηρεάζεται από τον τρόπο με τον οποίο είναι κατανομημένες οι παρατηρήσεις και από τον βαθμό επικάλυψης μεταξύ των κλάσεων.

Τα μεγαλύτερα ποσοστά μείωσης παρατηρούνται στα σύνολα wine, ys, ring και pm, με ποσοστά 47,165%, 45,543%, 28,659% και 26,938%, αντίστοιχα. Επομένως, στα συγκεκριμένα σύνολα δεδομένων η ENN απομάκρυνε ένα σημαντικό μέρος των αρχικών παρατηρήσεων. Ειδικά στην περίπτωση του wine, το μέσο πλήθος παρατηρήσεων μειώθηκε σε 2.070,6, ενώ στο ys μειώθηκε σε 646,4 παρατηρήσεις. Τα αποτελέσματα αυτά δείχνουν ότι σε ορισμένες περιπτώσεις η ENN μπορεί να επιτύχει ουσιαστική μείωση του συνόλου εκπαίδευσης.

Αντίθετα, σε ορισμένα μεγαλύτερα σύνολα δεδομένων η μείωση ήταν ιδιαίτερα περιορισμένη. Χαρακτηριστικά παραδείγματα αποτελούν τα kdd και sh, στα οποία τα ποσοστά μείωσης ήταν μόλις 0,291% και 0,099%, αντίστοιχα. Στο kdd, από ένα μέσο πλήθος 113.184 περίπου παρατηρήσεων στο αρχικό σύνολο εκπαίδευσης παρέμειναν κατά μέσο όρο 112.854,203 παρατηρήσεις. Αντίστοιχα, στο sh παρέμειναν 46.353,199 παρατηρήσεις. Συνεπώς, η εφαρμογή της ENN σε αυτά τα δύο σύνολα δεν οδήγησε σε ουσιαστική μείωση του αριθμού των παρατηρήσεων.

Παρόμοια συμπεριφορά, αν και σε μικρότερο βαθμό, παρατηρείται και στα pd, mnk, tn και txr, όπου τα ποσοστά μείωσης είναι μικρότερα από 5%. Αυτό δείχνει ότι η ENN δεν οδηγεί απαραίτητα σε μεγάλη συμπίεση του συνόλου δεδομένων. Η τεχνική διατηρεί μεγάλο μέρος των παρατηρήσεων όταν αυτές δεν χαρακτηρίζονται ως παρατηρήσεις που μπορούν να απομακρυνθούν σύμφωνα με το κριτήριο της μεθόδου.

Ένα ακόμη σημαντικό στοιχείο είναι ο αριθμός των υπολογισμών που απαιτούνται. Επειδή η ENN εξετάζει τις παρατηρήσεις και πραγματοποιεί υπολογισμούς αποστάσεων μεταξύ παρατηρήσεων του συνόλου εκπαίδευσης, το υπολογιστικό κόστος αυξάνεται σημαντικά όταν αυξάνεται το μέγεθος του dataset. Αυτό γίνεται ιδιαίτερα εμφανές στο kdd, όπου απαιτήθηκαν περισσότεροι από 6,4 δισεκατομμύρια υπολογισμοί, ενώ στο sh απαιτήθηκαν περισσότεροι από 1 δισεκατομμύριο υπολογισμοί. Αντίθετα, στα μικρότερα datasets, όπως τα bl και ecl, ο αριθμός των απαιτούμενων υπολογισμών ήταν πολύ μικρότερος.

Επομένως, η ENN παρουσιάζει έναν σημαντικό συμβιβασμό μεταξύ της μείωσης του αριθμού των δεδομένων και του υπολογιστικού κόστους της ίδιας της διαδικασίας μείωσης. Σε ορισμένα σύνολα μπορεί να επιτύχει σημαντική μείωση, ενώ σε άλλα διατηρεί σχεδόν το σύνολο των αρχικών παρατηρήσεων. Παράλληλα, η διαδικασία υπολογισμού της μπορεί να είναι ιδιαίτερα απαιτητική για μεγάλα σύνολα δεδομένων.

Μείωση δεδομένων με CNN

Στη συνέχεια εφαρμόστηκε η τεχνική CNN (Condensed Nearest Neighbour). Σε αντίθεση με την ENN, η CNN δεν αποσκοπεί στην αφαίρεση παρατηρήσεων που θεωρούνται προβληματικές, αλλά στη δημιουργία ενός μικρότερου συνόλου αντιπροσωπευτικών παρατηρήσεων, οι οποίες μπορούν να χρησιμοποιηθούν ως prototypes για την ταξινόμηση.

Πίνακας 5.3: Αποτελέσματα της μείωσης δεδομένων με τη χρήση της τεχνικής CNN

Dataset	Μέσο πλήθος παρατηρήσεων	Computation	Reduction Rate (%)
Balance (bl)	170,8	183.969,594	65,840
Banana (bl)	959,8	12.095.480	77,363
Ecoli (ecl)	111,2	63.305,199	58,507
EEG Eye State (eeg)	4.062,0	146.134.016	66,105
KDD Cup (kdd)	992,8	475.607.456	99,123
Letter Recognition (lir)	2.701,4	166.004.672	83,116
Landsat Satellite (ls)	1.033,8	19.989.678	79,918
Magic Gamma Telescope (mgt)	5.464,8	266.736.992	64,085
MONK's Problems (mnk)	44,2	41.944	87,225
Pen-Based Digits	408,8	12.026.862	95,351
Phoneme (ph)	1.058,0	14.306.656	75,526
Pima Diabetes (pm)	309,2	387.071,594	49,642
Ringnorm (ring)	1.583,0	32.015.242	73,260
Segment (sg)	205,2	1.072.498,250	88,896

Dataset	Μέσο πλήθος παρατηρήσεων	Computation	Reduction Rate (%)
Shuttle (sh)	162,6	20.206.740	99,650
Twonorm (tn)	1.058,8	23.191.842	82,115
Texture (txr)	353,8	5.017.755,500	91,959
Waveform (wf)	1.626,4	20.358.028	59,340
Wine Quality (wine)	2.172,6	18.595.792	44,562
Yeast (ys)	794,4	1.464.261,375	33,075

Τα αποτελέσματα δείχνουν ότι η CNN επιτυγχάνει, γενικά, σημαντικά μεγαλύτερη μείωση σε σχέση με την ENN. Στα περισσότερα σύνολα δεδομένων το ποσοστό μείωσης ξεπερνά το 50%, γεγονός που σημαίνει ότι το σύνολο των prototypes αποτελεί μόνο ένα μέρος του αρχικού συνόλου εκπαίδευσης. Αυτό αποτελεί βασικό χαρακτηριστικό της CNN, καθώς ο στόχος της είναι να διατηρήσει ένα μικρότερο σύνολο αντιπροσωπευτικών παρατηρήσεων, αντί να διατηρήσει το μεγαλύτερο μέρος του αρχικού συνόλου.

Ιδιαίτερα υψηλά ποσοστά μείωσης παρατηρούνται στα kdd και sh, με 99,123% και 99,650%, αντίστοιχα. Στην περίπτωση του kdd, από περίπου 113 χιλιάδες παρατηρήσεις χρησιμοποιήθηκαν κατά μέσο όρο μόλις 992,8 prototypes. Αντίστοιχα, στο sh, από περίπου 46 χιλιάδες παρατηρήσεις παρέμειναν κατά μέσο όρο 162,6 prototypes. Επομένως, η CNN παρουσιάζει πολύ υψηλή ικανότητα συμπίκνωσης των συγκεκριμένων συνόλων δεδομένων. Τα αποτελέσματα αυτά είναι ιδιαίτερα σημαντικά, καθώς δείχνουν ότι μπορεί να διατηρηθεί ένα πολύ μικρό υποσύνολο των αρχικών δεδομένων για τη διαδικασία της ταξινόμησης.

Υψηλά ποσοστά μείωσης παρατηρούνται επίσης στα pd, txr, sg, mnk, lir, ls και tn. Για παράδειγμα, στο pd το ποσοστό μείωσης φτάνει το 95,351%, ενώ στο txr φτάνει το 91,959%. Αυτό σημαίνει ότι η CNN κατάφερε να περιορίσει σημαντικά τον αριθμό των παρατηρήσεων, διατηρώντας ένα σχετικά μικρό σύνολο prototypes. Στα συγκεκριμένα datasets η συμπίκνωση είναι επομένως ιδιαίτερα έντονη.

Από την άλλη πλευρά, υπάρχουν και datasets στα οποία η CNN παρουσιάζει μικρότερη αποτελεσματικότητα. Τα χαμηλότερα ποσοστά μείωσης παρατηρούνται στα ys με 33,075%, στο wine με 44,562% και στο pm με 49,642%. Παρότι και σε αυτές τις περιπτώσεις πραγματοποιείται μείωση, το τελικό σύνολο prototypes παραμένει σχετικά μεγαλύτερο σε σχέση με το αρχικό μέγεθος του dataset. Αυτό δείχνει ότι η αποτελεσματικότητα της CNN εξαρτάται από τη δομή του εκάστοτε συνόλου δεδομένων και από το πόσο καλά μπορεί να αντιπροσωπευθεί το σύνολο από ένα μικρό αριθμό prototypes.

Σε σχέση με το υπολογιστικό κόστος, η εφαρμογή της CNN απαιτεί επίσης σημαντικό αριθμό υπολογισμών. Για παράδειγμα, στο kdd απαιτήθηκαν κατά μέσο όρο περίπου 475,6 εκατομμύρια υπολογισμοί, ενώ στο mgt περίπου 266,7 εκατομμύρια και στο lir περίπου 166 εκατομμύρια. Αντίθετα, στα μικρότερα datasets, όπως το mnk και το ecl, το υπολογιστικό κόστος ήταν σαφώς χαμηλότερο.

Αξίζει να σημειωθεί ότι το υψηλό υπολογιστικό κόστος της διαδικασίας μείωσης δεν αναιρεί το όφελος που μπορεί να προκύψει στη συνέχεια. Η CNN μπορεί να απαιτεί αρκετούς υπολογισμούς κατά τη δημιουργία του συνόλου των prototypes, όμως μετά την ολοκλήρωση της μείωσης η ταξινόμηση μπορεί να πραγματοποιηθεί πάνω σε σημαντικά μικρότερο σύνολο παρατηρήσεων. Επομένως, η αξιολόγηση

της CNN δεν πρέπει να περιορίζεται μόνο στο κόστος της διαδικασίας μείωσης, αλλά πρέπει να εξετάζεται και σε συνδυασμό με το κόστος της ταξινόμησης που ακολουθεί.

Συνολικά, τα αποτελέσματα της CNN δείχνουν ότι η συγκεκριμένη τεχνική είναι ιδιαίτερα αποτελεσματική ως προς τη συμπίκνωση των δεδομένων. Σε αρκετά datasets επιτυγχάνεται μείωση μεγαλύτερη από 80% ή ακόμη και μεγαλύτερη από 90%. Επομένως, σε σύγκριση με την ENN, η CNN φαίνεται να έχει μεγαλύτερη δυνατότητα δημιουργίας ενός ιδιαίτερα μικρού συνόλου αντιπροσωπευτικών παρατηρήσεων.

Συνδυασμός ENN και CNN

Τέλος, εξετάστηκε ο συνδυασμός των δύο τεχνικών με τη σειρά ENN→CNN. Στην περίπτωση αυτή εφαρμόζεται αρχικά η ENN, ώστε να δημιουργηθεί ένα επεξεργασμένο σύνολο δεδομένων από το οποίο έχουν απομακρυνθεί παρατηρήσεις σύμφωνα με το κριτήριο της ENN. Στη συνέχεια, στο μειωμένο αυτό σύνολο εφαρμόζεται η CNN, με σκοπό την περαιτέρω συμπίκνωσή του και τη δημιουργία του τελικού συνόλου prototypes.

Ο συνδυασμός αυτός επιτρέπει την εφαρμογή δύο διαφορετικών διαδικασιών μείωσης σε διαδοχικά στάδια. Η ENN πραγματοποιεί αρχικά έναν καθαρισμό του συνόλου δεδομένων, ενώ η CNN επιχειρεί στη συνέχεια να περιορίσει ακόμη περισσότερο τον αριθμό των παρατηρήσεων. Τα αποτελέσματα της δεύτερης φάσης, δηλαδή της εφαρμογής της CNN πάνω στα σύνολα που έχουν ήδη επεξεργαστεί από την ENN, παρουσιάζονται παρακάτω:

Πίνακας 5.4: Αποτελέσματα της μείωσης δεδομένων με τη συνδυαστική χρήση ENN →CNN

Dataset	Μέσο πλήθος παρατηρήσεων	Computation	Reduction Rate (%)
Balance (bl)	57,4	64.904,398	86,034
Banana (bn)	197,4	2.221.717,250	94,709
Ecoli (ec1)	32,0	16.802,400	84,689
EEG Eye State (eeg)	2.007,4	73.712.184	80,079
KDD Cup (kdd)	422,4	163.765.216	99,626
Letter Recognition (lir)	1.971,0	117.284.944	87,108
Landsat Satellite (ls)	438,6	6.865.961	90,630
Magic Gamma Telescope (mgt)	1.444,0	65.005.236	88,551
MONK's Problems (mnk)	38,6	34.783	88,614
Pen-Based Digits	308,0	9.247.011	96,474
Phoneme (ph)	414,6	5.610.753	89,178
Pima Diabetes (pm)	86,0	104.252,602	80,543
Ringnorm (ring)	433,6	5.576.760	89,790
Segment (sg)	115,4	529.192,812	92,378

Dataset	Μέσο πλήθος παρατηρήσεων	Computation	Reduction Rate (%)
Shuttle (sh)	103,8	12.324.265	99,776
Twonorm (tn)	637,6	14.547.443	88,798
Texture (txr)	291,6	4.187.667,250	93,287
Waveform (wf)	721,8	7.715.034,500	77,598
Wine Quality (wine)	476,6	3.307.142,750	75,795
Yeast (ys)	177,8	253.642,406	72,938

Από τον προηγούμενο πίνακα προκύπτει ότι ο συνδυασμός ENN→CNN οδηγεί σε ιδιαίτερα μικρά τελικά σύνολα prototypes. Σε όλα τα datasets το ποσοστό μείωσης της δεύτερης φάσης, δηλαδή της CNN επί του ήδη επεξεργασμένου συνόλου ENN, είναι αρκετά υψηλό. Σε αρκετές περιπτώσεις ξεπερνά το 90%, γεγονός που δείχνει ότι η εφαρμογή της CNN μετά την ENN μπορεί να οδηγήσει σε πολύ έντονη περαιτέρω συμπίκνωση των δεδομένων.

Τα υψηλότερα ποσοστά μείωσης της CNN μετά την ENN παρατηρούνται στα sh, kdd, pd, bn, txr, sg και ls. Ειδικότερα, στο sh το ποσοστό μείωσης φτάνει το 99,776%, ενώ στο kdd φτάνει το 99,626%. Στο pd η αντίστοιχη τιμή είναι 96,474%, ενώ στα bn και txr είναι 94,709% και 93,287%, αντίστοιχα. Τα αποτελέσματα αυτά δείχνουν ότι, αφού προηγηθεί η επεξεργασία από την ENN, η CNN μπορεί να περιορίσει ακόμη περισσότερο το σύνολο των διαθέσιμων παρατηρήσεων.

Η διαφορά γίνεται ιδιαίτερα εμφανής αν εξεταστεί το kdd. Μετά την εφαρμογή της ENN, το μέσο μέγεθος του συνόλου ήταν 112.854,203 παρατηρήσεις, ενώ μετά την εφαρμογή της CNN στο συγκεκριμένο σύνολο προέκυψαν κατά μέσο όρο μόλις 422,4 prototypes. Επομένως, η διαδικασία δύο σταδίων οδηγεί σε ένα τελικό σύνολο που είναι πολύ μικρότερο από το αρχικό σύνολο εκπαίδευσης. Παρόμοια συμπεριφορά παρατηρείται και στο sh, όπου η ENN διατηρεί κατά μέσο όρο 46.353,199 παρατηρήσεις και η επόμενη εφαρμογή της CNN οδηγεί σε μόλις 103,8 prototypes.

Στα μικρότερα datasets παρατηρείται επίσης σημαντική συμπίκνωση. Για παράδειγμα, στο ecl ο μέσος αριθμός prototypes μετά τον συνδυασμό των δύο τεχνικών είναι μόλις 32,0, ενώ στο bl είναι 57,4 και στο pm 86,0. Αυτό δείχνει ότι ο συνδυασμός μπορεί να οδηγήσει σε πολύ μικρά σύνολα εκπαίδευσης ακόμη και όταν το αρχικό dataset περιέχει σχετικά λίγες παρατηρήσεις.

Σε σύγκριση με την εφαρμογή της CNN απευθείας στο αρχικό σύνολο, ο συνδυασμός ENN→CNN οδηγεί επίσης σε μικρότερο τελικό αριθμό prototypes για όλα τα εξεταζόμενα datasets. Για παράδειγμα, στο bl η CNN μόνη της οδηγεί κατά μέσο όρο σε 170,8 prototypes, ενώ μετά την προηγούμενη εφαρμογή της ENN ο αριθμός μειώνεται σε 57,4. Στο bn ο αριθμός μειώνεται από 959,8 σε 197,4 prototypes, ενώ στο ecl από 111,2 σε 32,0. Αντίστοιχη εικόνα παρατηρείται και στα μεγαλύτερα datasets, όπως το kdd, όπου ο αριθμός των prototypes μειώνεται από 992,8 στην απευθείας εφαρμογή της CNN σε 422,4 μετά τον συνδυασμό ENN→CNN.

Παρόμοια είναι η εικόνα και ως προς το υπολογιστικό κόστος της δεύτερης φάσης. Σε όλα τα εξεταζόμενα datasets, οι υπολογισμοί που απαιτούνται από την CNN μετά την ENN είναι μικρότεροι από τους αντίστοιχους υπολογισμούς της CNN όταν αυτή εφαρμόζεται απευθείας στο αρχικό σύνολο. Η διαφορά αυτή είναι ιδιαίτερα σημαντική στα μεγάλα datasets. Για παράδειγμα, στο kdd η απευθείας εφαρμογή της CNN απαιτεί κατά μέσο όρο 475.607.456 υπολογισμούς, ενώ μετά την εφαρμογή της ENN απαιτούνται κατά μέσο όρο 163.765.216 υπολογισμοί για τη δεύτερη φάση της CNN. Αντίστοιχα, στο mgt οι υπολογισμοί της CNN μειώνονται από 266.736.992 σε 65.005.236.

Ωστόσο, κατά την ερμηνεία του Πίνακα 5.4 πρέπει να διευκρινιστεί ότι το ποσοστό μείωσης που εμφανίζεται στην τελευταία στήλη αφορά τη μείωση που πραγματοποιεί η CNN πάνω στο σύνολο που έχει ήδη προκύψει από την ENN. Επομένως, δεν αποτελεί από μόνο του το συνολικό ποσοστό μείωσης από το αρχικό dataset έως το τελικό σύνολο prototypes. Η συνολική μείωση είναι αποτέλεσμα και των δύο διαδοχικών σταδίων. Η διάκριση αυτή είναι σημαντική, καθώς διαφορετικά θα μπορούσε να θεωρηθεί ότι, για παράδειγμα, το 99,626% του kdd αντιστοιχεί στη συνολική μείωση από το αρχικό σύνολο, ενώ στην πραγματικότητα αφορά τη δεύτερη φάση της διαδικασίας, δηλαδή την εφαρμογή της CNN μετά την ENN.

Συνολικά, ο συνδυασμός ENN→CNN παρουσιάζει την πιο έντονη συμπίκνωση των δεδομένων από τις τρεις εξεταζόμενες διαδικασίες. Η ENN προηγείται και περιορίζει το σύνολο των δεδομένων, ενώ η CNN χρησιμοποιεί το ήδη επεξεργασμένο σύνολο για να δημιουργήσει ένα ακόμη μικρότερο σύνολο prototypes. Ως αποτέλεσμα, σε αρκετές περιπτώσεις ο τελικός αριθμός παρατηρήσεων είναι πολύ μικρός σε σχέση με το αρχικό μέγεθος του dataset. Παράλληλα, η μικρότερη είσοδος στη δεύτερη φάση έχει ως αποτέλεσμα να μειώνεται και το υπολογιστικό κόστος της CNN σε σχέση με την απευθείας εφαρμογή της στο αρχικό σύνολο.

Συγκριτική αξιολόγηση των τεχνικών μείωσης δεδομένων

Με βάση τα αποτελέσματα των προηγούμενων ενοτήτων, προκύπτουν σημαντικές διαφορές μεταξύ των τριών διαδικασιών μείωσης δεδομένων. Η ENN εμφανίζει γενικά πιο περιορισμένη μείωση του αριθμού των παρατηρήσεων, καθώς σε αρκετά datasets διατηρεί το μεγαλύτερο μέρος του αρχικού συνόλου. Αντίθετα, η CNN παρουσιάζει σαφώς μεγαλύτερη ικανότητα συμπίκνωσης, καθώς στα περισσότερα datasets το ποσοστό μείωσης είναι σημαντικά υψηλότερο. Ο συνδυασμός ENN→CNN οδηγεί σε ακόμη μικρότερα σύνολα, καθώς η CNN εφαρμόζεται σε δεδομένα που έχουν ήδη μειωθεί από την ENN.

Η συμπεριφορά της ENN φαίνεται να εξαρτάται σε μεγάλο βαθμό από το εκάστοτε dataset. Ενδεικτικά, στο wine και στο ys επιτεύχθηκαν ποσοστά μείωσης 47,165% και 45,543%, αντίστοιχα, ενώ στο kdd και στο sh τα αντίστοιχα ποσοστά ήταν μόλις 0,291% και 0,099%. Συνεπώς, η ENN δεν μπορεί να θεωρηθεί μια τεχνική που επιτυγχάνει σταθερό επίπεδο μείωσης ανεξάρτητα από τα χαρακτηριστικά των δεδομένων.

Η CNN παρουσιάζει πιο σταθερή συμπεριφορά ως προς τη δυνατότητα συμπίκνωσης. Ακόμη και στα datasets όπου το ποσοστό μείωσης είναι σχετικά χαμηλότερο, όπως το ys και το wine, η μείωση εξακολουθεί να είναι σημαντική, με τιμές 33,075% και 44,562%, αντίστοιχα. Στα kdd και sh, η μείωση ξεπερνά το 99%, γεγονός που καταδεικνύει τη δυνατότητα της μεθόδου να αντικαταστήσει ένα πολύ μεγάλο σύνολο δεδομένων από ένα μικρό σύνολο prototypes.

Ο συνδυασμός ENN→CNN παρουσιάζει την πιο έντονη συμπίκνωση. Η προηγούμενη εφαρμογή της ENN μειώνει αρχικά το σύνολο, ενώ στη συνέχεια η CNN περιορίζει περαιτέρω τις παρατηρήσεις. Ως αποτέλεσμα, ο αριθμός των τελικών prototypes είναι μικρότερος από τον αντίστοιχο αριθμό που προκύπτει όταν η CNN εφαρμόζεται απευθείας στα αρχικά δεδομένα. Η συμπεριφορά αυτή είναι ιδιαίτερα εμφανής στα μεγάλα datasets, όπου ακόμη και μια μικρή αρχική μείωση από την ENN μπορεί να οδηγήσει σε μικρότερη είσοδο για τη δεύτερη φάση της CNN και επομένως σε σημαντικά μικρότερο τελικό σύνολο.

Ένα δεύτερο σημαντικό συμπέρασμα αφορά το υπολογιστικό κόστος. Η εφαρμογή των τεχνικών μείωσης απαιτεί η ίδια υπολογιστική προσπάθεια, η οποία μπορεί να είναι ιδιαίτερα μεγάλη όταν το αρχικό σύνολο δεδομένων περιέχει μεγάλο αριθμό παρατηρήσεων. Η ENN, για παράδειγμα, απαιτεί περισσότερους από 6,4 δισεκατομμύρια υπολογισμούς στο kdd, ενώ η CNN απαιτεί περίπου 475,6 εκατομμύρια. Παρότι οι απόλυτες τιμές δεν είναι άμεσα συγκρίσιμες ως προς τον τρόπο υπολογισμού τους μεταξύ των διαφορετικών υλοποιήσεων, καταδεικνύουν το σημαντικό υπολογιστικό κόστος που μπορεί να έχει η διαδικασία προεπεξεργασίας.

Παρά το κόστος αυτό, η σημαντική μείωση του μεγέθους του συνόλου εκπαίδευσης μπορεί να προσφέρει σημαντικό όφελος στο επόμενο στάδιο, δηλαδή στην ταξινόμηση με χρήση των αλγορίθμων k-NN, Mutual k-NN και Symmetric k-NN. Ειδικά για τους αλγορίθμους που βασίζονται στον υπολογισμό αποστάσεων μεταξύ των παρατηρήσεων, η χρήση ενός μικρότερου συνόλου εκπαίδευσης μπορεί να περιορίσει τον αριθμό των απαιτούμενων υπολογισμών κατά την ταξινόμηση. Για τον λόγο αυτό, η αξιολόγηση των τεχνικών μείωσης δεν μπορεί να βασιστεί αποκλειστικά στο ποσοστό μείωσης, αλλά πρέπει να εξεταστεί σε συνδυασμό με την απόδοση και το υπολογιστικό κόστος της ταξινόμησης που ακολουθεί.

Συνεπώς, τα αποτελέσματα του παρόντος σταδίου δείχνουν ότι οι τεχνικές ENN, CNN και ENN→CNN παρουσιάζουν διαφορετικά χαρακτηριστικά και διαφορετικό βαθμό συμπίκνωσης των δεδομένων. Η ENN είναι περισσότερο επιλεκτική ως προς τις παρατηρήσεις που απομακρύνει, ενώ η CNN επιτυγχάνει γενικά μεγαλύτερη μείωση μέσω της δημιουργίας ενός συνόλου αντιπροσωπευτικών prototypes. Ο συνδυασμός των δύο τεχνικών οδηγεί στην ισχυρότερη συμπίκνωση, καθώς πραγματοποιούνται δύο διαδοχικά στάδια μείωσης. Η τελική αξιολόγηση, ωστόσο, θα πρέπει να βασιστεί όχι μόνο στο μέγεθος των μειωμένων συνόλων, αλλά και στην επίδραση που έχει κάθε τεχνική στην ακρίβεια της ταξινόμησης και στον χρόνο εκτέλεσης, τα οποία εξετάζονται στις επόμενες ενότητες του πειραματικού μέρους.

5.4 Πειραματικές μετρήσεις κατηγοριοποίησης

Στην ενότητα αυτή παρουσιάζονται οι πειραματικές μετρήσεις στα 20 datasets για κάθε διαφορετικό k (1,3,5,7) για τους αλγορίθμους k-NN, Mutual k-NN, Symmetric k-NN καθώς και μετά τη χρήση τεχνικών μείωσης δεδομένων, δηλαδή ENN, CNN και ENN σε CNN.

Πίνακας 5.5: Σύγκριση μεθόδων χωρίς μείωση δεδομένων για k=1:

Dataset	Μετρική	KNN	Mutual	Symmetric
	CPU Time	0	0,1719	0,0844
	Accuracy	77,55	77,55	85,46

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Balance (bl)	Avg Precision	77,26	77,26	82,27
	Avg Recall	77,55	77,55	85,46
	F1-Score	77,31	77,31	83,62
	Fallback	-	100,00	-
Banana (bn)	CPU Time	0,003125	1,3438	0,6562
	Accuracy	87,06	87,06	87,40
	Avg Precision	87,07	87,07	87,42
	Avg Recall	87,06	87,06	87,40
	F1-Score	87,06	87,06	87,36
	Fallback	-	99,96	-
	CPU Time	0,00625	0,1031	0,0406
	Accuracy	78,48	78,48	78,79
Ecoli (ecl)	Avg Precision	79,75	79,75	79,67
	Avg Recall	78,48	78,48	78,79
	F1-Score	78,58	78,58	78,76
	Fallback	-	100,00	-
	CPU Time	0,31875	4,1188	5,3906
	Accuracy	44,62	44,62	46,50
Eye State (eeg)	Avg Precision	60,93	60,93	60,99
	Avg Recall	44,62	44,62	46,50
	F1-Score	46,80	46,80	48,55
	Fallback	-	100,00	-
	CPU Time	24,596875	1069,0312	963,975
	Accuracy	99,76	99,76	99,68
KDD Cup (kdd)	Avg Precision	99,76	99,76	99,68
	Avg Recall	99,76	99,76	99,68
	F1-Score	99,76	99,76	99,68
	Fallback	-	100,00	-
	CPU Time	0,696875	31,7875	11
	Accuracy	95,09	95,09	95,38
Letter	Avg Precision	95,13	95,13	95,44

Dataset	Μετρική	KNN	Mutual	Symmetric
Recognition (lir)	Avg Recall	95,09	95,09	95,38
	F1-Score	95,09	95,09	95,39
	Fallback	-	100,00	-
Landsat Satellite (ls)	CPU Time	0,165625	9,575	3,2344
	Accuracy	90,17	90,17	89,89
	Avg Precision	90,46	90,46	90,18
	Avg Recall	90,17	90,17	89,89
	F1-Score	90,27	90,27	89,96
	Fallback	-	100,00	-
Magic Gamma Telescope (mgt)	CPU Time	0,190625	4,9125	6,3406
	Accuracy	81,49	81,49	81,02
	Avg Precision	81,22	81,22	80,74
	Avg Recall	81,49	81,49	81,02
	F1-Score	81,16	81,16	80,57
	Fallback	-	100,00	-
MONK's Problems (mnk)	CPU Time	0,003125	0,15	0,0625
	Accuracy	73,30	73,30	99,53
	Avg Precision	74,11	74,11	99,54
	Avg Recall	73,30	73,30	99,53
	F1-Score	73,03	73,03	99,53
	Fallback	-	100,00	-
Pen-Based Digits (pd)	CPU Time	0,28125	17,0813	8,3094
	Accuracy	99,37	99,37	99,14
	Avg Precision	99,37	99,37	99,14
	Avg Recall	99,37	99,37	99,14
	F1-Score	99,37	99,37	99,14
	Fallback	-	100,00	-
Phoneme (ph)	CPU Time	0,01875	1,4781	0,7281
	Accuracy	90,11	90,11	89,63
	Avg Precision	90,02	90,02	89,50
	Avg Recall	90,11	90,11	89,63

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Pima Diabetes (pm)	F1-Score	90,03	90,03	89,42
	Fallback	-	100,00	-
	CPU Time	0,003125	0,2	0,0938
	Accuracy	70,21	70,21	70,47
	Avg Precision	69,87	69,87	69,38
	Avg Recall	70,21	70,21	70,47
	F1-Score	69,96	69,96	69,60
	Fallback	-	100,00	-
	CPU Time	0,140625	10,9281	2,725
	Accuracy	74,36	74,36	70,3
	Avg Precision	82,64	82,64	70,40
	Avg Recall	74,36	74,36	70,30
Ringnorm (ring)	F1-Score	72,54	72,54	70,28
	Fallback	-	100,00	-
	CPU Time	0,046875	2,8469	0,9812
	Accuracy	96,14	96,14	96,22
	Avg Precision	96,20	96,20	96,26
	Avg Recall	96,14	96,14	96,22
Segment (sg)	F1-Score	96,12	96,12	96,20
	Fallback	-	84,85	-
	CPU Time	0,746875	16,575	41,5156
	Accuracy	99,89	99,89	99,93
	Avg Precision	99,89	99,89	99,93
	Avg Recall	99,89	99,89	99,93
Shuttle (sh)	F1-Score	99,89	99,89	99,93
	Fallback	-	100,00	-
	CPU Time	0,16875	10,725	2,4937
	Accuracy	94,58	94,58	94,91
	Avg Precision	94,59	94,59	94,95
	Avg Recall	94,58	94,58	94,91
Twonorm (tn)	F1-Score	94,58	94,58	94,91

Dataset	Μετρική	KNN	Mutual	Symmetric
	Fallback	-	100,00	-
Texture (txr)	CPU Time	0,1875	8,7969	2,9031
	Accuracy	98,85	98,85	98,65
	Avg Precision	98,86	98,86	98,67
	Avg Recall	98,85	98,85	98,65
	F1-Score	98,85	98,85	98,65
	Fallback	-	100,00	-
Waveform (wf)	CPU Time	0,090625	7,5344	1,6625
	Accuracy	76,4	76,4	78,2
	Avg Precision	76,42	76,42	78,41
	Avg Recall	76,40	76,40	78,20
	F1-Score	76,35	76,35	78,22
	Fallback	-	100,00	-
Wine Quality (wine)	CPU Time	0,06875	1,3687	0,7688
	Accuracy	45,13	45,13	45,15
	Avg Precision	45,27	45,27	45,28
	Avg Recall	45,13	45,13	45,15
	F1-Score	45,00	45,00	44,86
	Fallback	-	99,80	-
Yeast (ys)	CPU Time	0,009375	0,4313	0,2
	Accuracy	51,15	51,15	50,74
	Avg Precision	52,42	52,42	52,02
	Avg Recall	51,15	51,15	50,74
	F1-Score	51,35	51,35	50,58
	Fallback	-	100,00	-

Για $k=1$, οι k -NN και Mutual k -NN παρουσιάζουν ίδια αποτελέσματα ταξινόμησης, καθώς στην περίπτωση του Mutual k -NN η διαδικασία οδηγεί σε χρήση του πλησιέστερου γείτονα μέσω Fallback, το οποίο εμφανίζεται στις περισσότερες περιπτώσεις στο 100%. Ο Symmetric k -NN παρουσιάζει καλύτερη ακρίβεια σε αρκετά σύνολα, με χαρακτηριστικό παράδειγμα το Monk, όπου επιτυγχάνει 99,53%, έναντι 73,30% των k -NN και Mutual k -NN.

Ως προς τον CPU Time, ο k-NN είναι σταθερά ο ταχύτερος, ενώ οι Mutual και Symmetric απαιτούν περισσότερο χρόνο, ιδιαίτερα στα μεγαλύτερα σύνολα δεδομένων.

Πίνακας 5.6: Σύγκριση μεθόδων χωρίς μείωση δεδομένων για k=3:

Dataset	Μετρική	KNN	Mutual	Symmetric
Balance (bl)	CPU Time	0,0031	0,0281	0,0813
	Accuracy	83,53	79,49	86,43
	Avg Precision	80,37	78,50	82,56
	Avg Recall	83,53	79,49	86,43
	F1-Score	81,77	78,43	84,24
	Fallback	-	1,94	-
Banana (bn)	CPU Time	0,0094	0,1594	0,6125
	Accuracy	88,36	89,38	88,97
	Avg Precision	88,37	89,46	89,01
	Avg Recall	88,36	89,38	88,97
	F1-Score	88,35	89,34	88,94
	Fallback	-	0,23	-
Ecoli (ecl)	CPU Time	0	0,0344	0,0437
	Accuracy	79,09	78,79	80,61
	Avg Precision	78,97	79,64	80,42
	Avg Recall	79,09	78,79	80,61
	F1-Score	78,22	78,71	79,81
	Fallback	-	17,27	-
Eye State (eeg)	CPU Time	0,35	3,6063	5,6437
	Accuracy	43,71	45,03	44,95
	Avg Precision	60,58	61,11	60,54
	Avg Recall	43,71	45,03	44,95
	F1-Score	45,60	46,83	46,63
	Fallback	-	63,13	-

Dataset	Μετρική	KNN	Mutual	Symmetric
KDD Cup (kdd)	CPU Time	23,5531	280,8906	1117,1937
	Accuracy	99,76	99,74	99,71
	Avg Precision	99,74	99,74	99,69
	Avg Recall	99,76	99,74	99,71
	F1-Score	99,74	99,73	99,70
	Fallback	-	14,95	-
Letter Recognition (lir)	CPU Time	0,6625	12,8313	11,1813
	Accuracy	94,55	94,58	95,63
	Avg Precision	94,66	94,64	95,72
	Avg Recall	94,55	94,58	95,63
	F1-Score	94,56	94,59	95,64
	Fallback	-	23,26	-
Landsat Satellite (ls)	CPU Time	0,1812	4,4562	3,1781
	Accuracy	90,68	90,4	90,87
	Avg Precision	90,80	90,57	90,86
	Avg Recall	90,68	90,40	90,87
	F1-Score	90,70	90,44	90,81
	Fallback	-	32,28	-
Magic Gamma Telescope (mgt)	CPU Time	0,2469	2,5531	6,5687
	Accuracy	83,22	82,21	83,43
	Avg Precision	83,18	82,00	83,35
	Avg Recall	83,22	82,21	83,43
	F1-Score	82,71	81,79	82,97
	Fallback	-	23,64	-
MONK's Problems (mnk)	CPU Time	0,0031	0,0187	0,0531
	Accuracy	94,61	79,64	99,29
	Avg Precision	94,68	81,13	99,32
	Avg Recall	94,61	79,64	99,29
	F1-Score	94,61	79,51	99,29
	Fallback	-	1,87	-
	CPU Time	0,3906	6,4781	8,1844

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Pen-Based Digits (pd)	Accuracy	99,37	99,32	99,3
	Avg Precision	99,37	99,32	99,30
	Avg Recall	99,37	99,32	99,30
	F1-Score	99,37	99,32	99,30
	Fallback	-	23,16	-
Phoneme (ph)	CPU Time	0,0156	0,4813	0,6969
	Accuracy	88,26	89,53	87,9
	Avg Precision	88,13	89,41	87,72
	Avg Recall	88,26	89,53	87,90
	F1-Score	88,15	89,38	87,69
	Fallback	-	15,78	-
Pima Diabetes (pm)	CPU Time	0,0063	0,0813	0,1062
	Accuracy	74,15	69,69	74,02
	Avg Precision	73,67	68,78	73,29
	Avg Recall	74,15	69,69	74,02
	F1-Score	73,74	68,94	73,27
	Fallback	-	23,88	-
Ringnorm (ring)	CPU Time	0,1469	7,8438	2,6469
	Accuracy	70,87	75,81	74,42
	Avg Precision	81,54	82,99	75,31
	Avg Recall	70,87	75,81	74,42
	F1-Score	68,06	74,34	74,16
	Fallback	-	64,76	-
Segment (sg)	CPU Time	0,0437	1,1219	0,975
	Accuracy	95,18	96,09	95,75
	Avg Precision	95,28	96,16	95,77
	Avg Recall	95,18	96,09	95,75
	F1-Score	95,13	96,08	95,71
	Fallback	-	19,36	-
	CPU Time	1,2125	7,5594	40,1656
	Accuracy	99,89	99,9	99,9

Dataset	Μετρική	KNN	Mutual	Symmetric
Shuttle (sh)	Avg Precision	99,89	99,90	99,90
	Avg Recall	99,89	99,90	99,90
	F1-Score	99,88	99,89	99,89
	Fallback	-	10,05	-
Twonorm (tn)	CPU Time	0,1656	6,1344	2,4
	Accuracy	96,5	94,78	96,47
	Avg Precision	96,50	94,81	96,48
	Avg Recall	96,50	94,78	96,47
	F1-Score	96,50	94,78	96,47
	Fallback	-	46,09	-
	CPU Time	0,1344	2,8781	2,9125
	Accuracy	98,53	98,78	98,74
Texture (txr)	Avg Precision	98,54	98,78	98,76
	Avg Recall	98,53	98,78	98,74
	F1-Score	98,53	98,78	98,75
	Fallback	-	18,87	-
	CPU Time	0,075	3,7625	1,5969
	Accuracy	79,38	76,24	80,54
Waveform (wf)	Avg Precision	79,43	76,37	80,61
	Avg Recall	79,38	76,24	80,54
	F1-Score	79,29	76,25	80,54
	Fallback	-	41,70	-
	CPU Time	0,0719	0,9031	0,8344
	Accuracy	45,09	44,62	47,09
Wine Quality (wine)	Avg Precision	44,78	44,53	46,18
	Avg Recall	45,09	44,62	47,09
	F1-Score	44,70	44,41	46,15
	Fallback	-	50,30	-
	CPU Time	0	0,15	0,2313
	Accuracy	52,84	50,07	54,06
Yeast (ys)	Avg Precision	53,51	50,96	54,16

Dataset	Μετρική	KNN	Mutual	Symmetric
	Avg Recall	52,84	50,07	54,06
	F1-Score	52,44	49,90	53,43
	Fallback	-	23,21	-

Για $k=3$, ο Symmetric k-NN παρουσιάζει πολύ καλή απόδοση σε αρκετά σύνολα δεδομένων, με σημαντικότερο παράδειγμα το Monk, όπου επιτυγχάνει 99,29% έναντι 94,61% του k-NN και 79,64% του Mutual k-NN. Ο Mutual k-NN υπερέχει σε ορισμένα σύνολα, όπως το Phoneme και το Ring, αλλά παρουσιάζει υψηλά ποσοστά Fallback, φτάνοντας στο Ring το 64,76%.

Ο k-NN παραμένει ο ταχύτερος αλγόριθμος, ενώ οι δύο παραλλαγές έχουν αυξημένο υπολογιστικό κόστος.

Πίνακας 5.7: Σύγκριση μεθόδων χωρίς μείωση δεδομένων για $k = 5$:

Dataset	Μετρική	KNN	Mutual	Symmetric
Balance (bl)	CPU Time	0,0063	0,0219	0,0688
	Accuracy	85,62	82,55	88,21
	Avg Precision	82,95	80,81	83,46
	Avg Recall	85,62	82,55	88,21
	F1-Score	84,04	81,36	85,01
	Fallback	-	0,00	-
Banana (bn)	CPU Time	0,0063	0,2031	0,6156
	Accuracy	89,1	88,44	89,35
	Avg Precision	89,11	88,52	89,40
	Avg Recall	89,10	88,44	89,35
	F1-Score	89,09	88,39	89,32
	Fallback	-	1,11	-
Ecoli (ec1)	CPU Time	0,0063	0,0156	0,0406
	Accuracy	80,61	78,18	79,7
	Avg Precision	80,94	78,74	79,61
	Avg Recall	80,61	78,18	79,70
	F1-Score	79,67	77,67	78,61
	Fallback	-	2,73	-
	CPU Time	0,375	3,1875	5,7344

Dataset	Μετρική	KNN	Mutual	Symmetric
Eye State (eeg)	Accuracy	42,59	44,65	44,41
	Avg Precision	59,72	60,75	60,58
	Avg Recall	42,59	44,65	44,41
	F1-Score	44,29	46,17	45,90
	Fallback	-	45,59	-
KDD Cup (kdd)	CPU Time	22,4531	175,4844	1125,5938
	Accuracy	99,76	99,77	99,69
	Avg Precision	99,74	99,76	99,66
	Avg Recall	99,76	99,77	99,69
	F1-Score	99,74	99,76	99,67
	Fallback	-	5,04	-
Letter Recognition (lir)	CPU Time	0,7094	7,9562	11,5062
	Accuracy	94,55	94,56	95,48
	Avg Precision	94,64	94,64	95,56
	Avg Recall	94,55	94,56	95,48
	F1-Score	94,56	94,57	95,49
	Fallback	-	7,18	-
Landsat Satellite (ls)	CPU Time	0,1844	3,2719	3,2188
	Accuracy	90,5	90,65	90,93
	Avg Precision	90,57	90,82	90,83
	Avg Recall	90,50	90,65	90,93
	F1-Score	90,49	90,68	90,82
	Fallback	-	14,79	-
Magic Gamma Telescope (mgt)	CPU Time	0,2969	2,1031	6,5625
	Accuracy	83,55	82,92	84,07
	Avg Precision	83,75	82,81	84,06
	Avg Recall	83,55	82,92	84,07
	F1-Score	82,91	82,45	83,61
	Fallback	-	8,20	-
	CPU Time	0	0,0187	0,0563
	Accuracy	92,98	95,55	98,12

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
MONK's	Avg Precision	93,21	95,77	98,26
Problems (mnk)	Avg Recall	92,98	95,55	98,12
	F1-Score	92,96	95,55	98,12
	Fallback	-	0,00	-
	CPU Time	0,375	4,3344	8,2625
Pen-Based Digits (pd)	Accuracy	99,27	99,32	99,28
	Avg Precision	99,28	99,32	99,29
	Avg Recall	99,27	99,32	99,28
	F1-Score	99,27	99,32	99,28
Phoneme (ph)	Fallback	-	8,05	-
	CPU Time	0,0156	0,2656	0,7063
	Accuracy	87,48	88,57	87,27
	Avg Precision	87,35	88,40	87,16
	Avg Recall	87,48	88,57	87,27
	F1-Score	87,35	88,36	87,13
Pima Diabetes (pm)	Fallback	-	4,54	-
	CPU Time	0,0063	0,0406	0,1062
	Accuracy	74,02	72,18	72,84
	Avg Precision	73,55	71,35	71,99
	Avg Recall	74,02	72,18	72,84
	F1-Score	73,43	71,26	72,04
Ringnorm (ring)	Fallback	-	7,61	-
	CPU Time	0,1469	6,5594	3,3031
	Accuracy	68,37	77,58	72,82
	Avg Precision	80,56	83,46	74,53
	Avg Recall	68,37	77,58	72,82
	F1-Score	64,72	76,49	72,29
Segment (sg)	Fallback	-	49,84	-
	CPU Time	0,0375	0,7844	0,9594
	Accuracy	94,18	95,92	95,14
	Avg Precision	94,27	95,98	95,23

Dataset	Μετρική	KNN	Mutual	Symmetric
	Avg Recall	94,18	95,92	95,14
	F1-Score	94,10	95,90	95,12
	Fallback	-	5,69	-
Shuttle (sh)	CPU Time	1,4594	8,9125	47,0063
	Accuracy	99,86	99,9	99,88
	Avg Precision	99,87	99,90	99,88
	Avg Recall	99,86	99,90	99,88
	F1-Score	99,86	99,90	99,87
	Fallback	-	3,30	-
Twonorm (tn)	CPU Time	0,1719	3,6719	3,0031
	Accuracy	96,78	95,25	97,07
	Avg Precision	96,78	95,29	97,07
	Avg Recall	96,78	95,25	97,07
	F1-Score	96,78	95,25	97,07
	Fallback	-	17,66	-
Texture (txr)	CPU Time	0,1844	1,7563	2,9531
	Accuracy	98,11	98,63	98,49
	Avg Precision	98,13	98,65	98,50
	Avg Recall	98,11	98,63	98,49
	F1-Score	98,11	98,64	98,49
	Fallback	-	5,11	-
Waveform (wf)	CPU Time	0,1062	2,6187	1,6125
	Accuracy	81,02	76,82	82,64
	Avg Precision	81,10	77,00	82,66
	Avg Recall	81,02	76,82	82,64
	F1-Score	80,94	76,84	82,62
	Fallback	-	22,66	-
Wine Quality (wine)	CPU Time	0,0719	0,7625	0,8688
	Accuracy	46,93	46,17	47,25
	Avg Precision	46,01	45,74	45,79
	Avg Recall	46,93	46,17	47,25

Dataset	Μετρική	KNN	Mutual	Symmetric
Yeast (ys)	F1-Score	46,86	45,67	45,72
	Fallback	-	27,04	-
	CPU Time	0,0156	0,1	0,2094
	Accuracy	55,34	50,27	57,71
	Avg Precision	55,59	50,45	57,46
	Avg Recall	55,34	50,27	57,71
	F1-Score	54,90	49,82	57,17
	Fallback	-	8,19	-

Για $k=5$, ο Symmetric k -NN παρουσιάζει υψηλή ακρίβεια σε αρκετά σύνολα, όπως τα Balance, Letter, Monk και Waveform. Στο Monk επιτυγχάνει 98,12%, ενώ ο k -NN και ο Mutual k -NN 92,98% και 95,55%, αντίστοιχα. Ο Mutual k -NN υπερέχει σε ορισμένες περιπτώσεις, όπως το Ring, όπου φτάνει το 77,58%, αλλά με υψηλό Fallback (49,84%).

Ο k -NN εξακολουθεί να παρουσιάζει το μικρότερο CPU Time, με τη διαφορά να αυξάνεται στα μεγαλύτερα σύνολα δεδομένων.

Πίνακας 5.8: Σύγκριση μεθόδων χωρίς μείωση δεδομένων για $k=7$:

Dataset	Μετρική	KNN	Mutual	Symmetric
Balance (bl)	CPU Time	0,0031	0,0281	0,0688
	Accuracy	88,04	86,43	90,14
	Avg Precision	83,83	82,68	85,09
	Avg Recall	88,04	86,43	90,14
	F1-Score	85,06	84,30	86,76
	Fallback	-	0,00	-
Banana (bn)	CPU Time	0,0031	0,1625	0,6188
	Accuracy	89,57	89,38	89,46
	Avg Precision	89,57	89,46	89,51
	Avg Recall	89,57	89,38	89,46
	F1-Score	89,56	89,34	89,43
	Fallback	-	0,23	-
	CPU Time	0	0,0156	0,0406
	Accuracy	80,91	80,91	80,61

Dataset	Μετρική	KNN	Mutual	Symmetric
Ecoli (ec1)	Avg Precision	81,97	81,54	80,85
	Avg Recall	80,91	80,91	80,61
	F1-Score	80,05	80,41	79,54
	Fallback	-	1,52	-
Eye State (eeg)	CPU Time	0,4031	3,0531	5,825
	Accuracy	42,27	44,73	44,19
	Avg Precision	59,28	61,03	60,49
	Avg Recall	42,27	44,73	44,19
	F1-Score	43,88	46,05	45,56
	Fallback	-	36,15	-
	CPU Time	23,0594	151,6844	1104,275
	Accuracy	99,72	99,77	99,67
KDD Cup (kdd)	Avg Precision	99,69	99,77	99,64
	Avg Recall	99,72	99,77	99,67
	F1-Score	99,70	99,77	99,65
	Fallback	-	2,81	-
	CPU Time	0,7063	7,1844	11,7594
Letter Recognition (lir)	Accuracy	94,22	94,6	95,13
	Avg Precision	94,34	94,67	95,22
	Avg Recall	94,22	94,60	95,13
	F1-Score	94,23	94,61	95,14
	Fallback	-	2,82	-
	CPU Time	0,2031	2,5094	3,2594
Landsat Satellite (ls)	Accuracy	90,31	90,9	90,99
	Avg Precision	90,28	90,99	90,86
	Avg Recall	90,31	90,90	90,99
	F1-Score	90,25	90,87	90,85
	Fallback	-	8,31	-
Magic Gamma	CPU Time	0,3312	1,9219	6,7156
	Accuracy	83,94	83,55	84,17
	Avg Precision	84,36	83,52	84,19

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Telescope (mgt)	Avg Recall	83,94	83,55	84,17
	F1-Score	83,23	83,07	83,71
	Fallback	-	3,67	-
MONK's Problems (mnk)	CPU Time	0	0,025	0,0531
	Accuracy	90,16	94,61	96,95
	Avg Precision	90,73	95,19	97,17
	Avg Recall	90,16	94,61	96,95
	F1-Score	90,11	94,59	96,95
	Fallback	-	0,00	-
	CPU Time	0,3906	3,7313	8,1375
Pen-Based Digits (pd)	Accuracy	99,17	99,37	99,18
	Avg Precision	99,18	99,37	99,19
	Avg Recall	99,17	99,37	99,18
	F1-Score	99,17	99,37	99,18
	Fallback	-	3,78	-
Phoneme (ph)	CPU Time	0,0156	0,2719	0,7156
	Accuracy	87,2	87,2	86,42
	Avg Precision	87,06	87,03	86,26
	Avg Recall	87,20	87,20	86,42
	F1-Score	87,06	87,00	86,22
	Fallback	-	2,06	-
Pima Diabetes (pm)	CPU Time	0,0063	0,0437	0,1
	Accuracy	74,02	72,84	74,93
	Avg Precision	73,40	71,96	74,27
	Avg Recall	74,02	72,84	74,93
	F1-Score	73,31	72,04	74,32
	Fallback	-	4,99	-
Ringnorm (ring)	CPU Time	0,1875	5,8719	2,2437
	Accuracy	66,56	79,16	70,86
	Avg Precision	79,90	84,20	73,24
	Avg Recall	66,56	79,16	70,86

Dataset	Μετρική	KNN	Mutual	Symmetric
	F1-Score	62,19	78,32	70,03
	Fallback	-	41,34	-
Segment (sg)	CPU Time	0,0531	0,6719	0,9812
	Accuracy	93,53	95,14	94,88
	Avg Precision	93,62	95,18	95,00
	Avg Recall	93,53	95,14	94,88
	F1-Score	93,42	95,09	94,86
	Fallback	-	3,21	-
Shuttle (sh)	CPU Time	1,7312	9,6031	54,5906
	Accuracy	99,84	99,9	99,86
	Avg Precision	99,85	99,90	99,84
	Avg Recall	99,84	99,90	99,86
	F1-Score	99,84	99,90	99,84
	Fallback	-	1,70	-
Twonorm (tn)	CPU Time	0,1656	3,5219	2,2437
	Accuracy	97,09	95,25	97,11
	Avg Precision	97,10	95,29	97,11
	Avg Recall	97,09	95,25	97,11
	F1-Score	97,09	95,25	97,11
	Fallback	-	17,66	-
Texture (txr)	CPU Time	0,2125	1,5281	2,9875
	Accuracy	98,07	98,65	98,25
	Avg Precision	98,10	98,67	98,27
	Avg Recall	98,07	98,65	98,25
	F1-Score	98,07	98,66	98,25
	Fallback	-	1,93	-
Waveform (wf)	CPU Time	0,0938	2,0594	1,6594
	Accuracy	82,6	78,06	83,94
	Avg Precision	82,69	78,20	83,99
	Avg Recall	82,60	78,06	83,94
	F1-Score	82,50	78,08	83,91

Dataset	Μετρική	KNN	Mutual	Symmetric
	Fallback	-	13,97	-
	CPU Time	0,075	0,6531	0,8875
	Accuracy	47,8	45,68	48,74
Wine Quality	Avg Precision	46,72	45,07	47,22
(wine)	Avg Recall	47,80	45,68	48,74
	F1-Score	46,45	45,04	46,87
	Fallback	-	15,63	-
	CPU Time	0,0156	0,0969	0,2031
	Accuracy	57,04	51,01	58,12
Yeast (ys)	Avg Precision	56,98	50,98	57,59
	Avg Recall	57,04	51,01	58,12
	F1-Score	56,57	50,50	57,50
	Fallback	-	4,60	-

Για $k=7$, ο Symmetric k-NN παρουσιάζει ιδιαίτερα καλή απόδοση σε αρκετά σύνολα, όπως Balance (90,14%), Letter (95,13%) και Monk (96,95%). Ο Mutual k-NN παρουσιάζει σημαντική υπεροχή στο Ring, με ακρίβεια 79,16%, έναντι 66,56% του k-NN και 70,86% του Symmetric k-NN.

Συνολικά, ο k-NN διατηρεί το πλεονέκτημα στον χρόνο εκτέλεσης, ενώ ο Symmetric k-NN εμφανίζει σε αρκετές περιπτώσεις καλύτερη ποιότητα ταξινόμησης.

Πίνακας 5.9: Σύγκριση αλγορίθμων μετά από χρήση του CNN για $k = 1$:

Dataset	Μετρική	KNN	Mutual	Symmetric
	CPU Time	0,0	0,175	0,0688
	Accuracy	70,6	70,6	85,46
Balance (bl)	Avg Precision	75,73	75,73	78,87
	Avg Recall	70,60	70,60	85,46
	F1-Score	73,03	73,03	82,01
	Fallback	-	100,00	-
	CPU Time	0,003125	1,25625	0,5844
	Accuracy	86,12	86,12	89,54
Banana (bn)	Avg Precision	86,14	86,14	89,55

Dataset	Μετρική	KNN	Mutual	Symmetric
Ecoli (ec1)	Avg Recall	86,12	86,12	89,54
	F1-Score	86,12	86,12	89,52
	Fallback	-	99,96	-
	CPU Time	0,0	0,071875	0,0375
	Accuracy	73,64	73,64	78,79
	Avg Precision	75,79	75,79	76,49
Eye State (eeg)	Avg Recall	73,64	73,64	78,79
	F1-Score	73,97	73,97	76,57
	Fallback	-	100,00	-
	CPU Time	0,121875	3,66875	2,3
	Accuracy	46,35	46,35	43,22
	Avg Precision	61,19	61,19	60,16
KDD Cup (kdd)	Avg Recall	46,35	46,35	43,22
	F1-Score	48,97	48,97	44,95
	Fallback	-	100,00	-
	CPU Time	0,3375	384,4875	18,6
	Accuracy	99,4	99,4	99,75
	Avg Precision	99,47	99,47	99,72
Letter Recognition (lir)	Avg Recall	99,40	99,40	99,75
	F1-Score	99,42	99,42	99,73
	Fallback	-	100,00	-
	CPU Time	0,09375	25,8625	3,5094
	Accuracy	92,16	92,16	94,05
	Avg Precision	92,32	92,32	94,10
Landsat Satellite (ls)	Avg Recall	92,16	92,16	94,05
	F1-Score	92,19	92,19	94,04
	Fallback	-	100,00	-
	CPU Time	0,0	18,784375	1,4812
	Accuracy	88,04	88,04	89,81
	Avg Precision	88,69	88,69	89,98
Satellite (ls)	Avg Recall	88,04	88,04	89,81

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
	F1-Score	88,27	88,27	89,85
	Fallback	-	100,00	-
Magic Gamma Telescope (mgt)	CPU Time	0,10625	4,621875	2,6844
	Accuracy	75,92	75,92	83,32
	Avg Precision	75,92	75,92	83,61
	Avg Recall	75,92	75,92	83,32
	F1-Score	75,92	75,92	82,62
	Fallback	-	100,00	-
MONK's Problems (mnk)	CPU Time	0,0	0,1	0,0469
	Accuracy	81,98	81,98	66,5
	Avg Precision	82,22	82,22	66,70
	Avg Recall	81,98	81,98	66,50
	F1-Score	81,92	81,92	66,42
	Fallback	-	100,00	-
Pen-Based Digits (pd)	CPU Time	0,0	18,909375	1,8656
	Accuracy	98,31	98,31	99,34
	Avg Precision	98,32	98,32	99,34
	Avg Recall	98,31	98,31	99,34
	F1-Score	98,31	98,31	99,34
	Fallback	-	100,00	-
Phoneme (ph)	CPU Time	0,003125	1,26875	0,6188
	Accuracy	87,77	87,77	87,9
	Avg Precision	87,86	87,86	87,83
	Avg Recall	87,77	87,77	87,90
	F1-Score	87,80	87,80	87,79
	Fallback	-	100,00	-
Pima Diabetes (pm)	CPU Time	0,0	0,184375	0,0844
	Accuracy	64,56	64,56	74,02
	Avg Precision	65,15	65,15	73,54
	Avg Recall	64,56	64,56	74,02
	F1-Score	64,76	64,76	73,30

Dataset	Μετρική	KNN	Mutual	Symmetric
	Fallback	-	100,00	-
	CPU Time	0,05625	9,6	1,6187
	Accuracy	82,61	82,61	62,16
Ringnorm (ring)	Avg Precision	84,88	84,88	78,39
	Avg Recall	82,61	82,61	62,16
	F1-Score	82,30	82,30	55,61
	Fallback	-	100,00	-
	CPU Time	0,0125	3,60625	0,8562
	Accuracy	95,27	95,27	81,03
Segment (sg)	Avg Precision	95,31	95,31	73,30
	Avg Recall	95,27	95,27	81,03
	F1-Score	95,26	95,26	76,21
	Fallback	-	98,52	-
	CPU Time	0,034375	13,4	6,3594
	Accuracy	99,77	99,77	99,86
Shuttle (sh)	Avg Precision	99,87	99,87	99,87
	Avg Recall	99,77	99,77	99,86
	F1-Score	99,80	99,80	99,86
	Fallback	-	100,00	-
	CPU Time	0,0375	23,515625	1,5844
	Accuracy	89,87	89,87	95,46
Twonorm (tn)	Avg Precision	89,89	89,89	95,47
	Avg Recall	89,87	89,87	95,46
	F1-Score	89,87	89,87	95,46
	Fallback	-	100,00	-
	CPU Time	0,0	9,934375	1,2406
	Accuracy	97,42	97,42	98,43
Texture (txr)	Avg Precision	97,46	97,46	98,45
	Avg Recall	97,42	97,42	98,43
	F1-Score	97,42	97,42	98,43
	Fallback	-	100,00	-

Dataset	Μετρική	KNN	Mutual	Symmetric
Waveform (wf)	CPU Time	0,05625	6,615625	1,3156
	Accuracy	71,47	71,47	79,06
	Avg Precision	71,62	71,62	79,23
	Avg Recall	71,47	71,47	79,06
	F1-Score	71,48	71,48	78,87
	Fallback	-	100,00	-
Wine Quality (wine)	CPU Time	0,065625	1,23125	0,6937
	Accuracy	43,27	43,27	49,05
	Avg Precision	43,90	43,90	37,80
	Avg Recall	43,27	43,27	49,05
	F1-Score	43,41	43,41	42,11
	Fallback	-	99,80	-
Yeast (ys)	CPU Time	0,003125	0,35625	0,175
	Accuracy	48,17	48,17	55,68
	Avg Precision	50,10	50,10	55,82
	Avg Recall	48,17	48,17	55,68
	F1-Score	48,65	48,65	55,03
	Fallback	-	100,00	-

Μετά την εφαρμογή του CNN για $k=1$, ο Symmetric k -NN παρουσιάζει την καλύτερη ακρίβεια στα περισσότερα σύνολα δεδομένων, με χαρακτηριστικά παραδείγματα το Balance (85,46%), το KDD Cup (99,75%) και το Pendigits (99,34%). Εξαιρέση αποτελούν σύνολα όπως το Monk και το Ring, όπου ο k -NN και ο Mutual k -NN παρουσιάζουν καλύτερη απόδοση.

Ο k -NN έχει σταθερά το μικρότερο CPU Time, ενώ ο Mutual k -NN παρουσιάζει πολύ υψηλό Fallback, στις περισσότερες περιπτώσεις κοντά στο 100%.

Πίνακας 5.10: Σύγκριση αλγορίθμων μετά από χρήση του CNN για $k = 3$:

Dataset	Μετρική	KNN	Mutual	Symmetric
Balance (bl)	CPU Time	0,003125	0,071875	0,0656
	Accuracy	75,12	76,09	86,11
	Avg Precision	79,72	79,33	79,77
	Avg Recall	75,12	76,09	86,11

Dataset	Μετρική	KNN	Mutual	Symmetric
	F1-Score	77,24	77,34	82,65
	Fallback	-	26,34	-
Banana (bn)	CPU Time	0,0	0,4375	0,5969
	Accuracy	87,5	87,31	89,52
	Avg Precision	87,49	87,45	89,55
	Avg Recall	87,50	87,31	89,52
	F1-Score	87,48	87,23	89,49
	Fallback	-	25,78	-
Ecoli (ec1)	CPU Time	0,003125	0,025	0,0437
	Accuracy	70,61	74,24	78,18
	Avg Precision	70,90	74,27	75,75
	Avg Recall	70,61	74,24	78,18
	F1-Score	69,22	73,24	75,95
	Fallback	-	19,39	-
Eye State (eeg)	CPU Time	0,146875	2,38125	2,3813
	Accuracy	44,57	46,37	43,19
	Avg Precision	60,42	61,13	60,06
	Avg Recall	44,57	46,37	43,19
	F1-Score	46,76	48,41	44,85
	Fallback	-	51,85	-
KDD Cup (kdd)	CPU Time	0,3625	91,496875	18,6812
	Accuracy	98,99	99,1	99,75
	Avg Precision	99,04	99,17	99,72
	Avg Recall	98,99	99,10	99,75
	F1-Score	98,98	99,12	99,73
	Fallback	-	20,47	-
Letter Recognition (lir)	CPU Time	0,09375	5,94375	3,425
	Accuracy	86,71	89,7	93,83
	Avg Precision	87,70	90,03	93,90
	Avg Recall	86,71	89,70	93,83
	F1-Score	86,88	89,73	93,83

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
	Fallback	-	8,15	-
	CPU Time	0,0	5,25625	1,4906
	Accuracy	88,72	88,32	90,03
Landsat	Avg Precision	88,83	88,57	90,17
Satellite (ls)	Avg Recall	88,72	88,32	90,03
	F1-Score	88,73	88,38	90,04
	Fallback	-	23,27	-
	CPU Time	0,140625	1,821875	2,7594
	Accuracy	79,78	78,08	83,66
Magic Gamma	Avg Precision	79,44	77,71	83,96
Telescope (mgt)	Avg Recall	79,78	78,08	83,66
	F1-Score	79,42	77,77	82,99
	Fallback	-	23,91	-
	CPU Time	0,003125	0,05	0,05
	Accuracy	82,67	83,14	83,14
MONK's	Avg Precision	83,30	83,44	84,95
Problems (mnk)	Avg Recall	82,67	83,14	83,14
	F1-Score	82,52	83,12	83,02
	Fallback	-	37,27	-
	CPU Time	0,05625	2,91875	1,8594
	Accuracy	97,25	96,66	99,36
Pen-Based	Avg Precision	97,30	96,75	99,37
Digits (pd)	Avg Recall	97,25	96,66	99,36
	F1-Score	97,25	96,67	99,36
	Fallback	-	10,56	-
	CPU Time	0,00625	0,384375	0,6219
	Accuracy	84,92	85,92	87,72
Phoneme (ph)	Avg Precision	85,09	85,70	87,63
	Avg Recall	84,92	85,92	87,72
	F1-Score	84,97	85,64	87,60
	Fallback	-	19,60	-

Dataset	Μετρική	KNN	Mutual	Symmetric
Pima Diabetes (pm)	CPU Time	0,003125	0,06875	0,0875
	Accuracy	70,21	68,5	74,02
	Avg Precision	70,28	67,97	73,34
	Avg Recall	70,21	68,50	74,02
	F1-Score	70,19	68,03	73,20
	Fallback	-	24,81	-
Ringnorm (ring)	CPU Time	0,05625	5,971875	1,6469
	Accuracy	84,54	82,79	63,84
	Avg Precision	87,49	83,82	78,94
	Avg Recall	84,54	82,79	63,84
	F1-Score	84,21	82,64	58,20
	Fallback	-	47,96	-
Segment (sg)	CPU Time	0,00625	0,578125	0,8313
	Accuracy	92,75	92,53	81,38
	Avg Precision	93,07	92,91	73,62
	Avg Recall	92,75	92,53	81,38
	F1-Score	92,80	92,56	76,54
	Fallback	-	10,59	-
Shuttle (sh)	CPU Time	0,04375	4,5625	6,3438
	Accuracy	97,95	95,84	99,87
	Avg Precision	99,36	99,19	99,87
	Avg Recall	97,95	95,84	99,87
	F1-Score	98,48	96,80	99,86
	Fallback	-	24,58	-
Twonorm (tn)	CPU Time	0,05625	11,709375	1,5656
	Accuracy	94,23	90,98	95,44
	Avg Precision	94,23	91,00	95,47
	Avg Recall	94,23	90,98	95,44
	F1-Score	94,22	90,98	95,44
	Fallback	-	47,98	-
	CPU Time	0,00625	1,15625	1,2375

Dataset	Μετρική	KNN	Mutual	Symmetric
Texture (txr)	Accuracy	92,72	95,76	98,29
	Avg Precision	93,51	95,90	98,31
	Avg Recall	92,72	95,76	98,29
	F1-Score	92,67	95,77	98,29
	Fallback	-	7,77	-
Waveform (wf)	CPU Time	0,06875	3,503125	1,3094
	Accuracy	76,0	72,23	78,96
	Avg Precision	76,09	72,55	78,99
	Avg Recall	76,00	72,23	78,96
	F1-Score	75,98	72,27	78,89
	Fallback	-	42,50	-
Wine Quality (wine)	CPU Time	0,034375	0,596875	0,7219
	Accuracy	42,45	42,61	48,83
	Avg Precision	42,63	42,88	37,56
	Avg Recall	42,45	42,61	48,83
	F1-Score	41,98	42,53	41,99
	Fallback	-	34,31	-
Yeast (ys)	CPU Time	0,009375	0,134375	0,1844
	Accuracy	48,98	48,71	55,41
	Avg Precision	50,26	50,11	55,32
	Avg Recall	48,98	48,71	55,41
	F1-Score	48,35	48,66	54,77
	Fallback	-	23,07	-

Για $k=3$, ο Symmetric k -NN παρουσιάζει την υψηλότερη ακρίβεια σε αρκετά σύνολα, όπως Balance (86,11%), Letter (93,83%), Kddcup (99,75%) και Texture (98,29%). Αντίθετα, ο Mutual k -NN υπερέχει σε EEG Eye State και Segment, ενώ στο Ring ο k -NN έχει καλύτερη επίδοση από τον Symmetric.

Ο k -NN παραμένει ο ταχύτερος αλγόριθμος, ενώ ο Mutual k -NN εμφανίζει αυξημένο Fallback, φτάνοντας έως 51,85% στο EEG Eye State και περίπου 48% στα Ring και Twonorm.

Πίνακας 5.11: Σύγκριση αλγορίθμων μετά από χρήση του CNN για $k = 5$:

Dataset	Μετρική	KNN	Mutual	Symmetric
Balance (bl)	CPU Time	0,009375	0,05	0,0688
	Accuracy	78,35	76,57	87,56
	Avg Precision	80,82	78,76	80,97
	Avg Recall	78,35	76,57	87,56
	F1-Score	78,86	77,37	84,00
	Fallback	-	13,26	-
Banana (bn)	CPU Time	0,003125	0,303125	0,5938
	Accuracy	88,27	87,61	89,69
	Avg Precision	88,26	87,79	89,75
	Avg Recall	88,27	87,61	89,69
	F1-Score	88,26	87,53	89,65
	Fallback	-	13,56	-
Ecoli (ecl)	CPU Time	0,0	0,015625	0,0375
	Accuracy	73,64	74,24	79,39
	Avg Precision	73,64	73,28	77,29
	Avg Recall	73,64	74,24	79,39
	F1-Score	71,84	72,86	77,38
	Fallback	-	5,15	-
Eye State (eeg)	CPU Time	0,165625	1,871875	2,4219
	Accuracy	43,55	45,83	43,17
	Avg Precision	59,75	60,80	59,77
	Avg Recall	43,55	45,83	43,17
	F1-Score	45,52	47,45	44,73
	Fallback	-	35,01	-
KDD Cup (kdd)	CPU Time	0,340625	68,65625	18,9094
	Accuracy	98,16	98,96	99,75
	Avg Precision	98,47	99,04	99,72
	Avg Recall	98,16	98,96	99,75
	F1-Score	98,24	98,97	99,73
	Fallback	-	14,40	-
	CPU Time	0,096875	4,546875	3,4906

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Letter Recognition (lir)	Accuracy	84,75	89,24	93,77
	Avg Precision	85,81	89,62	93,85
	Avg Recall	84,75	89,24	93,77
	F1-Score	84,87	89,26	93,78
	Fallback	-	2,54	-
Landsat Satellite (ls)	CPU Time	0,028125	2,8375	1,5
	Accuracy	88,97	87,84	90,33
	Avg Precision	88,97	88,13	90,38
	Avg Recall	88,97	87,84	90,33
	F1-Score	88,93	87,86	90,28
	Fallback	-	10,97	-
Magic Gamma Telescope (mgt)	CPU Time	0,1625	1,25625	2,7938
	Accuracy	81,11	79,7	83,91
	Avg Precision	80,87	79,35	84,21
	Avg Recall	81,11	79,70	83,91
	F1-Score	80,61	79,23	83,25
	Fallback	-	8,58	-
MONK's Problems (mnk)	CPU Time	0,0	0,025	0,05
	Accuracy	79,15	78,21	95,78
	Avg Precision	79,20	79,51	95,98
	Avg Recall	79,15	78,21	95,78
	F1-Score	79,12	78,06	95,77
	Fallback	-	14,29	-
Pen-Based Digits (pd)	CPU Time	0,0	1,50625	1,8656
	Accuracy	95,99	97,05	99,38
	Avg Precision	96,07	97,11	99,38
	Avg Recall	95,99	97,05	99,38
	F1-Score	95,99	97,06	99,38
	Fallback	-	4,16	-
	CPU Time	0,0125	0,2625	0,6219
	Accuracy	84,36	84,98	87,01

Dataset	Μετρική	KNN	Mutual	Symmetric
Phoneme (ph)	Avg Precision	84,49	84,69	86,88
	Avg Recall	84,36	84,98	87,01
	F1-Score	84,41	84,63	86,85
	Fallback	-	9,39	-
Pima Diabetes (pm)	CPU Time	0,009375	0,040625	0,0906
	Accuracy	68,38	69,3	72,05
	Avg Precision	67,90	68,29	71,24
	Avg Recall	68,38	69,30	72,05
	F1-Score	68,07	68,50	70,98
	Fallback	-	9,58	-
	CPU Time	0,01875	4,4	1,6406
Ringnorm (ring)	Accuracy	84,53	84,57	65,57
	Avg Precision	87,98	85,02	79,55
	Avg Recall	84,53	84,57	65,57
	F1-Score	84,14	84,51	60,77
	Fallback	-	34,00	-
Segment (sg)	CPU Time	0,00625	0,346875	0,8469
	Accuracy	82,51	91,02	81,03
	Avg Precision	83,87	91,60	73,26
	Avg Recall	82,51	91,02	81,03
	F1-Score	81,06	91,00	76,22
	Fallback	-	3,95	-
Shuttle (sh)	CPU Time	0,0375	3,175	6,3969
	Accuracy	97,0	95,36	99,87
	Avg Precision	98,38	97,92	99,87
	Avg Recall	97,00	95,36	99,87
	F1-Score	97,52	96,19	99,87
	Fallback	-	13,18	-
Twonorm (tn)	CPU Time	0,01875	7,615625	1,6
	Accuracy	95,21	91,76	95,42
	Avg Precision	95,22	91,82	95,44

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
	Avg Recall	95,21	91,76	95,42
	F1-Score	95,21	91,76	95,41
	Fallback	-	29,10	-
Texture (txr)	CPU Time	0,015625	0,61875	1,2469
	Accuracy	83,11	95,36	98,27
	Avg Precision	87,21	95,57	98,29
	Avg Recall	83,11	95,36	98,27
	F1-Score	81,20	95,37	98,27
	Fallback	-	2,68	-
Waveform (wf)	CPU Time	0,0375	2,49375	1,3125
	Accuracy	78,0	74,85	79,68
	Avg Precision	78,04	75,30	79,70
	Avg Recall	78,00	74,85	79,68
	F1-Score	77,96	74,90	79,65
	Fallback	-	23,28	-
Wine Quality (wine)	CPU Time	0,04375	0,415625	0,7281
	Accuracy	47,68	43,49	48,21
	Avg Precision	46,69	43,65	37,15
	Avg Recall	47,68	43,49	48,21
	F1-Score	46,37	43,12	41,55
	Fallback	-	16,10	-
Yeast (ys)	CPU Time	0,00625	0,078125	0,1875
	Accuracy	50,67	48,17	54,67
	Avg Precision	50,55	49,08	54,15
	Avg Recall	50,67	48,17	54,67
	F1-Score	49,75	47,64	54,01
	Fallback	-	9,07	-

Για $k=5$, ο Symmetric k-NN διατηρεί την καλύτερη απόδοση στα περισσότερα σύνολα, με πολύ υψηλές τιμές στα Monk (95,78%), Pendigits (99,38%), Kddcup (99,75%) και Texture (98,27%). Στο Ring, αντίθετα, ο Symmetric παρουσιάζει αισθητά χαμηλότερη ακρίβεια (65,57%), ενώ k-NN και Mutual ξεπερνούν το 84%.

Ο k-NN εξακολουθεί να έχει το μικρότερο CPU Time, ενώ το Fallback του Mutual αυξάνεται σε αρκετά σύνολα, φτάνοντας το 34% στο Ring και 29,10% στο Twonorm.

Πίνακας 5.12: Σύγκριση αλγορίθμων μετά από χρήση του CNN για $k = 7$:

Dataset	Μετρική	KNN	Mutual	Symmetric
Balance (bl)	CPU Time	0,0	0,034375	0,075
	Accuracy	81,75	82,39	88,21
	Avg Precision	81,43	81,12	81,54
	Avg Recall	81,75	82,39	88,21
	F1-Score	81,23	81,25	84,61
	Fallback	-	8,73	-
Banana (bn)	CPU Time	0,00625	0,234375	0,5969
	Accuracy	88,76	88,57	89,69
	Avg Precision	88,75	88,66	89,72
	Avg Recall	88,76	88,57	89,69
	F1-Score	88,75	88,52	89,66
	Fallback	-	8,59	-
Ecoli (ecl)	CPU Time	0,0	0,015625	0,0406
	Accuracy	75,15	73,94	79,09
	Avg Precision	72,86	73,49	76,66
	Avg Recall	75,15	73,94	79,09
	F1-Score	73,00	72,70	76,92
	Fallback	-	1,52	-
Eye State (eeg)	CPU Time	0,18125	1,621875	2,45
	Accuracy	42,54	45,64	43,19
	Avg Precision	58,85	60,73	59,61
	Avg Recall	42,54	45,64	43,19
	F1-Score	44,31	47,14	44,63
	Fallback	-	26,62	-
KDD Cup (kdd)	CPU Time	0,434375	59,915625	18,8938
	Accuracy	97,5	98,84	99,74
	Avg Precision	97,62	98,89	99,72
	Avg Recall	97,50	98,84	99,74

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Letter Recognition (lir)	F1-Score	97,46	98,83	99,73
	Fallback	-	11,81	-
	CPU Time	0,1125	3,6375	3,4906
	Accuracy	82,1	88,51	93,82
	Avg Precision	83,50	88,95	93,92
	Avg Recall	82,10	88,51	93,82
Landsat Satellite (ls)	F1-Score	82,28	88,53	93,82
	Fallback	-	1,17	-
	CPU Time	0,0375	2,03125	1,5469
	Accuracy	89,35	88,88	90,54
	Avg Precision	89,32	88,88	90,50
	Avg Recall	89,35	88,88	90,54
Magic Gamma Telescope (mgt)	F1-Score	89,27	88,81	90,46
	Fallback	-	6,50	-
	CPU Time	0,178125	1,10625	2,85
	Accuracy	82,24	80,81	84,32
	Avg Precision	82,21	80,55	84,65
	Avg Recall	82,24	80,81	84,32
MONK's Problems (mnk)	F1-Score	81,64	80,29	83,69
	Fallback	-	4,05	-
	CPU Time	0,00625	0,01875	0,0469
	Accuracy	78,66	76,57	94,37
	Avg Precision	78,81	77,71	94,92
	Avg Recall	78,66	76,57	94,37
Pen-Based Digits (pd)	F1-Score	78,63	76,46	94,35
	Fallback	-	5,62	-
	CPU Time	0,0375	1,01875	1,875
	Accuracy	94,85	96,88	99,4
	Avg Precision	95,01	96,95	99,40
Pen-Based Digits (pd)	Avg Recall	94,85	96,88	99,40
	F1-Score	94,83	96,89	99,40

Dataset	Μετρική	KNN	Mutual	Symmetric
	Fallback	-	1,55	-
Phoneme (ph)	CPU Time	0,009375	0,221875	0,6312
	Accuracy	84,4	84,46	86,42
	Avg Precision	84,43	84,17	86,26
	Avg Recall	84,40	84,46	86,42
	F1-Score	84,40	84,19	86,23
	Fallback	-	5,30	-
Pima Diabetes (pm)	CPU Time	0,003125	0,0375	0,0906
	Accuracy	70,48	66,81	72,84
	Avg Precision	69,98	65,78	72,13
	Avg Recall	70,48	66,81	72,84
	F1-Score	70,11	65,99	71,75
	Fallback	-	5,51	-
Ringnorm (ring)	CPU Time	0,071875	3,634375	1,6344
	Accuracy	84,5	86,63	66,91
	Avg Precision	88,07	86,82	79,95
	Avg Recall	84,50	86,63	66,91
	F1-Score	84,10	86,60	62,70
	Fallback	-	26,19	-
Segment (sg)	CPU Time	0,0125	0,25	0,8313
	Accuracy	66,1	90,41	81,03
	Avg Precision	65,68	90,89	73,29
	Avg Recall	66,10	90,41	81,03
	F1-Score	60,70	90,40	76,20
	Fallback	-	1,95	-
Shuttle (sh)	CPU Time	0,046875	2,596875	6,4062
	Accuracy	96,86	95,48	99,87
	Avg Precision	98,16	97,40	99,86
	Avg Recall	96,86	95,48	99,87
	F1-Score	97,35	96,24	99,86
	Fallback	-	8,12	-

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Twonorm (tn)	CPU Time	0,0375	5,75625	1,5781
	Accuracy	95,96	92,25	95,59
	Avg Precision	95,96	92,30	95,62
	Avg Recall	95,96	92,25	95,59
	F1-Score	95,96	92,25	95,59
	Fallback	-	20,11	-
Texture (txr)	CPU Time	0,0125	0,571875	1,2594
	Accuracy	77,9	94,99	98,23
	Avg Precision	82,92	95,29	98,26
	Avg Recall	77,90	94,99	98,23
	F1-Score	72,76	95,02	98,24
	Fallback	-	1,16	-
Waveform (wf)	CPU Time	0,009375	1,778125	1,3531
	Accuracy	79,32	75,96	80,64
	Avg Precision	79,34	76,37	80,63
	Avg Recall	79,32	75,96	80,64
	F1-Score	79,24	76,01	80,60
	Fallback	-	14,01	-
Wine Quality (wine)	CPU Time	0,046875	0,340625	0,7406
	Accuracy	47,86	44,57	48,11
	Avg Precision	46,10	44,64	36,98
	Avg Recall	47,86	44,57	48,11
	F1-Score	46,09	44,06	41,39
	Fallback	-	8,46	-
Yeast (ys)	CPU Time	0,00625	0,065625	0,1875
	Accuracy	52,43	49,8	56,36
	Avg Precision	52,11	49,83	56,07
	Avg Recall	52,43	49,80	56,36
	F1-Score	51,55	48,94	55,71
	Fallback	-	4,33	-

Dataset	Μετρική	KNN	Mutual	Symmetric
---------	---------	-----	--------	-----------

Για $k=7$, ο Symmetric k -NN παρουσιάζει και πάλι πολύ καλή απόδοση, με κορυφαίες τιμές στα Shuttle (99,87%), Kddcup (99,74%), Pendigits (99,40%) και Monk (94,37%).

Εξαίρεση αποτελεί το Ring, όπου ο Mutual k -NN επιτυγχάνει 86,63%, έναντι 84,50% του k -NN και 66,91% του Symmetric. Συνολικά, ο k -NN παραμένει ο ταχύτερος, ενώ ο Mutual εμφανίζει υψηλότερο Fallback, με τιμές έως 26,62% στο EEG Eye State και 26,19% στο Ring.

Πίνακας 5.13: Σύγκριση αλγορίθμων μετά από χρήση του ENN για $k = 1$:

Dataset	Μετρική	KNN	Mutual	Symmetric
Balance (bl)	CPU Time	0,003125	0,16875	0,0813
	Accuracy	85,46	85,46	89,82
	Avg Precision	78,87	78,87	82,94
	Avg Recall	85,46	85,46	89,82
	F1-Score	82,01	82,01	86,17
	Fallback	-	100,00	-
Banana (bn)	CPU Time	0,00625	1,24375	0,6312
	Accuracy	89,54	89,54	89,37
	Avg Precision	89,55	89,55	89,39
	Avg Recall	89,54	89,54	89,37
	F1-Score	89,52	89,52	89,34
	Fallback	-	99,98	-
Ecoli (ecl)	CPU Time	0,003125	0,078125	0,0375
	Accuracy	78,79	78,79	78,48
	Avg Precision	76,49	76,49	75,80
	Avg Recall	78,79	78,79	78,48
	F1-Score	76,57	76,57	76,15
	Fallback	-	100,00	-
Eye State (eeg)	CPU Time	0,325	4,04375	5,3281
	Accuracy	43,22	43,22	45,39
	Avg Precision	60,16	60,16	61,05
	Avg Recall	43,22	43,22	45,39
	F1-Score	44,95	44,95	47,11

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
	Fallback	-	100,00	-
	CPU Time	22,81875	1042,134375	1170,3344
	Accuracy	99,75	99,75	99,67
KDD Cup (kdd)	Avg Precision	99,72	99,72	99,64
	Avg Recall	99,75	99,75	99,67
	F1-Score	99,73	99,73	99,65
	Fallback	-	100,00	-
	CPU Time	0,46875	28,628125	11,975
	Accuracy	94,05	94,05	94,61
Letter	Avg Precision	94,10	94,10	94,69
Recognition (lir)	Avg Recall	94,05	94,05	94,61
	F1-Score	94,04	94,04	94,62
	Fallback	-	100,00	-
	CPU Time	0,1125	8,83125	3,3031
	Accuracy	89,81	89,81	89,77
Landsat	Avg Precision	89,98	89,98	89,94
Satellite (ls)	Avg Recall	89,81	89,81	89,77
	F1-Score	89,85	89,85	89,79
	Fallback	-	100,00	-
	CPU Time	0,15625	4,796875	3,5688
	Accuracy	83,32	83,32	83,05
Magic Gamma	Avg Precision	83,61	83,61	83,28
Telescope (mgt)	Avg Recall	83,32	83,32	83,05
	F1-Score	82,62	82,62	82,36
	Fallback	-	100,00	-
	CPU Time	0,0	0,103125	0,0531
	Accuracy	66,5	66,5	98,82
MONK's	Avg Precision	66,70	66,70	98,88
Problems (mnk)	Avg Recall	66,50	66,50	98,82
	F1-Score	66,42	66,42	98,82
	Fallback	-	100,00	-

Dataset	Μετρική	KNN	Mutual	Symmetric
Pen-Based Digits (pd)	CPU Time	0,3125	15,540625	5,05
	Accuracy	99,34	99,34	99,15
	Avg Precision	99,34	99,34	99,16
	Avg Recall	99,34	99,34	99,15
	F1-Score	99,34	99,34	99,15
	Fallback	-	100,00	-
Phoneme (ph)	CPU Time	0,009375	1,3125	0,6906
	Accuracy	87,9	87,9	87,79
	Avg Precision	87,83	87,83	87,68
	Avg Recall	87,90	87,90	87,79
	F1-Score	87,79	87,79	87,62
	Fallback	-	100,00	-
Pima Diabetes (pm)	CPU Time	0,00625	0,178125	0,0875
	Accuracy	74,02	74,02	74,68
	Avg Precision	73,54	73,54	73,98
	Avg Recall	74,02	74,02	74,68
	F1-Score	73,30	73,30	73,75
	Fallback	-	100,00	-
Ringnorm (ring)	CPU Time	0,109375	9,76875	2,0312
	Accuracy	62,16	62,16	67,42
	Avg Precision	78,39	78,39	69,41
	Avg Recall	62,16	62,16	67,42
	F1-Score	55,61	55,61	66,48
	Fallback	-	100,00	-
Segment (sg)	CPU Time	0,0	2,725	0,9281
	Accuracy	81,03	81,03	80,77
	Avg Precision	73,30	73,30	71,57
	Avg Recall	81,03	81,03	80,77
	F1-Score	76,21	76,21	75,37
	Fallback	-	87,76	-
	CPU Time	0,621875	15,996875	45,6437

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Shuttle (sh)	Accuracy	99,86	99,86	99,89
	Avg Precision	99,87	99,87	99,89
	Avg Recall	99,86	99,86	99,89
	F1-Score	99,86	99,86	99,89
	Fallback	-	100,00	-
Twonorm (tn)	CPU Time	0,1875	10,0125	2,3531
	Accuracy	95,46	95,46	95,47
	Avg Precision	95,47	95,47	95,49
	Avg Recall	95,46	95,46	95,47
	F1-Score	95,46	95,46	95,47
	Fallback	-	100,00	-
Texture (txr)	CPU Time	0,134375	8,103125	2,9375
	Accuracy	98,43	98,43	98,42
	Avg Precision	98,45	98,45	98,43
	Avg Recall	98,43	98,43	98,42
	F1-Score	98,43	98,43	98,42
	Fallback	-	100,00	-
Waveform (wf)	CPU Time	0,040625	6,55625	1,5625
	Accuracy	79,06	79,06	80,16
	Avg Precision	79,23	79,23	80,17
	Avg Recall	79,06	79,06	80,16
	F1-Score	78,87	78,87	80,11
	Fallback	-	100,00	-
Wine Quality (wine)	CPU Time	0,0625	1,2875	0,6687
	Accuracy	49,05	49,05	48,52
	Avg Precision	37,80	37,80	37,47
	Avg Recall	49,05	49,05	48,52
	F1-Score	42,11	42,11	41,82
	Fallback	-	100,00	-
	CPU Time	0,009375	0,35	0,1781
	Accuracy	55,68	55,68	55,0

Dataset	Μετρική	KNN	Mutual	Symmetric
Yeast (ys)	Avg Precision	55,82	55,82	55,15
	Avg Recall	55,68	55,68	55,00
	F1-Score	55,03	55,03	54,51
	Fallback	-	100,00	-

Για $k=1$: Μετά την εφαρμογή του ENN, ο Symmetric k -NN παρουσιάζει καλύτερη ακρίβεια σε αρκετά σύνολα δεδομένων, με χαρακτηριστικό παράδειγμα το Monk, όπου φτάνει το 98,82%, έναντι 66,50% των k -NN και Mutual k -NN. Αντίστοιχα, παρουσιάζει υψηλή απόδοση στα Balance και Ring. Ο k -NN παραμένει ο ταχύτερος αλγόριθμος, ενώ ο Mutual k -NN εμφανίζει πολύ υψηλό ποσοστό Fallback, το οποίο στις περισσότερες περιπτώσεις είναι 100%.

Πίνακας 5.14: Σύγκριση αλγορίθμων μετά από χρήση του ENN για $k = 3$:

Dataset	Μετρική	KNN	Mutual	Symmetric
Balance (bl)	CPU Time	0,0	0,034375	0,0688
	Accuracy	89,01	86,11	88,85
	Avg Precision	82,00	79,77	82,00
	Avg Recall	89,01	86,11	88,85
	F1-Score	85,35	82,65	85,21
	Fallback	-	5,01	-
Banana (bn)	CPU Time	0,003125	0,284375	0,6469
	Accuracy	89,55	89,52	89,65
	Avg Precision	89,56	89,55	89,68
	Avg Recall	89,55	89,52	89,65
	F1-Score	89,53	89,49	89,62
	Fallback	-	12,22	-
Ecoli (ecl)	CPU Time	0,0	0,034375	0,0406
	Accuracy	77,88	78,18	77,27
	Avg Precision	75,66	75,75	74,91
	Avg Recall	77,88	78,18	77,27
	F1-Score	75,52	75,95	74,71
	Fallback	-	22,73	-
	CPU Time	0,340625	4,078125	5,5875
	Accuracy	42,75	43,19	44,46

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Eye State (eeg)	Avg Precision	59,84	60,06	60,54
	Avg Recall	42,75	43,19	44,46
	F1-Score	44,19	44,85	45,94
	Fallback	-	81,55	-
KDD Cup (kdd)	CPU Time	22,940625	264,128125	1136,8187
	Accuracy	99,72	99,75	99,67
	Avg Precision	99,69	99,72	99,65
	Avg Recall	99,72	99,75	99,67
	F1-Score	99,71	99,73	99,66
	Fallback	-	15,05	-
	CPU Time	0,621875	12,4625	12,15
	Accuracy	93,6	93,83	94,61
Letter	Avg Precision	93,75	93,90	94,73
Recognition (lir)	Avg Recall	93,60	93,83	94,61
	F1-Score	93,62	93,83	94,62
	Fallback	-	24,60	-
Landsat Satellite (ls)	CPU Time	0,165625	4,015625	3,3969
	Accuracy	89,7	90,03	90,0
	Avg Precision	89,77	90,17	89,99
	Avg Recall	89,70	90,03	90,00
	F1-Score	89,69	90,04	89,94
	Fallback	-	33,78	-
Magic Gamma Telescope (mgt)	CPU Time	0,2	2,296875	3,8125
	Accuracy	83,38	83,66	83,59
	Avg Precision	83,92	83,96	83,90
	Avg Recall	83,38	83,66	83,59
	F1-Score	82,56	82,99	82,89
	Fallback	-	25,88	-
MONK's	CPU Time	0,0	0,01875	0,05
	Accuracy	95,77	83,14	98,82
	Avg Precision	95,84	84,95	98,88

Dataset	Μετρική	KNN	Mutual	Symmetric
Problems (mnk)	Avg Recall	95,77	83,14	98,82
	F1-Score	95,77	83,02	98,82
	Fallback	-	2,57	-
Pen-Based Digits (pd)	CPU Time	0,2625	5,503125	5,0906
	Accuracy	99,24	99,36	99,18
	Avg Precision	99,25	99,37	99,19
	Avg Recall	99,24	99,36	99,18
	F1-Score	99,25	99,36	99,18
	Fallback	-	23,18	-
Phoneme (ph)	CPU Time	0,0125	0,4	0,7
	Accuracy	86,75	87,72	86,5
	Avg Precision	86,63	87,63	86,33
	Avg Recall	86,75	87,72	86,50
	F1-Score	86,62	87,60	86,30
	Fallback	-	19,71	-
Pima Diabetes (pm)	CPU Time	0,003125	0,06875	0,0906
	Accuracy	74,55	74,02	74,67
	Avg Precision	74,17	73,34	74,06
	Avg Recall	74,55	74,02	74,67
	F1-Score	73,97	73,20	74,03
	Fallback	-	26,91	-
Ringnorm (ring)	CPU Time	0,09375	6,91875	2,0781
	Accuracy	58,61	63,84	66,0
	Avg Precision	77,27	78,94	72,61
	Avg Recall	58,61	63,84	66,00
	F1-Score	49,74	58,20	63,17
	Fallback	-	66,84	-
Segment (sg)	CPU Time	0,0	1,23125	0,9563
	Accuracy	80,34	81,38	80,77
	Avg Precision	72,78	73,62	71,62
	Avg Recall	80,34	81,38	80,77

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
	F1-Score	75,58	76,54	75,36
	Fallback	-	31,90	-
Shuttle (sh)	CPU Time	1,028125	7,053125	47,0406
	Accuracy	99,85	99,87	99,87
	Avg Precision	99,84	99,87	99,86
	Avg Recall	99,85	99,87	99,87
	F1-Score	99,84	99,86	99,86
	Fallback	-	10,01	-
Twonorm (tn)	CPU Time	0,078125	5,459375	2,4031
	Accuracy	96,63	95,44	96,74
	Avg Precision	96,64	95,47	96,75
	Avg Recall	96,63	95,44	96,74
	F1-Score	96,63	95,44	96,74
	Fallback	-	46,20	-
Texture (txr)	CPU Time	0,134375	2,5	2,9125
	Accuracy	98,11	98,29	98,47
	Avg Precision	98,14	98,31	98,49
	Avg Recall	98,11	98,29	98,47
	F1-Score	98,11	98,29	98,47
	Fallback	-	19,44	-
Waveform (wf)	CPU Time	0,05625	3,35	1,575
	Accuracy	81,74	78,96	82,72
	Avg Precision	82,00	78,99	82,84
	Avg Recall	81,74	78,96	82,72
	F1-Score	81,57	78,89	82,63
	Fallback	-	41,96	-
Wine Quality (wine)	CPU Time	0,0375	0,8	0,7156
	Accuracy	49,13	48,83	49,36
	Avg Precision	38,15	37,56	38,01
	Avg Recall	49,13	48,83	49,36
	F1-Score	42,23	41,99	42,38

Dataset	Μετρική	KNN	Mutual	Symmetric
	Fallback	-	54,46	-
	CPU Time	0,003125	0,146875	0,1781
	Accuracy	57,91	55,41	58,39
Yeast (ys)	Avg Precision	58,14	55,32	58,37
	Avg Recall	57,91	55,41	58,39
	F1-Score	57,20	54,77	57,86
	Fallback	-	30,65	-

Για $k=3$: Ο Symmetric k-NN παρουσιάζει την υψηλότερη ακρίβεια σε αρκετά σύνολα, όπως Monk (98,82%), Letter (94,61%) και Ring (66,00%). Ο Mutual k-NN υπερέρχει σε ορισμένα σύνολα, όπως Phoneme και Segment, αλλά συνοδεύεται από σχετικά υψηλά ποσοστά Fallback. Ο k-NN εξακολουθεί να έχει τον μικρότερο CPU Time.

Πίνακας 5.15: Σύγκριση αλγορίθμων μετά από χρήση του ENN για $k = 5$:

Dataset	Μετρική	KNN	Mutual	Symmetric
	CPU Time	0,0	0,025	0,0844
	Accuracy	88,85	87,56	89,18
Balance (bl)	Avg Precision	81,93	80,97	82,23
	Avg Recall	88,85	87,56	89,18
	F1-Score	85,21	84,00	85,53
	Fallback	-	0,32	-
	CPU Time	0,009375	0,159375	0,6531
	Accuracy	89,82	89,69	89,88
Banana (bn)	Avg Precision	89,83	89,75	89,91
	Avg Recall	89,82	89,69	89,88
	F1-Score	89,80	89,65	89,85
	Fallback	-	1,44	-
	CPU Time	0,003125	0,015625	0,0406
	Accuracy	76,36	79,39	76,36
Ecoli (ecl)	Avg Precision	74,36	77,29	74,05
	Avg Recall	76,36	79,39	76,36
	F1-Score	73,68	77,38	73,77
	Fallback	-	5,15	-
	CPU Time	0,36875	3,6125	5,4688

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Eye State (eeg)	Accuracy	42,85	43,17	43,89
	Avg Precision	60,07	59,77	60,12
	Avg Recall	42,85	43,17	43,89
	F1-Score	44,16	44,73	45,26
	Fallback	-	64,27	-
KDD Cup (kdd)	CPU Time	22,79375	169,6625	1157,9781
	Accuracy	99,71	99,75	99,66
	Avg Precision	99,69	99,72	99,64
	Avg Recall	99,71	99,75	99,66
	F1-Score	99,70	99,73	99,65
	Fallback	-	5,15	-
Letter Recognition (lir)	CPU Time	0,625	7,28125	12,3594
	Accuracy	93,47	93,77	94,31
	Avg Precision	93,60	93,85	94,43
	Avg Recall	93,47	93,77	94,31
	F1-Score	93,48	93,78	94,32
	Fallback	-	7,70	-
Landsat Satellite (ls)	CPU Time	0,16875	2,784375	3,3062
	Accuracy	89,69	90,33	90,36
	Avg Precision	89,77	90,38	90,38
	Avg Recall	89,69	90,33	90,36
	F1-Score	89,68	90,28	90,31
	Fallback	-	15,83	-
Magic Gamma Telescope (mgt)	CPU Time	0,228125	1,725	3,8656
	Accuracy	83,61	83,91	83,65
	Avg Precision	84,33	84,21	84,01
	Avg Recall	83,61	83,91	83,65
	F1-Score	82,75	83,25	82,95
	Fallback	-	9,12	-
	CPU Time	0,0	0,0125	0,0531
	Accuracy	92,27	95,78	97,41

Dataset	Μετρική	KNN	Mutual	Symmetric
MONK's	Avg Precision	92,43	95,98	97,61
Problems (mnk)	Avg Recall	92,27	95,78	97,41
	F1-Score	92,26	95,77	97,41
	Fallback	-	0,00	-
	CPU Time	0,31875	3,58125	5,0531
Pen-Based Digits (pd)	Accuracy	99,21	99,38	99,16
	Avg Precision	99,22	99,38	99,17
	Avg Recall	99,21	99,38	99,16
	F1-Score	99,21	99,38	99,16
Phoneme (ph)	Fallback	-	8,34	-
	CPU Time	0,015625	0,253125	0,6906
	Accuracy	86,38	87,01	86,27
	Avg Precision	86,22	86,88	86,14
	Avg Recall	86,38	87,01	86,27
	F1-Score	86,19	86,85	86,08
Pima Diabetes (pm)	Fallback	-	6,19	-
	CPU Time	0,00625	0,040625	0,0969
	Accuracy	73,49	72,05	74,54
	Avg Precision	72,97	71,24	73,99
	Avg Recall	73,49	72,05	74,54
	F1-Score	72,70	70,98	73,87
Ringnorm (ring)	Fallback	-	10,11	-
	CPU Time	0,078125	6,009375	2,0875
	Accuracy	56,67	65,57	62,56
	Avg Precision	76,70	79,55	71,48
	Avg Recall	56,67	65,57	62,56
	F1-Score	46,31	60,77	57,98
Segment (sg)	Fallback	-	53,78	-
	CPU Time	0,0	0,9125	0,9563
	Accuracy	80,21	81,03	80,47
	Avg Precision	72,67	73,26	71,61

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
	Avg Recall	80,21	81,03	80,47
	F1-Score	75,46	76,22	75,18
	Fallback	-	19,88	-
Shuttle (sh)	CPU Time	1,315625	7,634375	47,9031
	Accuracy	99,82	99,87	99,85
	Avg Precision	99,81	99,87	99,84
	Avg Recall	99,82	99,87	99,85
	F1-Score	99,81	99,87	99,83
	Fallback	-	3,23	-
Twonorm (tn)	CPU Time	0,1125	3,990625	2,3531
	Accuracy	96,89	95,42	97,08
	Avg Precision	96,89	95,44	97,09
	Avg Recall	96,89	95,42	97,08
	F1-Score	96,89	95,41	97,08
	Fallback	-	26,93	-
Texture (txr)	CPU Time	0,134375	1,740625	2,9531
	Accuracy	97,82	98,27	98,23
	Avg Precision	97,85	98,29	98,25
	Avg Recall	97,82	98,27	98,23
	F1-Score	97,82	98,27	98,23
	Fallback	-	5,50	-
Waveform (wf)	CPU Time	0,09375	2,296875	1,575
	Accuracy	83,04	79,68	84,1
	Avg Precision	83,45	79,70	84,36
	Avg Recall	83,04	79,68	84,10
	F1-Score	82,83	79,65	83,98
	Fallback	-	22,60	-
Wine Quality (wine)	CPU Time	0,04375	0,565625	0,7219
	Accuracy	49,5	48,21	49,73
	Avg Precision	38,91	37,15	39,10
	Avg Recall	49,50	48,21	49,73

Dataset	Μετρική	KNN	Mutual	Symmetric
Yeast (ys)	F1-Score	42,42	41,55	42,58
	Fallback	-	29,80	-
	CPU Time	0,003125	0,090625	0,1844
	Accuracy	58,39	54,67	58,52
	Avg Precision	58,89	54,15	58,51
	Avg Recall	58,39	54,67	58,52
	F1-Score	57,65	54,01	57,79
	Fallback	-	12,52	-

Για $k=5$: Ο Symmetric k -NN διατηρεί πολύ καλή απόδοση, με υψηλότερη ακρίβεια στα Balance, Letter, Monk, Pima, Twonorm και Waveform. Στο Monk επιτυγχάνει 97,41%, ενώ ο Mutual k -NN παρουσιάζει την καλύτερη επίδοση στο Ring με 65,57%. Ο k -NN παραμένει ο ταχύτερος, ενώ το Fallback του Mutual αυξάνεται σε αρκετά σύνολα.

Πίνακας 5.16: Σύγκριση αλγορίθμων μετά από χρήση του ENN για $k = 7$:

Dataset	Μετρική	KNN	Mutual	Symmetric
Balance (bl)	CPU Time	0,0	0,025	0,0813
	Accuracy	89,02	88,21	89,5
	Avg Precision	82,06	81,54	82,62
	Avg Recall	89,02	88,21	89,50
	F1-Score	85,36	84,61	85,85
	Fallback	-	0,00	-
	CPU Time	0,009375	0,15625	0,6469
Banana (bn)	Accuracy	89,95	89,69	90,05
	Avg Precision	89,98	89,72	90,10
	Avg Recall	89,95	89,69	90,05
	F1-Score	89,93	89,66	90,01
	Fallback	-	0,30	-
	CPU Time	0,003125	0,015625	0,0375
Ecoli (ecl)	Accuracy	76,36	79,09	75,15
	Avg Precision	74,03	76,66	72,83
	Avg Recall	76,36	79,09	75,15
	F1-Score	73,37	76,92	71,87
	Fallback	-	-	-

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
	Fallback	-	2,42	-
	CPU Time	0,3875	3,38125	5,6562
	Accuracy	42,42	43,19	43,53
Eye State (eeg)	Avg Precision	59,87	59,61	59,83
	Avg Recall	42,42	43,19	43,53
	F1-Score	43,63	44,63	44,79
	Fallback	-	52,38	-
	CPU Time	23,115625	150,065625	1173,55
	Accuracy	99,71	99,74	99,63
KDD Cup (kdd)	Avg Precision	99,68	99,72	99,60
	Avg Recall	99,71	99,74	99,63
	F1-Score	99,69	99,73	99,61
	Fallback	-	2,88	-
	CPU Time	0,621875	6,228125	12,9313
	Accuracy	93,11	93,82	93,85
Letter	Avg Precision	93,29	93,92	93,96
Recognition (lir)	Avg Recall	93,11	93,82	93,85
	F1-Score	93,13	93,82	93,85
	Fallback	-	3,19	-
	CPU Time	0,153125	2,3125	3,3125
	Accuracy	89,56	90,54	89,91
Landsat	Avg Precision	89,62	90,50	89,83
Satellite (ls)	Avg Recall	89,56	90,54	89,91
	F1-Score	89,54	90,46	89,81
	Fallback	-	8,85	-
	CPU Time	0,246875	1,65	3,9375
	Accuracy	83,48	84,32	83,7
Magic Gamma	Avg Precision	84,33	84,65	84,13
Telescope (mgt)	Avg Recall	83,48	84,32	83,70
	F1-Score	82,55	83,69	82,96
	Fallback	-	4,46	-

Dataset	Μετρική	KNN	Mutual	Symmetric
MONK's Problems (mnk)	CPU Time	0,0	0,01875	0,0563
	Accuracy	90,16	94,37	96,95
	Avg Precision	90,66	94,92	97,20
	Avg Recall	90,16	94,37	96,95
	F1-Score	90,12	94,35	96,94
	Fallback	-	0,00	-
Pen-Based Digits (pd)	CPU Time	0,278125	3,053125	5,0656
	Accuracy	99,07	99,4	99,07
	Avg Precision	99,08	99,40	99,08
	Avg Recall	99,07	99,40	99,07
	F1-Score	99,07	99,40	99,07
	Fallback	-	4,01	-
Phoneme (ph)	CPU Time	0,0125	0,21875	0,6906
	Accuracy	85,75	86,42	86,0
	Avg Precision	85,58	86,26	85,83
	Avg Recall	85,75	86,42	86,00
	F1-Score	85,53	86,23	85,76
	Fallback	-	2,95	-
Pima Diabetes (pm)	CPU Time	0,003125	0,040625	0,0969
	Accuracy	74,02	72,84	74,41
	Avg Precision	73,48	72,13	73,71
	Avg Recall	74,02	72,84	74,41
	F1-Score	72,94	71,75	73,46
	Fallback	-	5,25	-
Ringnorm (ring)	CPU Time	0,09375	5,8875	2,1
	Accuracy	55,43	66,91	60,37
	Avg Precision	76,34	79,95	70,59
	Avg Recall	55,43	66,91	60,37
	F1-Score	43,99	62,70	54,45
	Fallback	-	46,86	-
	CPU Time	0,01875	0,878125	0,975

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Segment (sg)	Accuracy	79,69	81,03	80,47
	Avg Precision	72,62	73,29	71,79
	Avg Recall	79,69	81,03	80,47
	F1-Score	75,16	76,20	75,23
	Fallback	-	17,49	-
Shuttle (sh)	CPU Time	1,56875	8,63125	49,125
	Accuracy	99,81	99,87	99,83
	Avg Precision	99,81	99,86	99,81
	Avg Recall	99,81	99,87	99,83
	F1-Score	99,80	99,86	99,81
	Fallback	-	1,63	-
Twonorm (tn)	CPU Time	0,05625	3,403125	2,3719
	Accuracy	97,12	95,59	97,08
	Avg Precision	97,12	95,62	97,09
	Avg Recall	97,12	95,59	97,08
	F1-Score	97,12	95,59	97,08
	Fallback	-	17,78	-
Texture (txr)	CPU Time	0,08125	1,540625	3,0656
	Accuracy	97,78	98,23	97,94
	Avg Precision	97,81	98,26	97,97
	Avg Recall	97,78	98,23	97,94
	F1-Score	97,78	98,24	97,94
	Fallback	-	2,15	-
Waveform (wf)	CPU Time	0,05625	1,853125	1,5656
	Accuracy	83,58	80,64	84,76
	Avg Precision	84,09	80,63	85,06
	Avg Recall	83,58	80,64	84,76
	F1-Score	83,36	80,60	84,64
	Fallback	-	14,25	-
	CPU Time	0,04375	0,453125	0,7312
	Accuracy	49,44	48,11	50,67

Dataset	Μετρική	KNN	Mutual	Symmetric
Wine Quality (wine)	Avg Precision	38,46	36,98	39,35
	Avg Recall	49,44	48,11	50,67
	F1-Score	42,27	41,39	43,30
	Fallback	-	17,47	-
Yeast (ys)	CPU Time	0,00625	0,071875	0,1781
	Accuracy	58,66	56,36	59,07
	Avg Precision	59,56	56,07	59,30
	Avg Recall	58,66	56,36	59,07
	F1-Score	57,78	55,71	58,33
	Fallback	-	6,50	-

Για $k=7$: Ο Symmetric k-NN παρουσιάζει και εδώ υψηλή απόδοση, με χαρακτηριστικό παράδειγμα το Monk (96,95%), ενώ επιτυγχάνει επίσης πολύ υψηλή ακρίβεια στα Letter, Kddcup και Shuttle. Ο Mutual k-NN παρουσιάζει την καλύτερη επίδοση στο Ring (66,91%) και στο Ecoli (79,09%). Ο k-NN παραμένει ο ταχύτερος, ενώ το Fallback του Mutual είναι γενικά χαμηλότερο από τις μικρότερες τιμές του kk, αν και στο Ring φτάνει το 46,86%.

Γενικά, μετά την εφαρμογή του ENN, ο Symmetric k-NN εμφανίζει συχνά την καλύτερη ακρίβεια, ενώ ο k-NN διατηρεί το πλεονέκτημα ως προς τον χρόνο εκτέλεσης.

Πίνακας 5.17: Σύγκριση αλγορίθμων μετά από συνδυαστική χρήση του ENN→CNN για $k = 1$:

Dataset	Μετρική	KNN	Mutual	Symmetric
Balance (bl)	CPU Time	0	0,1625	0,0781
	Accuracy	83,68	83,68	84,49
	Avg Precision	77,43	77,43	79,20
	Avg Recall	83,68	83,68	84,49
	F1-Score	80,40	80,40	81,13
	Fallback	-	100,00	-
Banana (bn)	CPU Time	0	1,2125	0,5875
	Accuracy	88,93	88,93	89,08
	Avg Precision	88,94	88,94	89,36
	Avg Recall	88,93	88,93	89,08
	F1-Score	88,91	88,91	89,00

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
	Fallback	-	99,98	-
	CPU Time	0,003125	0,078125	0,0375
Ecoli (ec1)	Accuracy	74,85	74,85	75,15
	Avg Precision	73,63	73,63	74,09
	Avg Recall	74,85	74,85	75,15
	F1-Score	73,24	73,24	72,82
	Fallback	-	100,00	-
	CPU Time	0,090625	3,68125	1,8906
Eye State (eeg)	Accuracy	44,18	44,18	46,32
	Avg Precision	59,42	59,42	61,21
	Avg Recall	44,18	44,18	46,32
	F1-Score	46,28	46,28	47,52
	Fallback	-	100,00	-
	CPU Time	0,228125	249,66875	17
KDD Cup (kdd)	Accuracy	99,34	99,34	93,46
	Avg Precision	99,37	99,37	99,05
	Avg Recall	99,34	99,34	93,46
	F1-Score	99,34	99,34	94,43
	Fallback	-	100,00	-
	CPU Time	0,075	27,35	3,1812
Letter Recognition (lir)	Accuracy	91,35	91,35	88,71
	Avg Precision	91,50	91,50	89,08
	Avg Recall	91,35	91,35	88,71
	F1-Score	91,37	91,37	88,73
	Fallback	-	100,00	-
	CPU Time	0,0375	11,534375	1,35
Landsat Satellite (ls)	Accuracy	88,54	88,54	87,20
	Avg Precision	88,75	88,75	87,08
	Avg Recall	88,54	88,54	87,20
	F1-Score	88,60	88,60	87,03
	Fallback	-	100,00	-
	CPU Time	0,0375	11,534375	1,35

Dataset	Μετρική	KNN	Mutual	Symmetric
Magic Gamma Telescope (mgt)	CPU Time	0,084375	4,509375	2,2156
	Accuracy	79,64	79,64	82,84
	Avg Precision	79,28	79,28	82,99
	Avg Recall	79,64	79,64	82,84
	F1-Score	79,24	79,24	82,17
	Fallback	-	100,00	-
MONK's Problems (mnk)	CPU Time	0,003125	0,096875	0,05
	Accuracy	80,79	80,79	88,75
	Avg Precision	80,83	80,83	90,63
	Avg Recall	80,79	80,79	88,75
	F1-Score	80,78	80,78	88,65
	Fallback	-	100,00	-
Pen-Based Digits (pd)	CPU Time	0,0375	19,571875	1,8
	Accuracy	98,43	98,43	96,83
	Avg Precision	98,44	98,44	96,97
	Avg Recall	98,43	98,43	96,83
	F1-Score	98,43	98,43	96,85
	Fallback	-	100,00	-
Phoneme (ph)	CPU Time	0,025	1,340625	0,5844
	Accuracy	86,38	86,38	84,77
	Avg Precision	86,40	86,40	84,50
	Avg Recall	86,38	86,38	84,77
	F1-Score	86,34	86,34	84,34
	Fallback	-	100,00	-
Pima Diabetes (pm)	CPU Time	0,003125	0,178125	0,0875
	Accuracy	72,70	72,70	73,62
	Avg Precision	72,45	72,45	73,10
	Avg Recall	72,70	72,70	73,62
	F1-Score	72,31	72,31	72,23
	Fallback	-	100,00	-
	CPU Time	0,01875	13,2875	1,4187

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Ringnorm (ring)	Accuracy	78,53	78,53	63,46
	Avg Precision	84,51	84,51	66,38
	Avg Recall	78,53	78,53	63,46
	F1-Score	77,50	77,50	61,53
	Fallback	-	100,00	-
Segment (sg)	CPU Time	0,00625	3,328125	0,8406
	Accuracy	79,99	79,99	75,04
	Avg Precision	73,31	73,31	65,78
	Avg Recall	79,99	79,99	75,04
	F1-Score	75,61	75,61	69,59
	Fallback	-	98,96	-
Shuttle (sh)	CPU Time	0,025	13,378125	6,2125
	Accuracy	97,27	97,27	96,26
	Avg Precision	98,56	98,56	97,91
	Avg Recall	97,27	97,27	96,26
	F1-Score	97,77	97,77	96,80
	Fallback	-	100,00	-
Twonorm (tn)	CPU Time	0,040625	16,3875	1,5
	Accuracy	91,72	91,72	92,79
	Avg Precision	91,73	91,73	92,84
	Avg Recall	91,72	91,72	92,79
	F1-Score	91,72	91,72	92,79
	Fallback	-	100,00	-
Texture (txr)	CPU Time	0,015625	9,75	1,2156
	Accuracy	96,67	96,67	95,49
	Avg Precision	96,72	96,72	95,64
	Avg Recall	96,67	96,67	95,49
	F1-Score	96,68	96,68	95,48
	Fallback	-	100,00	-
	CPU Time	0,0125	11,009375	1,2031
	Accuracy	74,77	74,77	77,94

Dataset	Μετρική	KNN	Mutual	Symmetric
Waveform (wf)	Avg Precision	74,77	74,77	78,04
	Avg Recall	74,77	74,77	77,94
	F1-Score	74,71	74,71	77,94
	Fallback	-	100,00	-
Wine Quality (wine)	CPU Time	0,03125	1,175	0,55
	Accuracy	46,37	46,37	45,09
	Avg Precision	35,61	35,61	34,78
	Avg Recall	46,37	46,37	45,09
	F1-Score	39,99	39,99	39,11
	Fallback	-	100,00	-
	CPU Time	0,003125	0,346875	0,1656
	Accuracy	53,18	53,18	53,92
Yeast (ys)	Avg Precision	53,25	53,25	53,97
	Avg Recall	53,18	53,18	53,92
	F1-Score	52,40	52,40	53,02
	Fallback	-	100,00	-

Για $k=1$: Μετά τη διαδοχική εφαρμογή ENN και CNN, ο Symmetric k -NN παρουσιάζει καλύτερη ακρίβεια σε ορισμένα σύνολα, όπως Monk (88,75%), EEG Eye State (46,32%) και Waveform (77,94%). Αντίθετα, στα Kddcup, Letter και Ring οι k -NN και Mutual k -NN παρουσιάζουν καλύτερα αποτελέσματα. Ο k -NN παραμένει ο ταχύτερος, ενώ ο Mutual εμφανίζει Fallback σχεδόν 100%.

Πίνακας 5.18: Σύγκριση αλγορίθμων μετά από συνδυαστική χρήση του ENN→CNN για $k = 3$:

Dataset	Μετρική	KNN	Mutual	Symmetric
Balance (bl)	CPU Time	0,003125	0,046875	0,075
	Accuracy	85,30	83,04	84,82
	Avg Precision	78,61	77,24	78,75
	Avg Recall	85,30	83,04	84,82
	F1-Score	81,79	79,64	81,35
	Fallback	-	15,98	-
	CPU Time	0	0,390625	0,5719

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Banana (bn)	Accuracy	89,10	88,76	89,14
	Avg Precision	89,11	89,00	89,34
	Avg Recall	89,10	88,76	89,14
	F1-Score	89,09	88,68	89,07
	Fallback	-	22,40	-
Ecoli (ec1)	CPU Time	0,009375	0,015625	0,0406
	Accuracy	68,79	72,12	70,91
	Avg Precision	66,86	71,91	69,31
	Avg Recall	68,79	72,12	70,91
	F1-Score	64,81	69,65	66,78
	Fallback	-	10,00	-
Eye State (eeg)	CPU Time	0,103125	2,325	1,9406
	Accuracy	43,01	44,19	44,57
	Avg Precision	58,25	59,50	60,13
	Avg Recall	43,01	44,19	44,57
	F1-Score	44,65	45,74	45,36
	Fallback	-	55,85	-
KDD Cup (kdd)	CPU Time	0,265625	48,35625	17,125
	Accuracy	98,70	98,97	92,24
	Avg Precision	98,78	99,07	97,56
	Avg Recall	98,70	98,97	92,24
	F1-Score	98,70	98,99	92,80
	Fallback	-	16,57	-
Letter Recognition (lir)	CPU Time	0,075	5,05625	3,25
	Accuracy	84,79	88,37	85,37
	Avg Precision	86,13	88,79	86,18
	Avg Recall	84,79	88,37	85,37
	F1-Score	85,02	88,41	85,46
	Fallback	-	7,11	-
	CPU Time	0	2,784375	1,3656
	Accuracy	88,07	88,04	87,49

Dataset	Μετρική	KNN	Mutual	Symmetric
Landsat	Avg Precision	88,02	87,94	87,13
Satellite (ls)	Avg Recall	88,07	88,04	87,49
	F1-Score	87,99	87,93	87,18
	Fallback	-	20,81	-
Magic Gamma Telescope (mgt)	CPU Time	0,065625	1,665625	2,2625
	Accuracy	81,56	80,51	83,30
	Avg Precision	81,49	80,28	83,45
	Avg Recall	81,56	80,51	83,30
	F1-Score	80,89	79,88	82,67
	Fallback	-	25,53	-
	CPU Time	0	0,04375	0,05
MONK's Problems (mnk)	Accuracy	78,66	79,85	87,58
	Avg Precision	79,08	80,61	89,12
	Avg Recall	78,66	79,85	87,58
	F1-Score	78,52	79,76	87,49
	Fallback	-	35,64	-
Pen-Based Digits (pd)	CPU Time	0,0375	2,303125	1,8219
	Accuracy	96,96	96,52	96,22
	Avg Precision	97,02	96,62	96,35
	Avg Recall	96,96	96,52	96,22
	F1-Score	96,97	96,53	96,24
	Fallback	-	7,76	-
Phoneme (ph)	CPU Time	0,00625	0,3125	0,5969
	Accuracy	84,33	84,75	82,94
	Avg Precision	84,34	84,49	82,65
	Avg Recall	84,33	84,75	82,94
	F1-Score	84,30	84,34	82,65
	Fallback	-	14,36	-
Pima	CPU Time	0,003125	0,053125	0,0813
	Accuracy	72,45	72,84	72,57
	Avg Precision	71,71	71,94	71,94

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Diabetes (pm)	Avg Recall	72,45	72,84	72,57
	F1-Score	71,75	71,51	71,23
	Fallback	-	18,89	-
Ringnorm (ring)	CPU Time	0	4,0375	1,4531
	Accuracy	79,36	84,21	50,82
	Avg Precision	85,36	87,02	51,86
	Avg Recall	79,36	84,21	50,82
	F1-Score	78,39	83,88	42,65
	Fallback	-	27,34	-
	CPU Time	0,009375	1,01875	0,8125
Segment (sg)	Accuracy	74,26	75,82	68,36
	Avg Precision	69,83	68,66	63,18
	Avg Recall	74,26	75,82	68,36
	F1-Score	70,24	71,54	63,91
	Fallback	-	22,83	-
	CPU Time	0,021875	4,20625	6,2531
Shuttle (sh)	Accuracy	94,98	94,27	95,39
	Avg Precision	97,98	98,08	96,01
	Avg Recall	94,98	94,27	95,39
	F1-Score	95,96	95,39	95,32
	Fallback	-	21,34	-
	CPU Time	0,075	7,484375	1,4531
Twonorm (tn)	Accuracy	94,81	92,17	95,58
	Avg Precision	94,82	92,22	95,61
	Avg Recall	94,81	92,17	95,58
	F1-Score	94,81	92,17	95,58
	Fallback	-	42,51	-
	CPU Time	0,015625	1,265625	1,2312
Texture (txr)	Accuracy	91,90	95,23	92,28
	Avg Precision	92,66	95,37	92,97
	Avg Recall	91,90	95,23	92,28

Dataset	Μετρική	KNN	Mutual	Symmetric
Waveform (wf)	F1-Score	91,89	95,24	92,29
	Fallback	-	7,70	-
	CPU Time	0,0125	4,228125	1,1906
	Accuracy	78,54	75,58	81,48
	Avg Precision	78,59	75,65	81,52
	Avg Recall	78,54	75,58	81,48
	F1-Score	78,43	75,59	81,44
Wine Quality (wine)	Fallback	-	34,67	-
	CPU Time	0,009375	0,465625	0,5563
	Accuracy	48,27	45,90	47,60
	Avg Precision	37,27	35,41	36,66
	Avg Recall	48,27	45,90	47,60
	F1-Score	41,57	39,85	41,09
	Fallback	-	30,98	-
Yeast (ys)	CPU Time	0,003125	0,1125	0,1656
	Accuracy	53,11	53,58	55,88
	Avg Precision	53,04	52,99	55,09
	Avg Recall	53,11	53,58	55,88
	F1-Score	51,89	52,71	54,59
	Fallback	-	23,35	-

Για $k=3$: Τα αποτελέσματα είναι περισσότερο μοιρασμένα μεταξύ των τριών αλγορίθμων. Ο Symmetric k -NN υπερέρχει σε σύνολα όπως Monk (87,58%), Magic Gamma Telescope (83,30%) και Waveform (81,48%), ενώ ο Mutual παρουσιάζει υψηλότερη ακρίβεια σε Kddcup (98,97%), Letter (88,37%) και Ring (84,21%). Ο k -NN εξακολουθεί να έχει σαφώς μικρότερο CPU Time.

Πίνακας 5.19: Σύγκριση αλγορίθμων μετά από συνδυαστική χρήση του ENN→CNN για $k = 5$:

Dataset	Μετρική	KNN	Mutual	Symmetric
Balance (bl)	CPU Time	0,00625	0,025	0,0719
	Accuracy	83,68	82,55	83,20
	Avg Precision	77,35	76,95	77,35
	Avg Recall	83,68	82,55	83,20

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
	F1-Score	80,26	79,23	79,82
	Fallback	-	5,49	-
Banana (bn)	CPU Time	0	0,25	0,5719
	Accuracy	88,63	88,61	88,65
	Avg Precision	88,67	88,91	88,85
	Avg Recall	88,63	88,61	88,65
	F1-Score	88,59	88,52	88,57
	Fallback	-	9,82	-
Ecoli (ec1)	CPU Time	0	0,0125	0,0437
	Accuracy	68,48	70,30	70,00
	Avg Precision	62,54	67,67	68,52
	Avg Recall	68,48	70,30	70,00
	F1-Score	62,90	66,90	65,43
	Fallback	-	3,33	-
Eye State (eeg)	CPU Time	0,115625	1,721875	1,9406
	Accuracy	43,32	44,74	44,03
	Avg Precision	58,27	59,71	59,20
	Avg Recall	43,32	44,74	44,03
	F1-Score	44,57	45,80	44,51
	Fallback	-	35,43	-
KDD Cup (kdd)	CPU Time	0,265625	34,45	17,2375
	Accuracy	97,73	98,69	91,33
	Avg Precision	97,94	98,79	93,82
	Avg Recall	97,73	98,69	91,33
	F1-Score	97,76	98,71	89,75
	Fallback	-	10,79	-
Letter Recognition (lir)	CPU Time	0,075	3,95	3,2344
	Accuracy	82,74	87,07	82,40
	Avg Precision	84,22	87,63	83,45
	Avg Recall	82,74	87,07	82,40
	F1-Score	82,96	87,12	82,47

Dataset	Μετρική	KNN	Mutual	Symmetric
	Fallback	-	2,12	-
	CPU Time	0	1,64375	1,3469
	Accuracy	87,32	87,70	87,81
Landsat	Avg Precision	87,25	87,51	87,50
Satellite (ls)	Avg Recall	87,32	87,70	87,81
	F1-Score	87,22	87,51	87,51
	Fallback	-	10,25	-
	CPU Time	0,065625	1,03125	2,2594
	Accuracy	82,17	81,43	83,23
Magic Gamma	Avg Precision	82,36	81,42	83,35
Telescope (mgt)	Avg Recall	82,17	81,43	83,23
	F1-Score	81,40	80,70	82,61
	Fallback	-	10,40	-
	CPU Time	0,003125	0,025	0,0469
	Accuracy	78,67	76,32	84,98
MONK's	Avg Precision	78,93	78,47	86,46
Problems (mnk)	Avg Recall	78,67	76,32	84,98
	F1-Score	78,56	76,04	84,89
	Fallback	-	11,73	-
	CPU Time	0,075	1,175	1,8219
	Accuracy	95,22	96,77	95,18
Pen-Based	Avg Precision	95,36	96,85	95,36
Digits (pd)	Avg Recall	95,22	96,77	95,18
	F1-Score	95,23	96,78	95,19
	Fallback	-	2,22	-
	CPU Time	0,00625	0,215625	0,5969
	Accuracy	83,14	83,79	81,10
Phoneme (ph)	Avg Precision	83,21	83,49	80,85
	Avg Recall	83,14	83,79	81,10
	F1-Score	83,14	83,35	80,89
	Fallback	-	6,11	-

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Pima Diabetes (pm)	CPU Time	0,009375	0,034375	0,0844
	Accuracy	73,10	74,02	73,22
	Avg Precision	72,36	73,58	72,42
	Avg Recall	73,10	74,02	73,22
	F1-Score	72,12	72,39	72,07
	Fallback	-	6,69	-
Ringnorm (ring)	CPU Time	0,05625	2,421875	1,4625
	Accuracy	79,86	89,33	46,92
	Avg Precision	85,65	90,46	38,02
	Avg Recall	79,86	89,33	46,92
	F1-Score	78,95	89,25	34,53
	Fallback	-	14,83	-
Segment (sg)	CPU Time	0,015625	0,7125	0,8625
	Accuracy	66,41	74,39	63,50
	Avg Precision	65,87	67,44	60,72
	Avg Recall	66,41	74,39	63,50
	F1-Score	61,53	70,12	59,51
	Fallback	-	17,62	-
Shuttle (sh)	CPU Time	0,028125	2,9375	6,3
	Accuracy	92,53	93,37	93,76
	Avg Precision	96,95	97,43	95,12
	Avg Recall	92,53	93,37	93,76
	F1-Score	93,96	94,64	93,87
	Fallback	-	10,81	-
Twonorm (tn)	CPU Time	0	4,81875	1,4531
	Accuracy	95,75	92,97	96,35
	Avg Precision	95,76	93,05	96,37
	Avg Recall	95,75	92,97	96,35
	F1-Score	95,75	92,96	96,35
	Fallback	-	25,99	-
	CPU Time	0,025	0,640625	1,2156

Dataset	Μετρική	KNN	Mutual	Symmetric
Texture (txr)	Accuracy	82,53	94,72	82,64
	Avg Precision	86,65	94,93	86,88
	Avg Recall	82,53	94,72	82,64
	F1-Score	80,75	94,73	80,74
	Fallback	-	2,64	-
Waveform (wf)	CPU Time	0,025	2,346875	1,225
	Accuracy	80,48	76,70	81,82
	Avg Precision	80,59	76,75	81,88
	Avg Recall	80,48	76,70	81,82
	F1-Score	80,38	76,71	81,75
	Fallback	-	17,80	-
Wine Quality (wine)	CPU Time	0,015625	0,29375	0,5563
	Accuracy	49,60	45,56	49,28
	Avg Precision	38,19	35,16	37,88
	Avg Recall	49,60	45,56	49,28
	F1-Score	42,47	39,58	42,13
	Fallback	-	14,69	-
Yeast (ys)	CPU Time	0,003125	0,06875	0,1594
	Accuracy	55,27	55,34	53,92
	Avg Precision	55,18	55,11	53,57
	Avg Recall	55,27	55,34	53,92
	F1-Score	54,04	54,32	52,54
	Fallback	-	9,00	-

Για $k=5$: Ο Mutual k -NN παρουσιάζει ιδιαίτερα καλή απόδοση σε αρκετά σύνολα, όπως Kddcup (98,69%), Letter (87,07%), Pendigits (96,77%) και Ring (89,33%). Ο Symmetric k -NN διατηρεί καλύτερη απόδοση σε περιπτώσεις όπως Monk (84,98%) και Twonorm (96,35%), ενώ ο k -NN παραμένει ο ταχύτερος αλγόριθμος.

Πίνακας 5.20: Σύγκριση αλγορίθμων μετά από συνδυαστική χρήση του ENN→CNN για $k = 7$:

Dataset	Μετρική	KNN	Mutual	Symmetric
	CPU Time	0,00625	0,028125	0,0781

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Balance (bl)	Accuracy	83,69	81,75	82,07
	Avg Precision	77,22	76,35	76,04
	Avg Recall	83,69	81,75	82,07
	F1-Score	80,22	78,47	78,70
	Fallback	-	3,07	-
Banana (bn)	CPU Time	0,003125	0,2	0,5656
	Accuracy	87,87	88,02	87,74
	Avg Precision	87,98	88,48	88,10
	Avg Recall	87,87	88,02	87,74
	F1-Score	87,81	87,90	87,62
	Fallback	-	4,72	-
Ecoli (ecl)	CPU Time	0,009375	0,0125	0,0344
	Accuracy	65,76	67,58	67,27
	Avg Precision	59,94	63,95	65,03
	Avg Recall	65,76	67,58	67,27
	F1-Score	60,46	63,46	62,98
	Fallback	-	1,52	-
Eye State (eeg)	CPU Time	0,134375	1,35	1,9563
	Accuracy	42,82	44,64	43,08
	Avg Precision	57,64	58,83	58,34
	Avg Recall	42,82	44,64	43,08
	F1-Score	44,02	45,49	43,48
	Fallback	-	24,51	-
KDD Cup (kdd)	CPU Time	0,290625	26,6875	17,225
	Accuracy	96,95	98,49	93,75
	Avg Precision	97,13	98,59	97,23
	Avg Recall	96,95	98,49	93,75
	F1-Score	96,89	98,49	95,00
	Fallback	-	7,60	-
	CPU Time	0,075	3,678125	3,2344
	Accuracy	80,10	86,36	79,13

Dataset	Μετρική	KNN	Mutual	Symmetric
Letter	Avg Precision	81,99	86,95	80,52
Recognition (lir)	Avg Recall	80,10	86,36	79,13
	F1-Score	80,37	86,38	79,23
	Fallback	-	0,89	-
	CPU Time	0,0375	1,1	1,3844
Landsat Satellite (ls)	Accuracy	86,83	87,70	87,09
	Avg Precision	86,71	87,45	86,66
	Avg Recall	86,83	87,70	87,09
	F1-Score	86,69	87,45	86,69
Magic Gamma Telescope (mgt)	Fallback	-	5,94	-
	CPU Time	0,075	0,821875	2,2594
	Accuracy	82,33	81,52	83,04
	Avg Precision	82,70	81,60	83,17
MONK's Problems (mnk)	Avg Recall	82,33	81,52	83,04
	F1-Score	81,47	80,75	82,39
	Fallback	-	5,44	-
	CPU Time	0,003125	0,025	0,0469
Pen-Based Digits (pd)	Accuracy	78,91	75,62	83,58
	Avg Precision	79,10	77,78	84,50
	Avg Recall	78,91	75,62	83,58
	F1-Score	78,85	75,32	83,54
Phoneme (ph)	Fallback	-	3,76	-
	CPU Time	0,01875	0,95	1,825
	Accuracy	93,58	96,32	92,04
	Avg Precision	93,90	96,41	92,62
Phoneme (ph)	Avg Recall	93,58	96,32	92,04
	F1-Score	93,60	96,34	91,98
	Fallback	-	1,08	-
	CPU Time	0,003125	0,190625	0,5969
Phoneme (ph)	Accuracy	81,94	82,77	80,97
	Avg Precision	82,08	82,43	80,74

Κεφάλαιο 5: Πειραματική Μελέτη

Dataset	Μετρική	KNN	Mutual	Symmetric
Pima Diabetes (pm)	Avg Recall	81,94	82,77	80,97
	F1-Score	81,97	82,35	80,81
	Fallback	-	3,28	-
	CPU Time	0,003125	0,03125	0,0813
	Accuracy	73,49	74,80	74,15
	Avg Precision	72,95	74,49	73,61
	Avg Recall	73,49	74,80	74,15
	F1-Score	72,17	73,17	72,91
	Fallback	-	4,98	-
	CPU Time	0	1,678125	1,4625
	Accuracy	80,32	92,29	46,34
	Avg Precision	85,91	92,83	31,78
Ringnorm (ring)	Avg Recall	80,32	92,29	46,34
	F1-Score	79,47	92,27	32,69
	Fallback	-	9,03	-
	CPU Time	0,009375	0,834375	0,8438
	Accuracy	57,98	73,52	50,00
	Avg Precision	48,36	66,61	41,88
Segment (sg)	Avg Recall	57,98	73,52	50,00
	F1-Score	50,17	69,34	42,81
	Fallback	-	16,10	-
	CPU Time	0,03125	2,209375	6,2781
	Accuracy	92,12	92,25	94,92
	Avg Precision	96,36	96,41	95,52
Shuttle (sh)	Avg Recall	92,12	92,25	94,92
	F1-Score	93,69	93,66	94,82
	Fallback	-	5,64	-
	CPU Time	0	3,38125	1,4781
	Accuracy	96,24	93,67	96,55
	Avg Precision	96,25	93,73	96,56
Twonorm (tn)	Avg Recall	96,24	93,67	96,55

Dataset	Μετρική	KNN	Mutual	Symmetric
	F1-Score	96,24	93,67	96,55
	Fallback	-	17,60	-
Texture (txr)	CPU Time	0,046875	0,515625	1,2125
	Accuracy	76,25	94,59	76,92
	Avg Precision	81,44	94,81	81,98
	Avg Recall	76,25	94,59	76,92
	F1-Score	71,01	94,61	71,89
	Fallback	-	1,18	-
Waveform (wf)	CPU Time	0	1,5375	1,2031
	Accuracy	81,66	77,44	82,56
	Avg Precision	81,89	77,54	82,68
	Avg Recall	81,66	77,44	82,56
	F1-Score	81,53	77,46	82,46
	Fallback	-	10,53	-
Wine Quality (wine)	CPU Time	0,015625	0,228125	0,5656
	Accuracy	49,99	45,88	49,99
	Avg Precision	38,69	35,32	39,05
	Avg Recall	49,99	45,88	49,99
	F1-Score	42,66	39,81	42,65
	Fallback	-	8,22	-
Yeast (ys)	CPU Time	0,00625	0,0625	0,1625
	Accuracy	54,73	56,63	54,46
	Avg Precision	53,52	57,08	53,71
	Avg Recall	54,73	56,63	54,46
	F1-Score	53,15	55,91	52,83
	Fallback	-	5,08	-

Για $k=7$: Ο Mutual k -NN εμφανίζει υψηλή ακρίβεια σε αρκετά σύνολα, όπως Kddcup (98,49%), Letter (86,36%), Pendigits (96,32%) και ιδιαίτερα Ring (92,29%). Ο Symmetric k -NN υπερέχει σε σύνολα όπως Monk (83,58%), Magic Gamma Telescope (83,04%) και Waveform (82,56%). Και σε αυτή την περίπτωση, ο k -NN διατηρεί το πλεονέκτημα ως προς τον CPU Time.

Γενικά, μετά τον συνδυασμό ENN → CNN δεν παρατηρείται ένας αλγόριθμος που να υπερέχει σταθερά ως προς την ακρίβεια. Οι Mutual και Symmetric παρουσιάζουν καλύτερα αποτελέσματα σε διαφορετικά σύνολα δεδομένων, ενώ ο k-NN διατηρεί σταθερά το μικρότερο υπολογιστικό κόστος.

5.5 Στατιστική Ανάλυση

Για την στατιστική ανάλυση των αποτελεσμάτων, χρησιμοποιήθηκε το Friedman Test για να εξετάσει αν υπάρχουν σημαντικές διαφορές μεταξύ των μεθόδων σε όλα τα datasets. Η ανάλυση πραγματοποιήθηκε ξεχωριστά για τις τιμές $k=1,3,5,7$, τόσο ως προς την Accuracy όσο και προς τον CPU Time.

Το Friedman Test είναι μη παραμετρικό στατιστικό τεστ κατάλληλο για τη σύγκριση πολλών μεθόδων πάνω στα ίδια σύνολα δεδομένων [14]. Στην παρούσα περίπτωση συγκρίθηκαν συνολικά δώδεκα διαφορετικοί συνδυασμοί, οι οποίοι προκύπτουν από τους τρεις αλγορίθμους ταξινόμησης και τις τέσσερις καταστάσεις των δεδομένων: χωρίς μείωση, μετά από CNN, μετά από ENN και μετά από ENN→CNN.

Πίνακας 5.21: Friedman Test ως προς την Accuracy

Μέτρηση	$k = 1$	$k = 3$	$k = 5$	$k = 7$
Friedman statistic	66,150	82,839	76,907	88,188
P-value	< 0.001	< 0.001	< 0.001	< 0.001
Degrees of freedom	11	11	11	11
Αποτέλεσμα	Στατιστικά σημαντικό	Στατιστικά σημαντικό	Στατιστικά σημαντικό	Στατιστικά σημαντικό

Πίνακας 5.22 Κατάταξη μεθόδων ως προς την Accuracy

Μέθοδος	$k = 1$	$k = 3$	$k = 5$	$k = 7$
Κλασσικός k-NN	5,400	4,700	4,900	5,050
Κλασσικός Mutual	5,400	5,575	5,275	4,750
Κλασσικός Symmetric	4,550	3,675	3,300	3,075
CNN + K-NN	9,000	8,925	8,950	9,000
CNN + Mutual	9,000	8,900	8,950	8,750
CNN + Symmetric	4,900	5,125	5,350	4,950
ENN + K-NN	4,900	5,200	5,625	5,825
ENN + Mutual	4,900	5,125	5,350	4,950
ENN + Symmetric	4,550	4,000	4,175	4,525

Μέθοδος	$k = 1$	$k = 3$	$k = 5$	$k = 7$
ENN→ CNN + K-NN	8,450	9,300	8,750	9,125
ENN→ CNN + Mutual	8,450	9,100	8,575	8,675
ENN→ CNN + Symmetric	8,500	8,375	8,800	9,325

Τα αποτελέσματα του Friedman test δείχνουν ότι υπάρχουν στατιστικά σημαντικές διαφορές μεταξύ των μεθόδων για όλες τις τιμές του k και ότι ο Symmetric k -NN χωρίς εφαρμογή τεχνικής μείωσης παρουσιάζει συνολικά την καλύτερη συμπεριφορά ως προς την ακρίβεια ταξινόμησης.

Πίνακας 5.24: Friedman Test ως προς τον CPU Time

Μέτρηση	$k = 1$	$k = 3$	$k = 5$	$k = 7$
Friedman statistic	197,182	167,534	174,575	178,571
P-value	< 0.001	< 0.001	< 0.001	< 0.001
Degrees of freedom	11	11	11	11
Αποτέλεσμα	Στατιστικά σημαντικό	Στατιστικά σημαντικό	Στατιστικά σημαντικό	Στατιστικά σημαντικό

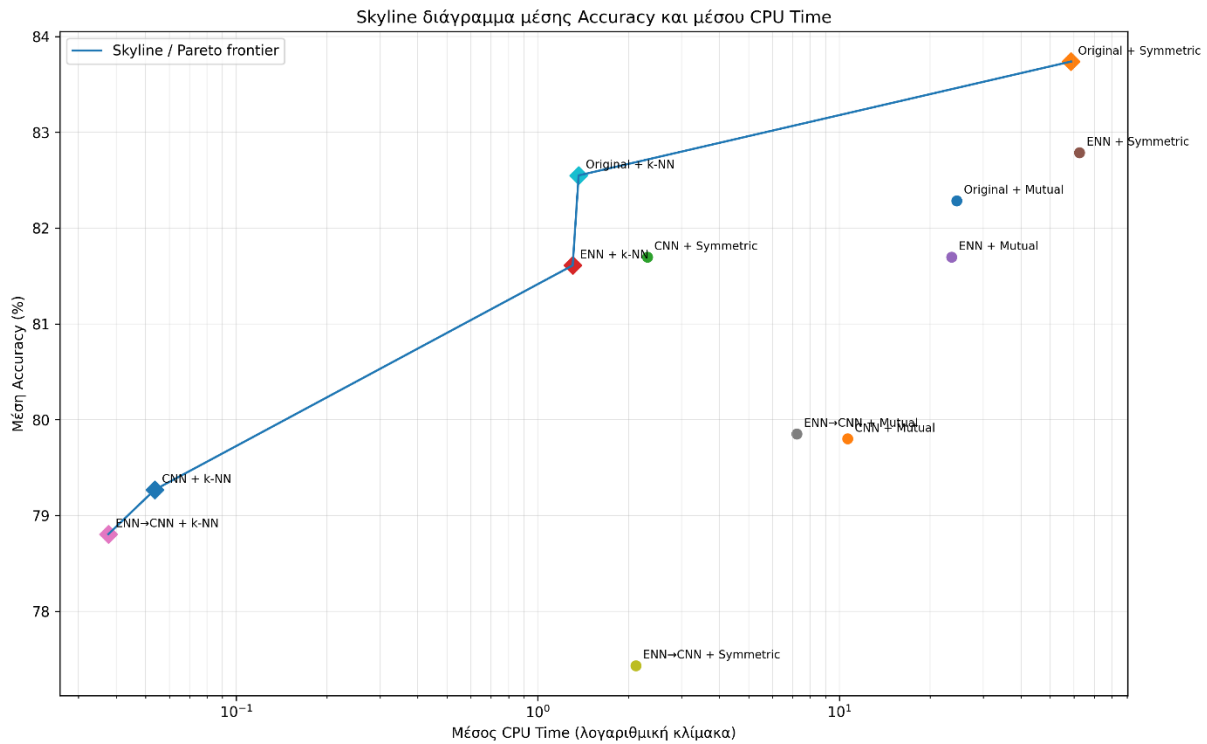
Πίνακας 5.24: Κατάταξη μεθόδων ως προς τον CPU Time

Μέθοδος	$k = 1$	$k = 3$	$k = 5$	$k = 7$
Κλασσικός k -NN	3,600	3,675	3,750	3,600
Κλασσικός Mutual	11,050	9,275	8,700	8,725
Κλασσικός Symmetric	8,450	10,550	10,925	10,975
CNN + K-NN	1,775	2,125	2,025	2,075
CNN + Mutual	10,100	7,800	7,475	7,200
CNN + Symmetric	5,750	7,950	8,175	8,350
ENN + K-NN	3,075	2,675	2,800	2,900
ENN + Mutual	9,750	8,625	8,125	8,250
ENN + Symmetric	7,725	9,825	10,550	10,750
ENN→ CNN + K-NN	1,750	1,725	1,625	1,625
ENN→ CNN + Mutual	9,800	7,000	6,400	6,125
ENN→ CNN + Symmetric	5,175	6,775	7,450	7,425

Αντίστοιχα, για τον CPU Time προκύπτουν στατιστικά σημαντικές διαφορές μεταξύ των μεθόδων για όλες τις τιμές του k . Την καλύτερη μέση κατάταξη παρουσιάζει σταθερά ο ENN→CNN + k -NN, ενώ ακολουθεί ο CNN + k -NN. Τα αποτελέσματα αυτά δείχνουν ότι η μείωση του συνόλου δεδομένων μπορεί να περιορίσει σημαντικά το υπολογιστικό κόστος, ιδιαίτερα όταν συνδυάζεται με τον κλασικό k -NN.

Συνολικά, τα αποτελέσματα παρουσιάζουν μια διαφορετική εικόνα ανάλογα με το κριτήριο αξιολόγησης καθώς ο Symmetric k-NN εμφανίζει την καλύτερη κατάταξη ως προς την Accuracy, αλλά όχι ως προς τον χρόνο εκτέλεσης. Αντίθετα, ο ENN→CNN + k-NN παρουσιάζει την καλύτερη συμπεριφορά ως προς τον CPU Time, χωρίς όμως να βρίσκεται στις πρώτες θέσεις ως προς την Accuracy. Προκύπτει δηλαδή ένας σαφής συμβιβασμός μεταξύ της ποιότητας ταξινόμησης και του υπολογιστικού κόστους, ο οποίος εξετάζεται στη συνέχεια μέσω της ανάλυσης Skyline.

Skyline Analysis



Για τη συνολική αξιολόγηση των μεθόδων χρησιμοποιήθηκε ένα Skyline διάγραμμα [15], στο οποίο εξετάζονται ταυτόχρονα η μέση Accuracy και ο μέσος CPU Time. Στον οριζόντιο άξονα παρουσιάζεται ο μέσος χρόνος εκτέλεσης, ενώ στον κατακόρυφο άξονα η μέση ακρίβεια ταξινόμησης. Επομένως, επιθυμητές είναι οι μέθοδοι που συνδυάζουν υψηλή Accuracy με χαμηλό CPU Time. Προκύπτει πως οι μέθοδοι ENN→CNN + k-NN, CNN + k-NN, ENN + k-NN, Original k-NN και Original Symmetric k-NN είναι αυτές που δεν υπολείπονται ταυτόχρονα και στα δύο κριτήρια από κάποια άλλη μέθοδο και παρουσιάζουν διαφορετικούς συμβιβασμούς μεταξύ ακρίβειας και υπολογιστικού κόστους. Ο Original Symmetric k-NN παρουσιάζει τη μεγαλύτερη μέση Accuracy (83,74%), αλλά απαιτεί σημαντικά μεγαλύτερο CPU Time. Αντίθετα, ο ENN→CNN + k-NN παρουσιάζει τον μικρότερο μέσο χρόνο εκτέλεσης, αλλά χαμηλότερη Accuracy (78,81%). Ανάμεσα στα δύο άκρα, οι CNN + k-NN, ENN + k-

NN και Original k-NN αποτελούν ενδιαμέσες επιλογές. Ιδιαίτερα ο Original k-NN επιτυγχάνει υψηλή μέση Accuracy (82,55%) με αρκετά μικρότερο υπολογιστικό κόστος σε σχέση με τον Original Symmetric k-NN. Προκύπτει δηλαδή ότι δεν υπάρχει μία μέθοδος που να είναι καλύτερη σε όλα τα κριτήρια. Η επιλογή της κατάλληλης μεθόδου εξαρτάται από το αν δίνεται μεγαλύτερη έμφαση στην ακρίβεια ταξινόμησης ή στη μείωση του υπολογιστικού κόστους.

Κεφάλαιο 6ο: Συμπεράσματα και Μελλοντικές επεκτάσεις

6.1 Συμπεράσματα

Στην παρούσα εργασία έγινε μελέτη της χρήσης τεχνικών μείωσης δεδομένων στη λειτουργία και την απόδοση αλγορίθμων ταξινόμησης που βασίζονται στους πλησιέστερους γειτόνους. Εξετάστηκαν ο κλασσικός k-NN καθώς και οι παραλλαγές του Mutual k-NN και Symmetric k-NN, σε συνδυασμό με τις τεχνικές μείωσης δεδομένων Condensed Nearest Neighbor (CNN) και Edited Nearest Neighbor (ENN). Επιπλέον, εξετάστηκε και η διαδοχική εφαρμογή των δύο τεχνικών, με αρχική εφαρμογή του ENN και στη συνέχεια του CNN.

Το βασικό κίνητρο της εργασίας προέκυψε από το γεγονός ότι οι τεχνικές μείωσης δεδομένων έχουν χρησιμοποιηθεί κυρίως σε συνδυασμό με τον κλασσικό k-NN αλλά όχι σε παραλλαγές του. Για το λόγο αυτό επάγεται το ενδιαφέρον της πειραματικής αξιολόγησης των CNN και ENN όχι μόνο στον κλασσικό k-NN, αλλά και στους Mutual k-NN και Symmetric k-NN.

Για τη πειραματική μελέτη χρησιμοποιήθηκαν είκοσι διαφορετικά σύνολα δεδομένων και τιμές $k = 1, 3, 5, 7$ τόσο χωρίς μείωση δεδομένων όσο και μετά την εφαρμογή των CNN, ENN και του συνδυασμού τους ENN σε CNN. Με τον τρόπο αυτό έγινε σύγκριση ανάμεσα στην απόδοση των αλγορίθμων ως προς την ακρίβεια ταξινόμησης και το υπολογιστικό κόστος.

Τα αποτελέσματα έδειξαν ότι οι τεχνικές μείωσης δεδομένων μπορούν να μειώσουν σημαντικά το μέγεθος του συνόλου εκπαίδευσης και τον χρόνο εκτέλεσης με την μεγαλύτερη μείωση να παρατηρείται κυρίως στο CNN και στον συνδυασμό ENN σε CNN. Ωστόσο, η μεγαλύτερη μείωση των δεδομένων δεν οδηγεί πάντα σε καλύτερη ακρίβεια. Το ENN διατήρησε γενικά την Accuracy πιο κοντά στα αρχικά δεδομένα, ενώ ο CNN και ιδιαίτερα ο ENN σε CNN παρουσίασαν σε κάποιες περιπτώσεις μεγαλύτερη μείωση της ακρίβειας.

Ακόμη, προέκυψε ότι οι τεχνικές μείωσης δεδομένων μπορούν να χρησιμοποιηθούν και με τους Mutual k-NN και Symmetric k-NN, με τα αποτελέσματα να διαφέρουν ανάλογα με το σύνολο δεδομένων και την τιμή του k . Συνολικά, οι τεχνικές μείωσης δεδομένων μπορούν να περιορίσουν το υπολογιστικό κόστος, όμως η επιλογή τους πρέπει να λαμβάνει υπόψη και την πιθανή επίδραση στην ακρίβεια ταξινόμησης.

6.2 Μελλοντικές επεκτάσεις

Τα αποτελέσματα της παρούσας εργασίας δείχνουν ότι η εφαρμογή τεχνικών μείωσης δεδομένων δεν περιορίζεται αποκλειστικά στον κλασσικό k-NN, αλλά μπορεί να παρουσιάζει ενδιαφέρον και σε διαφορετικές παραλλαγές του αλγορίθμου. Το γεγονός αυτό δημιουργεί τη δυνατότητα περαιτέρω διερεύνησης της χρήσης τους σε άλλους αλγορίθμους πλησιέστερων γειτόνων που δεν εξετάστηκαν στο πλαίσιο της παρούσας εργασίας.

Μια πιθανή μελλοντική επέκταση θα μπορούσε επομένως να αφορά την εφαρμογή των CNN και ENN σε επιπλέον παραλλαγές του k-NN και τη σύγκριση της συμπεριφοράς τους με τα αποτελέσματα των k-NN, Mutual k-NN και Symmetric k-NN. Με αυτόν τον τρόπο θα μπορούσε να διερευνηθεί κατά πόσο τα οφέλη που παρατηρήθηκαν ως προς τη μείωση του υπολογιστικού κόστους μπορούν να εμφανιστούν και σε άλλες μεθόδους ταξινόμησης που βασίζονται στην έννοια της γειτνίασης.

Επιπλέον, θα μπορούσαν να εξεταστούν και άλλες τεχνικές μείωσης δεδομένων, πέρα από τις CNN, ENN και τον συνδυασμό τους που χρησιμοποιήθηκαν στην παρούσα εργασία. Η σύγκριση περισσότερων τεχνικών θα μας έδινε τη δυνατότητα για μία πληρέστερη αξιολόγηση της σχέσης μεταξύ του ποσοστού μείωσης, της ακρίβειας ταξινόμησης και του απαιτούμενου υπολογιστικού χρόνου.

Συνεπώς, τα αποτελέσματα της εργασίας μπορούν να αποτελέσουν βάση για περαιτέρω μελέτη των τεχνικών μείωσης δεδομένων και της εφαρμογής τους σε ένα ευρύτερο σύνολο αλγορίθμων ταξινόμησης, με στόχο την ανάπτυξη μεθόδων που συνδυάζουν αποτελεσματική ταξινόμηση με χαμηλότερες υπολογιστικές απαιτήσεις.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.
- [2] E. Fix and J. L. Hodges, "Discriminatory analysis. Nonparametric discrimination: Consistency properties," USAF School of Aviation Medicine, Randolph Field, Texas, Project 21-49-004, Rep. 4, 1951.
- [3] P. E. Hart, "The condensed nearest neighbor rule," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 515–516, May 1968.
- [4] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-2, no. 3, pp. 408–421, Jul. 1972.
- [5] K. C. Gowda and G. Krishna, "The condensed nearest neighbor rule using the concept of mutual nearest neighborhood," *IEEE Transactions on Information Theory*, vol. 25, no. 4, pp. 488–490, Jul. 1979.
- [6] H.-C. Kim, "Bayesian model selection methods for mutual and symmetric k-nearest neighbor classification," *arXiv preprint arXiv:1608.04063*, 2016.
- [7] S. Ougiaroglou and D. A. Dervos, "Adaptive k-nearest-neighbor classification using a dynamic number of nearest neighbors," in *Proc. 7th Balkan Conference on Informatics (BCI '15)*, Craiova, Romania, 2015, pp. 1–8.
- [8] K. Batsios, S. Ougiaroglou, and C. Bratsas, "AutoKNN: An AutoML web application for k-nearest neighbours classification," in *Proc. 2025 16th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pp. 1–5, 2025.
- [9] S. Ougiaroglou and D. A. Dervos, "Exploring the effect of data reduction on neural network classifiers," *Information*, vol. 12, no. 8, p. 308, Aug. 2021.
- [10] S. García, J. Derrac, J. R. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: Taxonomy and empirical study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 417–435, Mar. 2012.
- [11] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.
- [12] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
- [13] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation," in *Proc. 14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, pp. 1137–1143, 1995.
- [14] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, Jan. 2006.
- [15] S. Börzsönyi, D. Kossmann, and K. Stocker, "The Skyline operator," in *Proc. 17th International Conference on Data Engineering (ICDE '01)*, Heidelberg, Germany, 2001, pp. 421–430.

- [16] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL data-set repository: An open source software tool to assess evolutionary algorithms for data mining: Problems and algorithms," *Soft Computing*, vol. 15, no. 2, pp. 255–287, Feb. 2011.
- [17] D. Dua and C. Graff, "UCI Machine Learning Repository," School of Information and Computer Sciences, University of California, Irvine, CA, 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.