Βαθιά Μάθηση - Εξαμηνιαία Εργασία

Δημήτριος Ζερκελίδης Μαρία Καϊκτζόγλου Μαρία-Φιλίππα Τριβυζά

Τμήμα Ηλεκτρολόγων Μηχανολόγων και Μηχανικών Υπολογιστών,
Εθνικό Μετσόβιο Πολυτεχνείο,
Ζωγράφου 15780, Έλλαδα

Περίληψη

Στη συγκεκριμένη ερευνητική εργασία παρουσιάζεται μια αναπαραγωγή αποτελεσμάτων ενός state-of-the-art μοντέλου που αφορά τη συμπλήρωση ελλιπών τιμών, το GAIN. Το GAIN χρησιμοποιεί μια παραλλαγή της αρχιτεκτονικής GAN, στην οποία ο generator (G) συμπληρώνει τις ελλιπείς τιμές, ενώ ο discriminator (D) έχει ως είσοδο το συμπληρωμένο διάνυσμα και βρίσκει ποια από αυτά τα στοιχεία συμπληρώθηκαν ή παρατηρήθηκαν. Για να διασφαλιστεί πως το μοντέλο εκπαιδεύεται, χρησιμοποιείται ένα hint vector που καταδεικνύει μερική πληροφορία ως προς την αυθεντικότητα των χαρακτηριστικών ενός δείγματος. Επιπρόσθετα, τροποποιούμε το μοντέλο αυτό βελτιώνοντας τα αποτελέσματα. Παράλληλα, κάνουμε επέκταση του GAIN σε δεδομένα εικόνων μελετώντας την επίδοση του με διάφορες τροποποιήσεις και με χρήση συνελικτικών επιπέδων.

1 Εισαγωγή

Η παρουσία ελλιπών τιμών (missing values) είναι ένα από τα συχνότερα προβλήματα που παρουσιάζουν τα πραγματικά σύνολα δεδομένων (datasets), και μπορεί να επηρεάσει αρνητικά τα μοντέλα μηχανικής μάθησης [1]. Τα δεδομένα μπορεί να λείπουν για διάφορους λόγους, όπως για παράδειγμα να μη συλλέχθηκαν ποτέ, και κατηγοριοποιούνται με τρεις διαφορετικούς τρόπους: MAR (Missing At Random), NMAR (Not Missing At Random) και MCAR (Missing Completely at Random). Τα MAR δεδομένα είναι εκείνα τα οποία λείπουν τυχαία, αλλά η έλλειψή τους εξαρτάται από τις παρατηρούμενες μεταβλητές του συνόλου δεδομένων. Τα NMAR δεδομένα είναι εκείνα των οποίων η έλλειψη εξαρτάται και από τις παρατηρούμενες και από τις μη παρατηρούμενες μεταβλητές. Τέλος, τα MCAR δεδομένα είναι εκείνα τα οποία λείπουν εντελώς τυχαία, δηλαδή η έλλειψή τους δεν εξαρτάται από τις παρατηρούμενες μεταβλητές, και αποτελούν την κατηγορία ελλιπών δεδομένων με την οποία θα ασχοληθούμε στην παρούσα εργασία.

Η Pigott [2] μας δίνει κάποια παραδείγματα για την κατανόηση των τριών κατηγοριών σε μια περίπτωση μελέτης του άσθματος στην οποία συμμετείχαν μαθητές και μαθήτριες ηλικίας 8-14 ετών. Εξηγεί πως μια αιτία για τα ελλιπή δεδομένα ήταν οτι κάποια παιδιά απλώς ξέχασαν να επισκεφθούν τη σχολική κλινική για να συμπληρώσουν το μέρος του ερωτηματολογίου που αφορά την ένταση των συμπτωμάτων τους. Κάτι τέτοιο δε σχετίζεται με την κατάσταση της υγείας τους και οφείλεται αποκλειστικά σε τυχαίο παράγοντα, επομένως αυτή είναι μια MCAR περίπτωση. Μια άλλη αιτία ήταν το ότι τα παιδιά με πιο σοβαρή μορφή άσθματος έμειναν περισσότερες ημέρες στο σπίτι και ως αποτέλεσμα δεν είχαν τη δυνατότητα να συμπληρώσουν το ερωτηματολόγιο. Σε αυτήν την περίπτωση η αιτία των ελλιπών δεδομένων σχετίζεται με την ίδια τη μεταβλητή και είναι μια περίπτωση NMAR. Μια τρίτη αιτία που υπάγεται στη MAR κατηγορία είναι ότι τα παιδιά σε μικρότερες ηλικίες δεν είχαν την ευχέρεια να ερμηνεύσουν το ερωτηματολόγιο στο δοθέντα χρόνο, αφήνοντας ασυμπλήρωτα κάποια σημεία. Επομένως, η αιτία δεν έχει να κάνει με τη σοβαρότητα της ασθένειας αλλά με την ηλικία, μια άλλη, παρατηρούμενη μεταβλητή του συνόλου δεδομένων.

Η ανάγκη διαχείρισης του προβλήματος αυτού οδήγησε στην πρόταση και υλοποίηση διαφόρων τεχνικών

συμπλήρωσης των ελλιπών τιμών (imputation[1] methods), έτσι ώστε να δημιουργηθεί ένας πλήρης πίνακας δεδομένων. Μία σχετικά νέα αλλά πολλά υποσχόμενη imputation τεχνική βασίζεται στα Generative Adversarial Networks (GAN) [3]. Τα GAN εισήχθησαν το 2014 ως νέες αλγοριθμικές αρχιτεκτονικές που χρησιμοποιούν δύο νευρωνικά δίκτυα, το ένα εναντίον του άλλου, για τη δημιουργία νέων, συνθετικών δειγμάτων δεδομένων που μπορούν να «περάσουν» για πραγματικά δεδομένα. Αν και η αρχική δομή τους δεν είναι κατάλληλη για τη συμπλήρωση ελλιπών δεδομένων, η τεράστια αποδοχή που γνώρισαν από την κοινότητα της βαθιάς μάθησης οδήγησε τελικά στη διεξαγωγή σχετικής έρευνας για την τροποποίηση των GAN προς αυτή την κατεύθυνση, όπως το GAIN.

Επικεντρωθήκαμε στο μοντέλο GAIN για να περιγράψουμε θεωρητικά τον τρόπο λειτουργίας του. Δοκιμάσαμε ορισμένες αλλαγές παρεμβαίνοντας στη δομή του δικτύου, όπως στο βάθος του νευρωνικού δικτύου, στο είδος των κρυφών επιπέδων (χρήση επιπέδων συνέλιξης), αλλά και στην αλλαγή των συναρτήσεων ενεργοποίησης. Πέραν αυτού, μελετήσαμε τις παραπάνω μεθοδολογίες στα σύνολα δεδομένων Spam [4], Letter [5], Credit [6], News [7], Breast [8] όπως και οι Yoon et al. [1], ενώ ταυτόχρονα πειραματιστήκαμε και σε δεδομένα εικόνων όπως το MNIST [9] και το Chest x-ray (Pneumonia) [10]. Οι αλλαγές που υλοποιήσαμε αναλύονται στην ενότητα 5.

2 Σχετική Έρευνα

Στο πεδίο έρευνας των μεθόδων imputation, οι αλγόριθμοι που χρησιμοποιούνται θα μπορούσαν να διαχωριστούν στις κατηγορίες: 1) discriminative και 2) generative imputation μοντέλα [11]. Ένα discriminative μοντέλο εκπαιδεύεται ώστε να βρεθεί το σημείο απόφασης, ενώ ένα generative μοντέλο μοντελοποιεί την κατανομή της κάθε κλάσης. Στην κατηγορία των discriminative μοντέλων ανήκουν αλγόριθμοι όπως ο KNNimpute [12], MissForest [13], Mice [14] και Matrix Completion [15]. Ενώ στη δεύτερη κατηγορία ανήκουν αλγόριθμοι όπως ο Expectation-Maximization (EM) [16], Denoising Auto-Encoders (DAE) [17] και GAN [3], το οποίο είναι η βάση του μοντέλου που παρουσιάζουμε.

Ειδοποιοί διαφορές μεταξύ των μεθόδων αφορούν τους περιορισμούς του κάθε αλγορίθμου. Τέτοιοι περιορισμοί μπορεί να σχετίζονται είτε με υποθέσεις για τη φύση των ελλιπών δεδομένων, όπως αν είναι MAR, MCAR ή MNAR, είτε με το αν ειναι γνωστή η κατανομή από όπου προέρχονται τα δείγματα μας, είτε με το ειδός των μεταβλητών, δηλαδή αν πρόκειται για μίξη κατηγορικών και αριθμητικών τιμών. Παρακάτω παρουσιάζουμε τη γενική ιδέα πίσω από κάποια μοντέλα και ορισμένα αποτελέσματα πειραμάτων που έχουν γίνει με αυτά.

Στις discriminative imputation μεθόδους, η μέθοδος Mice υλοποιεί μια τεχνική που βασίζεται στην ιδέα ότι κάθε μεταβλητή η οποία περιέχει ελλιπή δεδομένα αντιμετωπίζεται σαν εξαρτώμενη μεταβλητή σε παλινδρόμηση και οι υπόλοιπες μεταβλητές λειτουργούν ως οι ανεξάρτητες [14]. Ο Mice αλγόριθμος λειτουργεί αρκετά καλά σε MAR δεδομένα, ενώ σε άλλη περίπτωση δεν μπορεί να εφαρμοστεί στα εκλιπόντα δεδομένα, καθιστώντας χρονοβόρο τον υπολογισμό του σφάλματος. Ο αλγόριθμος KNNimpute [12] αντικαθιστά τα δεδομένα που λείπουν χρησιμοποιώντας τις αντίστοιχες τιμές από τους κοντινότερους γείτονες. Αν η αντίστοιχη τιμή λείπει επίσης από κάποιον γείτονα, τότε αυτός ο γείτονας αντικαθίσταται με τον επόμενο κοντινότερο. Όταν εφαρμόστηκε και συγκρίθηκε με άλλους αλγόριθμους σε δεδομένα DNA microarrays, έδειξε ότι δεν είναι ανθεκτικός στην αύξηση ποσοστού των missing δεδομένων και το θόρυβο, ωστόσο είναι ταχύς [12]. Από την άλλη, ο MissForest [13] ένας πιο αργός αλγόριθμος αλλά με καλύτερα αποτελέσματα ανεξαρτήτως κατηγορίας δεδομένων (κατηγορικά-αριθμητικά), χωρίς να κάνει υποθέσεις για τις κατανομές των δειγμάτων. Κάνει χρήση των Random Forests για την ανάκτηση των ελλιπών δεδομένων και έχει το πλεονέκτημα ότι δεν έχει υπερπαραμέτρους για ρύθμιση, ενώ λειτουργεί καλά ακόμα κι όταν το πλήθος των διαστάσεων είναι μεγαλύτερο από τον αριθμό δειγμάτων.

Τα generative μοντέλα περιλαμβάνουν αλγορίθμους που βασίζονται στη στατιστική μέθοδο EM [16] για την εκτίμηση παραμέτρων με την παρουσία κρυφών μεταβλητών, και αλγορίθμους που βασίζονται στη βαθιά μάθηση (π.χ. DAE και GAN) [1]. Τα generative μοντέλα που έχουν προταθεί αντιμετωπίζουν διάφορα προβλήματα. Για παράδειγμα, ο EM αλγόριθμος απαιτεί τη γνώση της κατανομής των δεδομένων, ενώ αποτυγχάνει στη γενίκευση όταν υπάρχουν δείγματα με διαφορετικά είδη μεταβλητών ταυτόχρονα, π.χ. και αριθμητικά και κατηγορικά. Τα DAE [17] δεν υποθέτουν κατανομή, ωστόσο απαιτούν ολοκληρωμένο σύνολο δεδομένων. Επομένως, τα DAE δεν είναι πρακτικά, διότι ουσιαστικά κυριαρχούν σύνολα με ελλιπή δεδομένα.

Βασισμένες στην αρχιτεκτονική των GAN [3], υπάρχουν και άλλες τροποποιήσεις για την ανάκτηση ελλιπών δεδομένων. Μια από αυτές είναι το MisGan [18]. Έχει παρατηρηθεί πως το MisGan δίνει καλά αποτελέσματα

[1]Χρησιμοποιούμε τον όρο "imputation", διότι δεν υπάρχει ακριβής αντιστοιχία στα ελληνικά.

τόσο για MCAR δεδομένα όσο και για MAR και NMAR. Το δυνατό σημείο της συγκεκριμένης αρχιτεκτονικής βασίζεται στην ευστάθεια της εκπαίδευσης, σε αντίθεση με το GAIN, το οποία μετά από ένα συγκεκριμένο αριθμό εποχών, δε λειτουργεί εύρυθμα και δεν μπορεί να ανακτήσει τα ελλιπή δεδομένα [18]. Άλλες μέθοδοι βασισμένες στην αρχιτεκτονική GAN είναι τα VIGAN [19] (p.9) και τα CollaGAN [20] (p.9) που δίνουν υψηλής ποιότητας εικόνες, αλλά αποτελούνται από μια αρχιτεκτονική που αυξάνει την πολυπλοκότητα στην εκπαίδευση.

Το GAIN [1] επιτυγχάνει χωρίς ολοκληρωμένο dataset την ανάκτηση ελλιπών δεδομένων, αποφεύγοντας υποθέσεις για τις κατανομές και με αρκετά καλό ποσοστό ακρίβειας στις MAR περιπτώσεις. Διαθέτει απλή λειτουργία, καθώς γίνεται χρήση ενός απλού GAN [3] και τροποποιώντας τα δεδομένα εισόδου, ξεπερνάει προηγούμενες state-of-the-art μεθόδους, π.χ. KNNimpute [12], MC [15], MissForest [13], Mice [14], DAE [17] και EM [16].

3 Σύνολα Δεδομένων και Χαρακτηριστικά

Χρησιμοποιήσαμε τα σύνολα δεδομένων breast [8], spam [4], letter [5], credit [6] και news [7], τα οποία χρησιμοποιήθηκαν στο paper των Yoon et. al. Τα διαμορφώσαμε εφαρμόζοντας min-max κανονικοποίηση και αφαιρώντας τις ετικέτες τους. Επίσης, χρησιμοποιήσαμε τα σύνολα δεδομένων Chest X-Ray Images (Pneumonia) [10] και MNIST [9] για περισσότερο πειραματισμό στο imputation. Ακολουθεί συνοπτική περιγραφή.

3.1 Breast Cancer Wisconsin (Diagnostic) Data Set

Τα χαρακτηριστικά του Breast Cancer Wisconsin (Diagnostic) dataset υπολογίζονται από μια ψηφιοποιημένη εικόνα αναρρόφησης με λεπτή βελόνα (fine needle aspirate - FNA) μάζας μαστού. Συγκεκριμένα, υπολογίζονται δέκα real-valued χαρακτηριστικά για κάθε πυρήνα κυττάρων που υπάρχει στην εικόνα. Τα χαρακτηριστικά των δεδομένων είναι αριθμητικές και κατηγορικές μεταβλητές και η μεταβλητή απόκρισης είναι σε δυαδική μορφή, υποδηλώνοντας τη διάγνωση (M = κακοήθη, B = καλοήθη).

3.2 Spambase Data Set (Βάση δεδομένων SPAM E-mail)

Με τον όρο spam συνήθως αναφερόμαστε στη μαζική αποστολή ηλεκτρονικών μηνυμάτων με εξωτερικούς συνδέσμους, διαφημίσεις και άλλα, σε μια προσπάθεια προώθησης προϊόντων ή ιδεών. Το Spambase dataset [4] αποτελείται από μια συλλογή ανεπιθύμητων μηνυμάτων ηλεκτρονικού ταχυδρομείου (spam e-mails), και μια συλλογή μη-ανεπιθύμητων μηνυμάτων (non-spam e-mails), τα οποία προήλθαν, κυρίως, από αρχειοθετημένα προσωπικά μηνύματα ηλεκτρονικού ταχυδρομείου των δημιουργών του dataset. Ως εκ τούτου η λέξη «george» και ο κωδικός περιοχής «650», που συνδέονται με έναν χρήστη και δημιουργό του dataset, αποτελούν δείκτες μη-ανεπιθύμητων μηνυμάτων. Τα 57 χαρακτηριστικά των δεδομένων είναι πραγματικοί, ακέραιοι αριθμοί. Τα περισσότερα από τα χαρακτηριστικά αυτά υποδεικνύουν εάν μια συγκεκριμένη λέξη ή χαρακτήρας εμφανιζόταν συχνά στο e-mail. Η μεταβλητή απόκρισης είναι σε δυαδική μορφή και υποδηλώνει αν το mail είναι spam(1) ή όχι (0).

3.3 Letter Recognition Data Set

Ο στόχος του Letter Recognition dataset [5] είναι να προσδιοριστεί κάθε μία από τις ασπρόμαυρες pixelated εικόνες ως ένα από τα 26 κεφαλαία γράμματα στο αγγλικό αλφάβητο. Οι εικόνες χαρακτήρων βασίστηκαν σε 20 διαφορετικές γραμματοσειρές και κάθε γράμμα σε αυτές τις 20 γραμματοσειρές παραμορφώθηκε τυχαία για να παράγει ένα αρχείο 20.000 μοναδικών ερεθισμάτων. Κάθε ερέθισμα μετατράπηκε σε 16 αριθμητικά χαρακτηριστικά (στατιστικές ροπές και μετρήσεις ακμών) τα οποία έγιναν scaled έτσι ώστε να χωρέσουν σε ένα εύρος ακέραιων τιμών από 0 έως 15. Σημειώνεται πως ένα εικονοστοιχείο (pixel) καλείται "on" εάν περιέχει θετική αριθμητική τιμή, αλλιώς καλείται "off". Τα χαρακτηριστικά αποτελούνται από αριθμητικές μεταβλητές και από μια κατηγορική μεταβλητή (γράμμα).

3.4 Default of credit card clients Data Set

Το default of credit card clients dataset [6] στοχεύει στον προσδιορισμό αθετημένων πληρωμών (default payments) πελατών στην Ταϊβάν. Ως μεταβλητή απόκρισης χρησιμοποιείται μια δυαδική μεταβλητή, η αθετημένη

πληρωμή (Ναι = 1, Όχι = 0). Από τα 23 χαρακτηριστικά, η πλειοψηφία είναι κατηγοριματικές μεταβλητές, ενώ υπάρχουν και κάποια αριθμητικά που περιγράφουν τα ποσά στο Bill Statement για συγκεκριμένους μήνες.

3.5 Online News Popularity Data Set

Το Online News Popularity dataset [7] συνοψίζει ένα ετερογενές σύνολο χαρακτηριστικών σχετικά με άρθρα που δημοσιεύθηκαν από το Mashable σε περίοδο δύο ετών. Ο στόχος είναι να προβλεφθεί ο αριθμός των shares στα κοινωνικά δίκτυα (δημοτικότητα). Το dataset αποτελείται από 61 χαρακτηριστικά (58 χαρακτηριστικά πρόβλεψης (predictive attributes), 2 μη προβλέψιμα (non-predictive), 1 πεδίο στόχου (goal field)), τα οποία είναι αριθμητικές και κατηγορικές μεταβλητές.

3.6 MNIST Data Set (Modified National Institute of Standards and Technology database)

Η βάση δεδομένων MNIST [9] αποτελείται από χειρόγραφα ψηφία και διαθέτει ένα σύνολο δεδομένων εκπαίδευσης (training set) 60.000 παραδειγμάτων και ένα σύνολο δεδομένων δοκιμής (test set) 10.000 παραδειγμάτων. Είναι ένα υποσύνολο ενός μεγαλύτερου dataset, το οποίο διατίθεται από το NIST. Τα ψηφία έχουν κανονικοποιηθεί ως προς το μέγεθος και έχουν κεντραριστεί σε μια εικόνα σταθερού μεγέθους 28x28 pixels. Αποτελεί μια καλή βάση δεδομένων για άτομα που θέλουν να δοκιμάσουν τεχνικές μάθησης και μεθόδους αναγνώρισης προτύπων σε πραγματικά δεδομένα, ενώ χρειάζεται να καταβληθούν ελάχιστες προσπάθειες για την προεπεξεργασία και τη μορφοποίησή της.

3.7 Chest X-Ray Images (Pneumonia) Data Set

Το Chest X-Ray Images (Pneumonia) Data Set [10] είναι οργανωμένο σε 3 φακέλους (train, test, val) και περιέχει υποφακέλους για κάθε κατηγορία εικόνας (Pneumonia/Normal). Υπάρχουν 5.863 εικόνες ακτίνων X (JPEG) και 2 κατηγορίες (Pneumonia/Normal). Στα δεδομένα πραγματοποιήσαμε μετατροπή σε gray scale και resize σε 128×128 με τη χρήση της βιβλιοθήκης openCV [21]. Οι εικόνες ακτινογραφίας θώρακα (πρόσθιο-οπίσθιο μέρος) επιλέχθηκαν από μελέτες κοόρτης (retrospective cohorts) παιδιατρικών ασθενών ηλικίας ενός έως πέντε ετών από το Guangzhou Women and Children's Medical Center. Όλη η απεικόνιση ακτινογραφίας θώρακα πραγματοποιήθηκε ως μέρος της ρουτίνας κλινικής φροντίδας των ασθενών.

Για την ανάλυση των X-Ray εικόνων θώρακα, όλες οι ακτινογραφίες θώρακα ελέγχθηκαν αρχικά για έλεγχο ποιότητας αφαιρώντας όλες τις χαμηλής ποιότητας ή δυσανάγνωστες σαρώσεις (scans). Στη συνέχεια, οι διαγνώσεις για τις εικόνες «βαθμολογήθηκαν» (graded) από δύο ειδικούς γιατρούς. Προκειμένου να ληφθούν υπόψη τυχόν σφάλματα «βαθμολόγησης», το σύνολο αξιολόγησης ελέγχθηκε επίσης από έναν τρίτο εμπειρογνώμονα.

4 Θεωρητικό Πλαίσιο

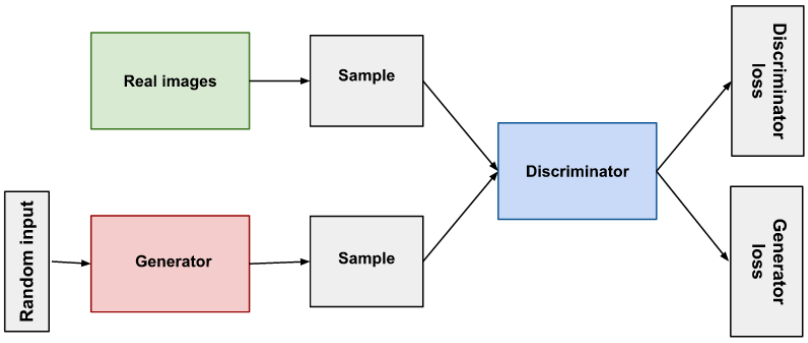
4.1 Generative Adversarial Networks (GAN)

Για να μπορέσουμε να αναλύσουμε τη μέθοδο GAIN [1] θα πρέπει πρώτα να περιγράψουμε την αρχιτεκτονική των GAN [3], τα οποία εκπαιδεύουν δύο μοντέλα νευρωνικών δικτύων ταυτόχρονα μέσω μιας ανταγωνιστικής διαδικασίας. Συγκεκριμένα, αξιοποιούν ένα generative μοντέλο και ένα discriminative μοντέλο τα οποία ανταγωνίζονται μεταξύ τους και ο ανταγωνισμός αυτός συντελεί στη βελτίωση και των δύο. Όπως προαναφέρθηκε, ο όρος "generative" περιγράφει μια κλάση στατιστικών μοντέλων τα οποία είναι σε θέση να γεννήσουν νέα δεδομένα μαθαίνοντας την κατανομή ενός δείγματος (π.χ. του συνόλου εκπαίδευσης). Αντιθέτως, τα discriminative μοντέλα αναλαμβάνουν να διαχωρίσουν τα δεδομένα σε κλάσεις. Με άλλα λόγια, το generative μοντέλο βρίσκει την (από κοινού) κατανομή $P(Q, Y)$ (ή $P(Q)$), όπου Q ειναι τα δεδομένα του δείγματος και ϒ οι ετικέτες τους, ενώ τα discriminative μοντέλα βρίσκουν την κατανομή $P(Y|X)$.

Οι Goodfellow et. al. [3] περιγράφοντας το GAN δίκτυο γράφουν πως το δίκτυο του generator παράγει δείγματα $x = g(z; \theta(g))$, όπου g είναι η συνάρτηση που προσομοιώνει τα πραγματικά δεδομένα με τη βοήθεια θορύβου z και παραμέτρων $\theta(g)$. Ο αντίπαλός του, ο discriminator, δέχεται ως δεδομένα εισόδου τόσο τα αυθεντικά δείγματα όσο και τα ψευδο-δείγματα που παράγονται από το generator, και αποφασίζει κάθε φορά για το αν το δείγμα είναι γνήσιο ή έχει παραχθεί από τον generator. Με αυτόν τον τρόπο, η εκπαίδευση του generator είναι μια διαδικασία στην οποία ο generator προσπαθεί να μάθει όσο καλύτερα γίνεται την

κατανομή των δειγμάτων έτσι ώστε να κάνει το discriminator να σφάλλει, και ο discriminator προσπαθεί να μεγιστοποιήσει την πιθανότητα να αποδόσει τη σωστή ετικέτα στο εισερχόμενο δείγμα (γνήσιο ή παραγόμενο).

Στην Εικόνα 3 φαίνεται η αρχιτεκτονική του GAN, όπου το input θεωρείται να είναι εικόνες. Ο generator δέχεται μια τυχαία είσοδο και με βάση αυτό παράγει ένα ψευτο-δείγμα. Η είσοδος μπορεί να ακολουθεί οποιαδήποτε κατανομή, π.χ. την κανονική, ή οποιαδήποτε άλλη. Ταυτόχρονα, επιλέγεται ένα πραγματικό δείγμα, και μαζί με το ψευτο-δείγμα του generator δίνεται στο discriminator. Αυτός με τη σειρά του θα ελέγξει τη εγγύτητα των δύο εικόνων-δειγμάτων και θα αποφασίσει ότι το ψευτο-δείγμα είναι πραγματικό αν υπολογίσει μεγάλη εγγύτητα. Σε αυτήν την περίπτωση ο generator θα εχει πετύχει σε μεγάλο βαθμό την προσομοίωση της πραγματικής κατανομής. Αν ο discriminator όμως διακρίνει διαφορές, και άρα μικρή εγγύητητα στα δείγματα, τότε θα αποφασίσει ορθά ότι το παραγόμενο δείγμα είναι ψεύτικο, αναγκάζοντας τον generator να ανανεώσει το δίκτυο του - με backpropagation - για να προσεγγίσει περισσότερο την πραγματική κατανομή των δειγμάτων-εικόνων.



Εικόνα 1: GAN αρχιτεκτονική.

4.2 Generative Adversarial Imputation Networks (GAIN)

Τα GAIN [1] δίκτυα ακολουθούν, όπως αναφέραμε, την ίδια περίπου διαδικασία εκπαίδευσης, αλλά αναλαμβάνουν επιπλέον να ανακτήσουν τα δεδομένα που λείπουν από το αρχικό δείγμα (imputation). Στο GAIN υπάρχει ομοίως ο generator (G) που παρατηρεί κάποια στοιχεία από ένα πραγματικό διάνυσμα δεδομένων, παράγει (imputes) τα στοιχεία που λείπουν με βάση τα στοιχεία που έχει παρατηρήσει και εξάγει ως output ένα πλήρες διάνυσμα δεδομένων. Ο discriminator (D) στη συνέχεια παίρνει το το πλήρες διάνυσμα που έχει εξάγει ο G και αποφασίζει ποια στοιχεία υπήρχαν εξαρχής και ποια παρήχθησαν με imputation. Είναι προφανής η σημασία του να διασφαλιστεί ότι ο D αναγκάζει τον G να μαθαίνει την πραγματική κατανομή. Για το σκοπό αυτό δίνεται στον D επιπλέον ως input ένα διάνυσμα που ονομάζεται hint vector, το οποίο περιέχει μερική πληροφορία σχετικά με τα ελλιπή στοιχεία του διανύσματος εισόδου. Έτσι ο D έχει μια κατεύθυνση για το πως να ελέγξει τα στοιχεία του διανύσματος εισόδου, δίνοντας την περισσότερη προσοχή σε κάποια συγκεκριμένα στοιχεία στα οποία τον παραπέμπει το hint vector.

4.2.1 Περιγραφή του Προβλήματος

Για να περιγράψουμε την αρχιτεκτονική και τη λειτουργία του GAIN, θεωρούμε τα παρακάτω:

- Τα πραγματικά δεδομένα $x = (x_1, ..., x_d)$, διανύσματα στον R^d, που ακολουθούν κατανομή $P(X)$.
- Τις ετικέτες y των x.
- Για κάθε x, ένα διάνυσμα $m = (m_1, ..., m_d)$ στο $[0,1]^d$, το οποίο έχει τιμή 1 στις θέσεις όπου τα αντίστοιχα στοιχεία του x υπάρχουν και 0 στις θέσεις όπου τα στοιχεία του x λείπουν. Έτσι προκύπτει και ο πίνακας M διαστάσεων $d \times d$ με δυαδικές τιμές (mask matrix) που λειτουργεί σαν μάσκα πάνω στον πίνακα X των πραγματικών δεδομένων.
- Τα διανύσματα $\tilde{x}$ στον R^d για τα οποία ισχύει:

$$\tilde{x}_i = \begin{cases} x_i, & \text{αν } m_i = 1. \\ *, & \text{αλλιώς.} \end{cases} \tag{1}$$

  Το σύμβολο $*$ συμβολίζει μια απαρατήρητη τιμή που δεν ανήκει στο x_i. Επομένως τα m_i δείχνουν ποιες τιμές έχουμε παρατηρήσει και ποιες όχι στο διάνυσμα εισόδου.

- Τα διανύσματα θορύβου $z = (z_1, ..., z_d)$ είναι ανεξάρτητα όλων των άλλων μεταβλητών και ακολουθούν μια τυχαία κατανομή πιθανότητας.

4.2.2 Generator και Discriminator

Ο generator παίρνει τα διανύσματα $\tilde{x}$, τα mask διανύσματα m και τα διανύσματα θορύβου z ως είσοδο, και δίνει στην έξοδο τα διανύσματα $\bar{x}$, το imputed διάνυσμα. Δηλαδή αν G η συνάρτηση του generator, τότε $\bar{x} = (\bar{x_1}, ..., \bar{x_d})$ στον R^d, με

$$\bar{x} = G(\tilde{x}, m, (1 - m)z) \tag{2}$$

και αντίστοιχα προκύπτουν τα imputed διανύσματα από το generator

$$\hat{x} = m \odot \tilde{x} + (1 - m) \odot \bar{x} \tag{3}$$

όπου $\odot$ κατά στοιχείο πολλαπλασιασμός διανυσμάτων.

Με άλλα λόγια, παράγεται κάποιος θόρυβος (Z) με τη βοήθεια του οποίου το generative δίκτυο προσομοιώνει τα πραγματικά δεδομένα. Τα προσομοιωμένα $\hat{x}$ στη συνέχεια δίνονται ως είσοδος στο discriminator δίκτυο, εναλλάξ με πραγματικά δεδομένα. Σε αντίθεση με τα GAN τώρα, ο discriminator δεν αποφασίζει κάθε φορά αν ολόκληρο το διάνυσμα εισόδου είναι imputed ή πραγματικό, αλλά αποφασίζει ποια στοιχεία του διανύσματος εισόδου είναι imputed και ποια πραγματικά.

4.2.3 Hint Vector

Το hint vector H είναι τυχαία μεταβλητή, εξαρτάται από τον πίνακα M, και πρέπει να το ορίσουμε έτσι ώστε να δίνει επαρκή πληροφορία στο discriminator, αλλιώς υπάρχει ο κίνδυνος να υπάρξουν πολλές κατανομές από το generator network που θα είναι βέλτιστα ως προς το discriminator network. Το H κατασκευάζεται με την εξής διαδικασία: αρχικά δημιουργείται ένα διάνυσμα b στον R^d, έτσι ώστε μόνο ένα από τα στοιχεία του να είναι 0 και τα υπόλοιπα να είναι 1. Το ποιο στοιχείο θα είναι 0 διαλέγεται τυχαία από την $U(1, ..., d)$. Τότε το h (ως instance της τυχαίας μεταβλητής H) ορίζεται ως

$$h = b \odot m + 0.5 * (1 - b) \tag{4}$$

δηλαδή είναι ίσο με το διάνυσμα m εκτός από ένα στοιχείο το οποίο παίρνει την τιμή 0.5. Έτσι, το H υποδεικνύει στο discriminator ποια στοιχεία έλλειπαν αρχικά παραλείποντας μόνο ένα από αυτά. Προφανώς η υπόδειξη αυτή δεν είναι άμεση, αλλά μια πληροφορία που μετασχηματίζεται μέσα στο δίκτυο του discriminator. Ο discriminator πλέον μπορεί να περιγραφεί ως μια συνάρτηση

$$D : (\hat{x}, h) \to [0, 1]^d \tag{5}$$

με το i-στοιχείο του $D(x)$ να δηλώνει την πιθανότητα το i-στοιχείο του διανύσματος x να είναι πραγματικό.

4.2.4 Συναρτήσεις Loss και Κόστους

Η εκπαίδευση του discriminator επικεντρώνεται στη μεγιστοποίηση της πιθανότητας να γίνει η σωστή πρόβλεψη για το ποια στοιχεία του διανύσματος εισόδου είναι imputed, το οποίο αντιστοιχεί στη σωστή πρόβλεψη του διανύσματος M. Ορίζεται λοιπόν η ποσότητα $V(G, D)$ ως:

$$V(G, D) = E_{\hat{X},M,H}[M^T log D(\hat{X}, H) + (1 - M)^T log(1 - D(\hat{X}, H))] \tag{6}$$

Έτσι ο στόχος της εκπαίδευσης του GAIN συνοψίζεται στο $min_G max_D V(D, G)$.

Για το back-propagation είναι απαραίτητο επίσης να οριστούν οι συναρτήσεις loss. Για το discriminator ορίζουμε την

$$L_D : \{0, 1\}^d \times [0, 1]^d \times \{0, 1\}^d \to R$$

$$L_D(m, \hat{m}, b) = \sum_{i:b_i=0} [m_i log(\hat{m}_i) + (1 - m_i) log(1 - \hat{m}_i)] \tag{7}$$

όπου $\hat{m}_i = D(\hat{x}_i)$, δηλαδή ένα διάνυσμα κάθε θέση του οποίου δείχνει την πιθανότητα το αντίστοιχο στοιχείο να είναι πραγματικό.

Για την εκπαίδευση του generator στόχος είναι να διασφαλιστεί ότι οι imputed τιμές των ελλιπών στοιχείων ($m_j = 0$) ξεγελούν τον discriminator, αλλά ταυτόχρονα πρέπει να διασφαλιστεί και ότι οι τιμές που παράγει ο

generator είναι κοντά στις πραγματικές τιμές που έχουν παρατηρηθεί. Για αυτόν ακριβώς το λόγο ορίζονται δύο loss συναρτήσεις για το generator. Η πρώτη είναι η

$$\mathcal{L}_G : \{0,1\}^d \times [0,1]^d \times \{0,1\}^d \to R$$

$$\mathcal{L}_G(m,\hat{m},b) = -\sum_{i:b_i=0} [(1-m_i)log(\hat{m}_i)] \tag{8}$$

και η δεύτερη η

$$\mathcal{L}_M : R^d \times R^d \to R$$

$$\mathcal{L}_M(x,x') = \sum_{i=1}^{d} m_i L_M(x,x') \tag{9}$$

όπου

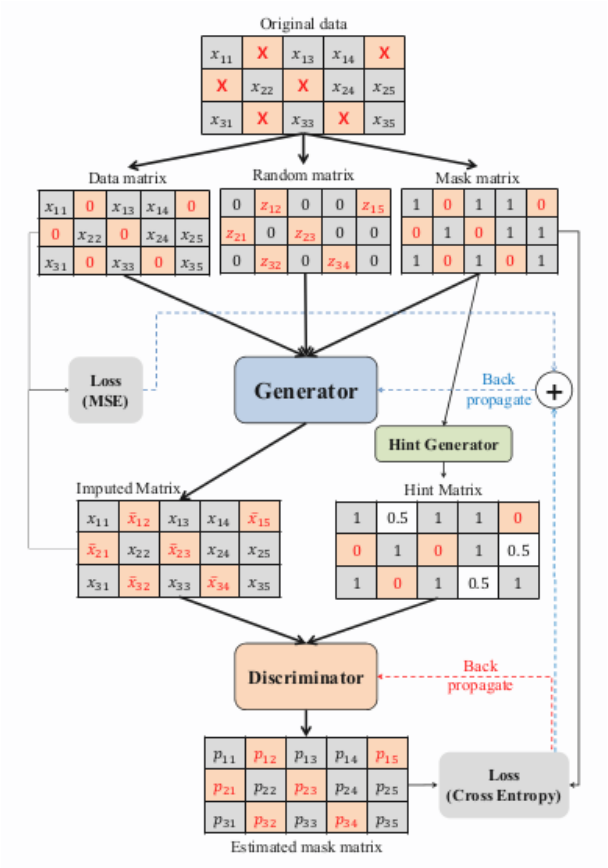
$$L_M(x_i,x_i') = \begin{cases} (x_i' - x_i), & \text{αν η } x_i \text{ είναι συνεχής.} \\ -x_i log(x_i'), & \text{αν η } x_i \text{ είναι διακριτή.} \end{cases} \tag{10}$$

Επομένως η $\mathcal{L}_G$ αντιστοιχεί στα ελλιπή στοιχεία ($m_i = 0$) και η $\mathcal{L}_M$ στα παρατηρούμενα στοιχεία ($m_i = 1$). Η $\mathcal{L}_G$ μικραίνει όταν το $\hat{m}_i$ τείνει στο 1 για i τέτοια ώστε $m_i = 0$. Με άλλα λόγια, η $\mathcal{L}_G$ είναι μικρότερη όταν ο discriminator δεν μπορεί να διαχωρίσει τις imputed τιμές και αποφασίζει ότι αυτές είναι πραγματικές. Η $\mathcal{L}_M$ ελαχιστοποιείται όταν τα ανακατασκευασμένα features είναι πολύ κοντά στα πραγματικά, παρατηρούμενα features. έτσι, το δίκτυο του generator εκπαιδεύεται ώστε να ελαχιστοποιεί το βαρυκεντρισμένο άθροισμα των δύο losses ως εξής:

$$min_G \sum_{j=1}^{k_G} \mathcal{L}_G(m(j),\hat{m}(j),b(j)) + \alpha \mathcal{L}_M(\tilde{x}(j),(\hat{x}(j))) \tag{11}$$

όπου α υπερπαράμετρος και k_G το μέγεθος των mini-batches.

Στην Εικόνα 2 δίνεται μια απεικόνιση της αρχιτεκτονικής του GAIN και στον Αλγόριθμο 1 ο αλγόριθμος σε ψευδοκώδικα.



Εικόνα 2: GAIN αρχιτεκτονική [1].

Αλγόριθμος 1 Ψευδοκώδικας GAIN

while Δεν έχουμε σύγκλιση του σφάλματος στην εκπαίδευση **do**
  τράβηξε k_D δείγματα από το $[(\tilde{x}(j), m(j))]_{k_D}^{j=1}$ ▷ Βελτιστοποίηση Discriminator (1)
  τράβηξε k_D i.i.d. δείγματα, $[z(j)]_{k_D}^{j=1}$ από το Z
  τράβηξε k_D i.i.d. δείγματα, $[b(j)]_{k_D}^{j=1}$ από το B
  for $j = 1, ..., k_D$ **do**
    $\tilde{x}(j) = G(\tilde{x}(j), m(j), z(j))$
    $\hat{x}(j) = m(j) \odot \tilde{x}(j) + (1 - m(j)) \odot \tilde{x}(j)$
    $h(j) = b(j) \odot m(j) + 0.5(1 - b(j))$
  end for
  Ανανέωσε τον Discriminator D, με χρήση Stochastic Gradient Descent (SGD)

$$\nabla_D - \sum L_D(m(j), D(\tilde{x}(j)), h(j), b(j))_{j=1}^{k_D}$$

  τράβηξε k_G δείγματα από το $[(\tilde{x}(j), m(j))]_{k_G}^{j=1}$ ▷ Βελτιστοποίηση Generator (2)
  τράβηξε k_G i.i.d. δείγματα, $[z(j)]_{k_G}^{j=1}$ από το Z
  τράβηξε k_G i.i.d. δείγματα, $[b(j)]_{k_G}^{j=1}$ από το B
  for $j = 1, ..., k_D$ **do**
    $h(j) = b(j) \odot m(j) + 0.5(1 - b(j))$
  end for
  Ανανέωσε τον generator G, με χρήση SGD

$$\nabla_G \sum \mathcal{L}_G(m(j), \hat{m}(j), b(j)) + \alpha \mathcal{L}_M(x(j), \tilde{x}(j))_{j=1}^{k_G}$$

end while

5 Πειράματα/Αποτελέσματα/Συζήτηση

Σε αυτήν την παράγραφο περιγράφουμε τα πειράματα που εκτελέσαμε για να αξιολογήσουμε την απόδοση του GAIN και παρουσιάζουμε τα αποτελέσματα αυτών. Τα πειράματα βασίστηκαν σε αυτά που εφαρμόστηκαν στο paper των Yoon, Jordon & van der Schaar [1] και στοχεύουν στην αξιολόγηση (α) της αρχικτονικής του GAIN, (β) της ικανότητάς του για imputation, (γ) της απόδοσής του στη μεταβολή παραμέτρων (όπως το ποσοστό των missing values) και (δ) της ικανότητας πρόβλεψης πάνω στα imputed δεδομένα χρησιμοποιώντας κάποιον ταξινομητή. Επιπρόσθετα, πραγματοποιούμε ορισμένες αλλαγές στην πρότυπη αρχιτεκτονική και κάνουμε επέκταση της μεθοδολογίας σε σύνολα δεδομένων με εικόνες, ώστε να δούμε και γραφικά πόσο καλά αποδίδει το GAIN με χρήση διάφορων αρχιτεκτονικών, όπως αναλύεται παρακάτω.

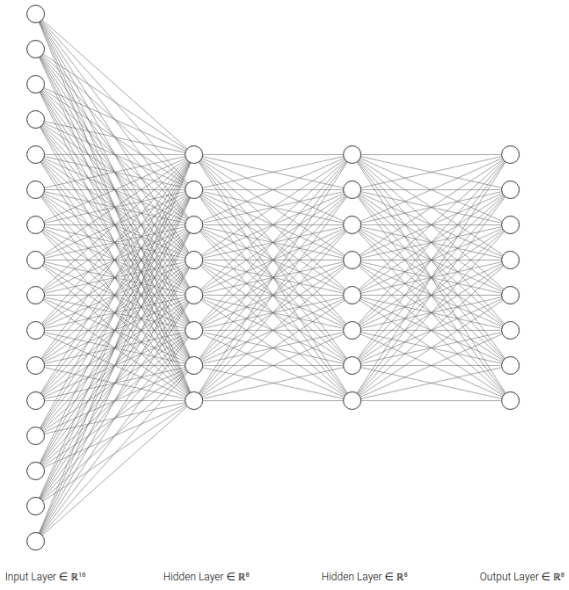
Ο κώδικας που υλοποιήσαμε σε python-3 [22] βασίστηκε στην πρότυπη υλοποίηση του GAIN framework για imputation, η οποία χρησιμοποίησε τη βιβλιοθήκη TensorFlow [23] και είναι διαθέσιμη στο github. Συγκεκριμένα, χρησιμοποιήσαμε τις βιβλιοθήκες PyTorch [24], NumPy [25] και scikit-learn [26] για την προσαρμογή της πρότυπης υλοποίησης, την αναπαραγωγή αποτελεσμάτων και την βελτίωση της αρχιτεκτονικής.

5.1 Πρότυπη Αρχιτεκτονική GAIN

Σύμφωνα με την πρότυπη υλοποίηση του GAIN framework που αναφέραμε προηγουμένως, το GAIN αρχικά υλοποιήθηκε με την αρχιτεκτονική της Εικόνας 3. Αποτελείται από το επίπεδο εισόδου με αριθμό νευρώνων = # Χαρακτηριστικών × 2. Στη συνέχεια το 1ο και το 2ο κρυφό επίπεδο, καθώς και το επίπεδο εξόδου, αποτελούνται από αριθμό νευρώνων = # Χαρακτηριστικών. Αυτή την αρχιτεκτονική ακολουθούν και ο generator και ο discriminator. Η είσοδος αποτελείται από διάνυσμα διπλάσιου μεγέθους των χαρακτηριστικών του συνόλου δεδομένων λόγω της χρήσης mask και hint vectors στα 2 μοντέλα του GAIN.

Χρησιμοποιούμε τη Rectified Linear Unit (ReLu) ως συνάρτηση ενεργοποίησης κάθε επιπέδου εκτός από το επίπεδο εξόδου όπου χρησιμοποιούμε τη σιγμοειδή συνάρτηση ενεργοποίησης. Ο αριθμός των batches είναι 128 τόσο για τον generator όσο και για τον discriminator. Για τη διαδικασία της εκπαίδευσης ορίστηκαν 10,000 εποχές και έγινε χρήση της μεθόδου καθόδου ADAM και ρυθμό μάθησης 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-08$. Επιπλέον, τα διανύσματα θορύβου Z προσομοιώθηκαν από την ομοιόμορφη κατανομή. Ειδικότερα

για missing rate 0.5 χρησιμοποιείται η ομοιόμορφη κατανομή $U((0,1)^d)$, ενώ για missing rate 0.2 η $U((0,0.01)^d)$. Τέλος, η υπερπαράμετρος α λαμβάνει την τιμή 10.



Εικόνα 3: Πρότυπη αρχιτεκτονική GAIN.

5.2 Ικανότητα Imputation - Αποτελέσματα RMSE

Στην παράγραφο αυτή παρουσιάζουμε τα αποτελέσματα των πειραμάτων μας σε 5 διαφορετικά datasets (breast, spam, letter, credit, news). Επικεντρωνόμαστε στη μετρική RMSE, την οποία μελετάμε σε σχέση με τις συναρτήσεις loss και στη συνέχεια αλλάζοντας κάποιες παραμέτρους που αφορούν τα σύνολα δεδομένων.

5.2.1 Συναρτήσεις Loss

Η λειτουργία του GAIN βασίζεται σε τρία δομικά στοιχεία, το loss $\mathcal{L}_G$ που αντιστοιχεί στο imputation των missing values από το δίκτυο του generator, το loss $\mathcal{L}_M$ που αντιστοιχεί στην ανακατασκευή των αρχικών δεδομένων από το δίκτυο του generator επίσης, και το hint vector που βοηθάει το δίκτυο του disriminator στη διάκριση μεταξύ των πραγματικών από τα imputed δεδομένα. Επομένως, για να αξιολογήσουμε τη συνεισφορά των στοιχείων αυτών, εξαιρέσαμε κάθε φορά ένα ή δύο από αυτά και μετρήσαμε το RMSE που προέκυπτε. Χρησιμοποιήσαμε 5-fold cross validation και εκτελέσαμε κάθε πείραμα 2 φορές. Τέλος, εξάγαμε το μέσο όρο και την τυπική απόκλιση των RMSE αποτελεσμάτων. Τα αποτελέσματα συνοψίζονται στον Πίνακα 1.

Πίνακας 1 Συνεισφορά κάθε loss συνάρτησης στον αλγόριθμο GAIN (Mean±Std του RMSE (Gain (%))).

Algorithm	**Breast**	**Spam**	**Letter**	**Credit**	**News**
GAIN	0.1073 ± 0.0006	0.0546 ± 0.0005	0.1357 ± 0.0015	0.1551 ± 0.0038	0.2077 ± 0.0004
GAIN w/o $\mathcal{L}_G$	0.1202 ±0.0007	0.0530 ±0.0001	0.1333 ±0.0015	0.1829 ±0.0019	0.2480 $\pm 9.437e-05$
GAIN w/o $\mathcal{L}_M$	0.4455 ±0.0072	0.1433 ±0.0254	0.4828 ±0.0049	0.3540 ±0.0329	0.5273 ±0.0338
GAIN w/o Hint	0.1030 ±0.0021	0.0561 $\pm 7.287e-05$	0.1349 ±0.0018	0.1795 ±0.0020	0.2097 ±0.0003
GAIN w/o Hint & $\mathcal{L}_M$	0.2513 ±0.0004	0.1407 ±0.0024	0.1764 ±0.0047	0.1787 ±0.0020	0.2880 ±0.0008

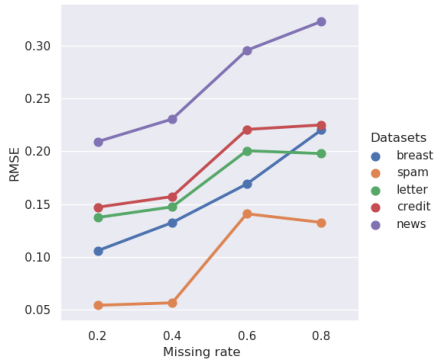
Όπως παρατηρούμε, το μικρότερο RMSE επιτυγχάνεται κάθε φορά όταν χρησιμοποιούνται και τα τρία loss functions. Αξίζει να σημειώσουμε ότι μεγάλο ρόλο φαίνεται να παίζει η $\mathcal{L}_M$ loss συνάρτηση καθώς, όταν παραλείπεται, το RMSE αυξάνεται κατά περίπου 2 - 4 φορές, αναλόγως το dataset (η αύξηση μεταβάλλεται φυσικά σε κάθε διαφορετικό τρέξιμο αλλά παραμένει σε ένα σχετικό εύρος). Τη μικρότερη συνεισφορά φαίνεται να έχει η $\mathcal{L}_G$ συνάρτηση καθώς το RMSE αυξάνεται σχετικά λίγο όταν αυτή παραλείπεται. Στα

αντίστοιχα αποτελέσματα του paper μπορεί κανείς να δει ότι η $\mathcal{L}_M$ επίσης είναι αυτή που συμβάλλει στο μικρότερο RMSE, ωστόσο τη μικρότερη συμβολή φαίνεται πως τη δίνει το hint vector. Τέτοιες διαφορές είναι λογικές καθώς στα αποτελέσματα για διάφορες μετρικές που παίρνουμε στην εκπαίδευση νευρωνικών δικτύων διαδραματίζει καθοριστικό ρόλο η ακριβής αρχιτεκτονική του δικτύου και η ρύθμιση των διάφορων υπερπαραμέτρων. Ωστόσο είναι σημαντικό να κρατήσουμε ότι ο τρόπος μεταβολής του RMSE είναι πολύ παρόμοιος πέραν αυτών των μικροδιαφορών, και έτσι μπορούμε να πούμε ότι οι πειραματισμοί μας προσέφεραν ακόμα μια ένδειξη υπέρ της αποδοτικότητας του GAIN δικτύου.

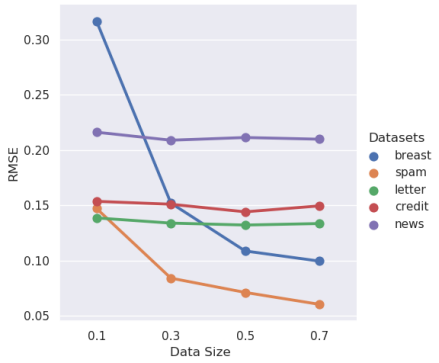
5.2.2 RMSE και Αλλες Παράμετροι

Για κάθε ένα από τα 5 dataset μετρήσαμε το RMSE σε σχέση με:

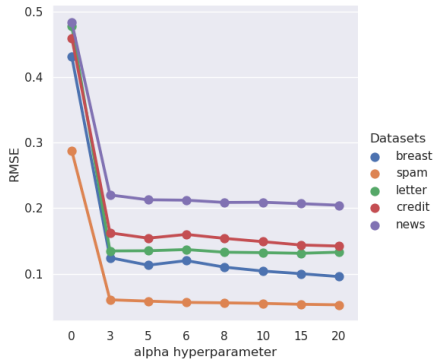
- το missing rate - το ποσοστό των components που λείπουν από τα διανύσματα.
- το data size - το μέγεθος ενός υποσυνόλου του αρχικού dataset.
- την υπερπαράμετρο *alpha* (α) - υπερπαράμετρος που εμφανίζεται στη loss συνάρτηση $\mathcal{L}_M$.
- τον αριθμό των εποχών (πάνω στο SPAM dataset).



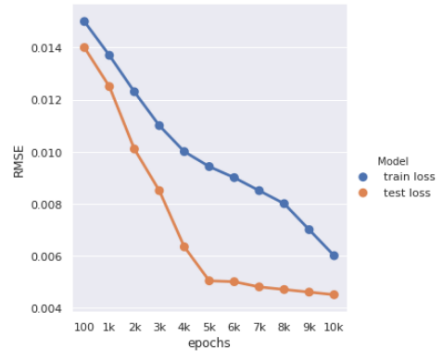
Εικόνα 4: RMSE - Missing Rate.



Εικόνα 5: RMSE - Data Size.



Εικόνα 6: RMSE - α hyperparameter.



Εικόνα 7: Learning Curve: εποχές - RMSE.

Στην Εικόνα 4 απεικονίζεται το διάγραμμα RMSE - missing rate, στο οποίο παρατηρούμε την τάση το RMSE να αυξάνεται όσο αυξάνεται και το missing rate. Η μεγαλύτερη αύξηση (στη συνολική μεταβολή του missing rate από το 0.2 στο 0.8) εντοπίζεται στο breast και το spam dataset, το οποίο μπορεί να δικαιολογηθεί από το γεγονός ότι και τα δύο έχουν μικρό μέγεθος, και άρα όταν εκλίπουν παραπάνω στοιχεία, τα εναπομείναντα δεν επαρκούν για την εκπαίδευση. Τα news και letter εμφανίζουν μια αύξηση της τάξεως περίπου 50%, ενώ το credit εμφανίζει μια ανωμαλία με τη μορφή αυξομείωσης, και τη μικρότερη συνολική αύξηση.

Όταν πειραματιζόμαστε με υποσύνολα του δείγματος (Εικόνα 5), παρατηρούμε ότι εν γένει το RMSE αυξάνεται όσο μειώνεται το μέγεθος του υποσυνόλου. Τα letter, credit και news datasets παρουσιάζουν τη μικρότερη αύξηση στο RMSE, ενδεχομένως λόγω του μεγάλου μεγέθους που έχουν αρχικά, με την έννοια ότι και υποσύνολα των αρχικών συνόλων επαρκούν για εκπαίδευση. Μεγάλη αύξηση γνώρισε το RMSE του breast, του μικρότερου σε μέγεθος συνόλου δεδομένων. Στο spam το RMSE μειώθηκε κατά περίπου 35%.

Η Εικόνα 6 αποτυπώνει τη μεταβολή του RMSE σε σχέση με την υπερπαράμετρο *alpha* στο loss function $\mathcal{L}_M$. Σε αυτό το διάγραμμα βασιστήκαμε ουσιαστικά για να ρυθμίσουμε στον κώδικά μας το alpha. Η Εικόνα 7 περιγράφει την καμπύλη μάθησης (learning curve) στο spam dataset με τη μεταβολή του RMSE σε σχέση με τις εποχές εκπαίδευσης - από 100 μέχρι 10,000. Το loss στην εκπαίδευση είναι σταθερά μεγαλύτερο του loss στην επικύρωση και φθίνει σχεδόν γραμμικά. Το loss στην επικύρωση φθίνει με μεγάλη κλίση μέχρι τις 5,000 εποχές και μετά ο ρυθμός μειώνεται εμφανώς και σταθερά μέχρι τις 10,000 εποχές, όπου και η διαφορά των δυο losses έχει μικρύνει αρκετά, κάτι που δείχνει ότι η επιλογή των 10,000 εποχών για την εκπαίδευση οδηγεί σε ένα good fit του μοντέλου.

5.2.3 Ικανότητα Πρόβλεψης

Αξιολογούμε την ικανότητα πρόβλεψης στα imputed data με το Area Under the Receiver Operating Characteristic Curve (AUROC) ως μετρική, το οποίο υπολογίζουμε για τα dataset breast, spam, credit και news. Η μετρική AUROC μετράει το εμβαδό κάτω από την καμπύλη ROC. Σε ένα γράφημα καμπύλης ROC ο οριζόντιος άξονας αντιπροσωπεύει το False Positive Rate (FPR) και ο κάθετος άξονας το True Positive Rate (TPR), τα οποία - θεωρώντας ότι σε μια δυαδική ταξινόμηση 1 = positive και 0 = negative - ορίζονται ως εξής:

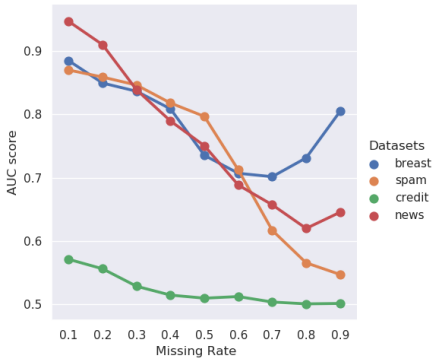
$$FPR = \frac{FP}{FP + TN} = \frac{\text{\# 0s labeled as 1s}}{\text{\# 0s labeled as 1s + \#1s labeled as 1s}}$$

και

$$TPR = \frac{TP}{TP + FN} = \frac{\text{\# 1s labeled as 1s}}{\text{\# 1s labeled as 1s + \#1s labeled as 0s}}$$

Κάθε σημείο της καμπύλης ROC συνδέει το FPR με το TPR για ένα δοθέν κατώφλι του αλγορίθμου ταξινόμισης. Χαρακτηριστικό της μετρικής AUROC είναι η ανεξαρτησία της από την επιλογή του κατωφλιού. Μεγαλύτερη AUROC τιμή σημαίνει μεγαλύτερο εμβαδό κάτω από τη ROC καμπύλη, και κατ'επέκταση μεγαλύτερο TPR από FPR στις ταξινομίσεις.

Η Εικόνα 8 δείχνει τη μετρική AUROC σε σχέση με το missing rate για τα dataset που αναφέραμε με τη χρήση του αλγορίθμου λογιστικής παλινδρόμησης. Παρατηρούμε πως γενικά όσο αυξάνεται ο θόρυβος στα σύνολα δεδομένων τόσο μειώνεται και η προβλεπτική ικανότητα, με μια εξαίρεση στο breast dataset, το οποίο στο τέλος παρουσιάζει μια αύξηση.



Εικόνα 8: AUROC vs Missing Rate.

5.2 Αλλαγή Αρχιτεκτονικής - Αποτελέσματα RMSE

Παραποιούμε την πρότυπη υλοποίηση του GAIN framework, προσθέτοντας ένα επιπλέον κρυφό επίπεδο. Συγκεκριμένα, η παραποιημένη αυτή αρχιτεκτονική GAIN, την οποία ονομάζουμε GAIN-TD, αποτελείται από το επίπεδο εισόδου με αριθμό νευρώνων = # Χαρακτηριστικών × 2, τρία κρυφά επίπεδα και το επίπεδο εξόδου με αριθμό νευρώνων = # Χαρακτηριστικών. Ο αριθμός νευρώνων των κρυφών επιπέδων διαμορφώθηκε ως εξής: το 1ο και το 3ο κρυφό επίπεδο αποτελούνται από αριθμό νευρώνων = # Χαρακτηριστικών, ενώ το 2ο κρυφό επίπεδο αποτελείται από αριθμό νευρώνων = # Χαρακτηριστικών / 2.

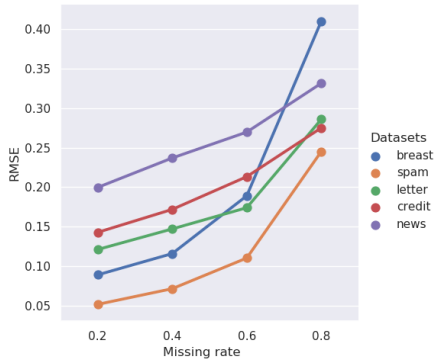
Χρησιμοποιούμε την *tanh* ως συνάρτηση ενεργοποίησης κάθε επιπέδου εκτός από το επίπεδο εξόδου όπου χρησιμοποιούμε τη σιγμοειδή συνάρτηση ενεργοποίησης. Ο αριθμός των batches είναι 64 τόσο για τον generator όσο και για τον discriminator. Όπως και στην πρότυπη υλοποίηση, για τη διαδικασία της εκπαίδευσης

ορίστηκαν 10,000 εποχές και έγινε χρήση της μεθόδου καθόδου ADAM και ρυθμό μάθησης 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-08$, τα διανύσματα θορύβου Z προσομοιώθηκαν από την ομοιόμορφη κατανομή και η υπερπαράμετρος α λαμβάνει την τιμή 10.

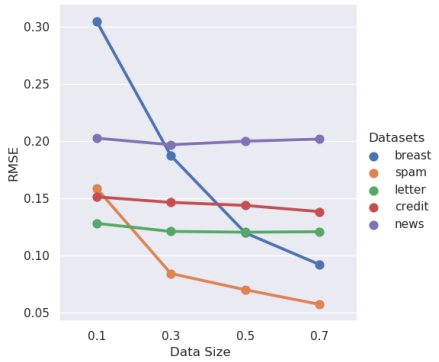
Τα αποτελέσματα των πειραμάτων μας σε αυτή την περίπτωση συνοψίζονται στον Πίνακα 2.

Πίνακας 2 Συνεισφορά κάθε loss συνάρτησης στον αλγόριθμο GAIN-TD (Mean±Std του RMSE (Gain (%))).

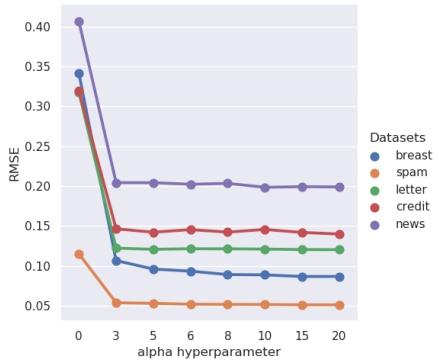
Algorithm	Breast	Spam	Letter	Credit	News
GAIN-TD	0.08687 ± 0.003	0.05156 ± 0.0006	0.1207 ± 0.002	0.1405 ± 0.0067	0.1978 ± 0.0018
GAIN-TD w/o $\mathcal{L}_G$	0.08689 ±0.0063	0.0518 ±0.0009	0.1223 ±0.0020	0.1791 ±0.0044	0.2246 ±0.0028
GAIN-TD w/o $\mathcal{L}_M$	0.3760 ±0.0216	0.1195 ±0.0417	0.3186 ±0.0154	0.2405 ±0.0273	0.3710 ±0.0334
GAIN-TD w/o Hint	0.0982 ±0.0086	0.0514 ±0.0011	0.1216 ±0.0021	0.1520 ±0.0031	0.2002 ±0.0059
GAIN-TD w/o Hint & $\mathcal{L}_M$	0.2955 ±0.0122	0.0762 ±0.0120	0.3269 ±0.0479	0.2054 ±0.0164	0.2640 ±0.0305



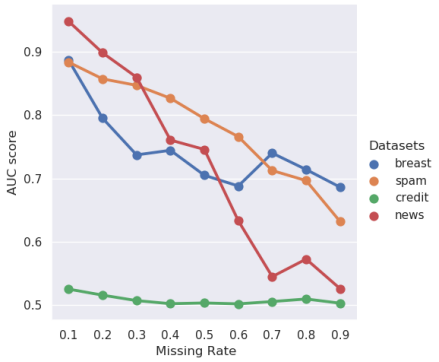
Εικόνα 9: RMSE - Missing Rate.



Εικόνα 10: RMSE - Data Size.



Εικόνα 11: RMSE - α hyperparameter.



Εικόνα 12: AUROC vs Missing Rate.

Συγκριτικά με τα αντίστοιχα αποτελέσματα της πρότυπης αρχιτεκτονικής, το GAIN-TD πέτυχε μικρότερο RMSE όταν συμπεριλήφθηκαν όλες οι συναρτήσεις loss. Το ίδιο ισχύει και στην περίπτωση που παραλείφθηκε η $\mathcal{L}_G$ ή το hint vector, και επίσης κι εδώ φάνηκε ότι αυτά τα δύο στοιχεία είχαν τη μικρότερη συνεισφορά στη μείωση του RMSE. Μικρότερα RMSE σε σχέση με την πρότυπη αρχιτεκτονική πήραμε και όταν εξαιρέσαμε την $\mathcal{L}_M$, η οποία και πάλι διαδραμάτισε τον πιο καθοριστικό ρόλο στη μεταβολή του RMSE, το οποίο αυξήθηκε 1.7 με 4.3 φορές. Η διαφορά των δύο αρχιτεκτονικών με βάση το RMSE βρίσκεται στην περίπτωση που παραλείπονται ταυτόχρονα και η $\mathcal{L}_M$ και το hint vector, όπου αφενός πήραμε μεγαλύτερα RMSE scores από την πρότυπη, αφετέρου τα scores αυτά ήταν μικρότερα συγκριτικά με την περίπτωση που μόνο το $\mathcal{L}_M$ εξαιρέθηκε. Όταν δηλαδή το RMSE υπολογίστηκε με βάση μόνο με το $\mathcal{L}_G$, δηλαδή την ικανότητα του generator να ξεγελάει το discriminator στα στοιχεία που λείπουν, η παρουσία του hint vector φαίνεται να επιδράει αρνητικά.

Στην Εικόνα 9 αποτυπώνεται η αύξηση του RMSE σε σχέση με την αύξηση του missing rate. Μπορούμε να παρατηρήσουμε ότι οι μεταβολές είναι πιο ήπιες στις περιπτώσεις των μεγάλων datasets, credit και news, μικρότερες στα spam και letter και ραγδαία -εκθετική- στην περίπτωση του breast, του μικρότερου dataset. Σε σχέση με το αντίστοιχο διάγραμμα στην πρότυπη αρχιτεκτονική, σε αυτήν την περίπτωση παρατηρούμε μεγαλύτερη ομοιομορφία και σταθερότητα στη μεταβολή του RMSE. Τα διαγράμματα RMSE-data size της Εικόνας 10 και RMSE-hyperparameter α της Εικόνας 11 είναι πολύ παρόμοια με αυτά της πρότυπης αρχιτεκτονικής. Τέλος, η Εικόνα 12 αντιστοιχεί στην Εικόνα 8 της πρότυπης αρχιτεκτονικής και αφορά τη μετρική AUROC σε σχέση με το missing rate. Με εξαίρεση το breast, που στην περίπτωσή της αρχιτεκτονικής που δοκιμάσαμε καταλήγει με καθοδική πορεία ενώ στην πρότυπη με ανοδική, τα υπόλοιπα datasets κινούνται με παρόμοιο τρόπο ως προς την κλίση της καθόδου του AUROC.

5.3 Επέκταση GAIN σε Δεδομένα Εικόνας - Αλλαγή Αρχιτεκτονικής

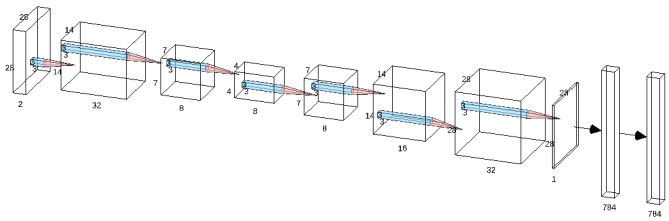
Στην προσπάθειά μας να επεκτείνουμε αποτελεσματικά τη μέθοδο GAIN και σε δεδομένα εικόνας, δοκιμάσαμε διάφορες παραλλαγές της πρότυπης αρχιτεκτονικής (Εικόνα 3), όπως π.χ. να γίνει χρήση Συνελικτικών Νευρωνικών Δικτύων (CNN). Τη συγκεκριμένη αρχιτεκτονική την ονομάσαμε GAIN-CNN. Παράλληλα επιστρατεύσαμε τη χρήση περισσότερων κρυφών επιπέδων και ονομάσαμε το μοντέλο GAIN-DEEP. Τέλος, πραγματοποιήθηκε και μια πιο απλή αρχιτεκτονική με όνομα GAIN-FAST, στην οποία έχουμε λιγότερους νευρώνες στα κρυφά επίπεδα. Η τελευταία φάνηκε συνάμα αποτελεσματική και γρήγορη.

Στη συνέχεια περιγράφουμε τις αρχιτεκτονικές που χρησιμοποιήσαμε και παρουσιάζουμε τα αποτελέσματα των πειραμάτων μας σε 2 διαφορετικά datasets (MNIST, Pneumonia). Επικεντρωνόμαστε στη μετρική RMSE, και οπτικοποιούμε τα αποτελέσματα του imputation που πραγματοποιείται.

5.3.1 Περιγραφή Αρχιτεκτονικών

Για το μοντέλο GAIN-FAST, έχουμε εμπνευστεί από την αρχιτεκτονική των Αυτο-Κωδικοποιητών (AE), η οποία είναι αρκετά αποδοτική στην εκμάθηση χαρακτηριστικών και στην εύρεση συγκεκριμένων μετασχηματισμών [27]. Συγκεκριμένα, το μοντέλο GAIN-FAST αποτελείται από το επίπεδο εισόδου με αριθμό νευρώνων = # Χαρακτηριστικών $\times$ 2, δύο κρυφά επίπεδα και το επίπεδο εξόδου με αριθμό νευρώνων = # Χαρακτηριστικών. Ο αριθμός νευρώνων των κρυφών επιπέδων διαμορφώθηκε ως εξής: το 1ο κρυφό επίπεδο αποτελείται από αριθμό νευρώνων = 256, ενώ το 2ο από αριθμό νευρώνων = 128.

Η ιδέα για το GAIN-CNN βασίζεται σε σχετικές έρευνες [28], [29], στις οποίες παρατηρείται πως το νευρωνικό δίκτυο μπορεί να μάθει αρκετά καλύτερα τα χαρακτηριστικά μιας εικόνας με τη χρήση της συνέλιξης. Το μοντέλο, το οποίο υλοποιήσαμε στο σύνολο δεδομένων MNIST, έχει ως είσοδο ένα διάνυσμα $28 \times 28 \times 2$, λόγω του ότι συνενώσαμε το mask και το hint vector στον generator και τον discriminator με τα δεδομένα σα δεύτερο κανάλι. Στη συνέχεια ακολουθούν πράξεις συνέλιξης μέχρι να κωδικοποιηθεί η εικόνα σε μικρότερη διάσταση και έπειτα ακολουθούν πράξεις αντιστροφής της συνέλιξης για να καταλήξουμε στο μέγεθος της εικόνας 28×28, όπως φαίνεται στην Εικόνα 13. Για το σύνολο δεδομένων Pneumonia, έγιναν οι κατάλληλες τροποποιήσεις ως προς τις εισόδους και τις πράξεις συνέλιξης.



Εικόνα 13: GAIN-CNN.

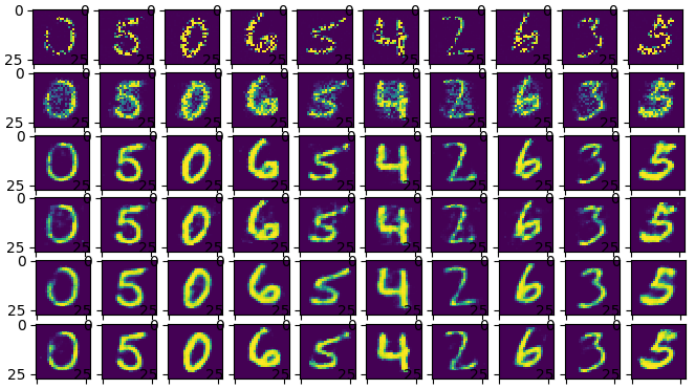
Για το το μοντέλο GAIN-DEEP, χρησιμοποιήσαμε τέσσερα κρυφά επίπεδα. Συγκεκριμένα, στο MNIST dataset το μοντέλο GAIN-DEEP αποτελείται από το επίπεδο εισόδου με αριθμό νευρώνων = # Χαρακτηριστικών $\times$ 2, τέσσερα κρυφά επίπεδα και το επίπεδο εξόδου, τα οποία έχουν αριθμό νευρώνων = # Χαρακτηριστικών. Στο Pneumonia dataset το μοντέλο GAIN-DEEP αποτελείται από το επίπεδο εισόδου με αριθμό νευρώνων = # Χαρακτηριστικών $\times$ 2, τέσσερα κρυφά επίπεδα και το επίπεδο εξόδου με αριθμό νευρώνων = # Χαρακτηριστικών. Ο αριθμός νευρώνων των κρυφών επιπέδων διαμορφώθηκε ως εξής: το 1ο και το 2ο κρυφό επίπεδο

αποτελούνται από αριθμό νευρώνων = 256, ενώ το 3ο και το 4ο κρυφό επίπεδο αποτελούνται από αριθμό νευρώνων = 128. Δηλαδή το GAIN-DEEP, στο συγκεκριμένο σύνολο δεδομένων, τροποποιήθηκε ώστε να είναι ένα πιο βαθύ GAIN-FAST για να μη δημιουργείται πρόβλημα με τη μνήμη της GPU.

Για τις παραπάνω αρχιτεκτονικές, χρησιμοποιούμε τη ReLu ως συνάρτηση ενεργοποίησης κάθε επιπέδου εκτός από το επίπεδο εξόδου όπου χρησιμοποιούμε τη σιγμοειδή συνάρτηση ενεργοποίησης. Ο αριθμός των batches είναι 128 τόσο για τον generator όσο και για τον discriminator. Για τη διαδικασία της εκπαίδευσης ορίστηκαν 10,000 εποχές και έγινε χρήση της μεθόδου καθόδου ADAM και ρυθμό μάθησης 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e{-}08$. Η υπερπαράμετρος α λαμβάνει την τιμή 10. Τα διανύσματα θορύβου Z προσομοιώθηκαν από την ομοιόμορφη κατανομή. Συγκεκριμένα, για missing rate 0.5 χρησιμοποιείται η ομοιόμορφη κατανομή $U((0,1)^d)$, ενώ για missing rate 0.2 η $U((0,0.01)^d)$.

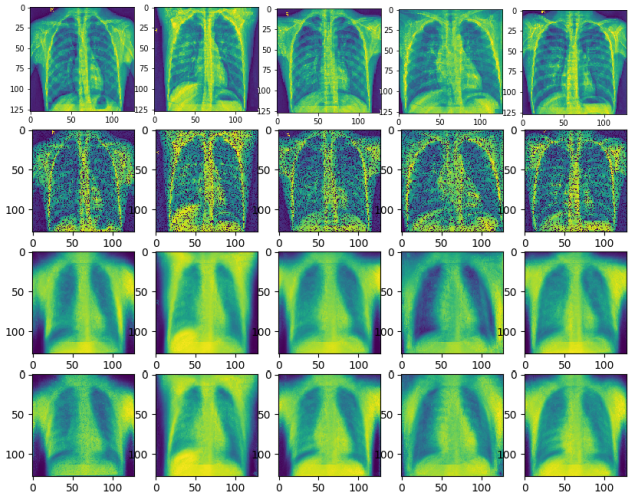
5.3.2 Οπτικοποίηση Imputation - Αποτελέσματα RMSE

Στην Εικόνα 14 έχουμε στην πρώτη γραμμή ένα υποσύνολο δεδομένων εικόνων από το MNIST dataset [9] με 20% θόρυβο. Έπειτα κατά σειρά έχουμε συμπλήρωση ελλιπών τιμών με το GAIN, GAIN-TD, GAIN-FAST, GAIN-CNN και GAIN-DEEP. Γραφικά, φαίνεται ότι το GAIN-CNN και το GAIN-DEEP παράγουν ευκρινέστερες εικόνες.



Εικόνα 14: Imputation από το MNIST dataset.

Στην Εικόνα 15 βλέπουμε στην πρώτη γραμμή ένα υποσύνολο πραγματικών εικόνων από το Pneumonia dataset [10]. Ακολουθούν στη 2η γραμμή οι εικόνες αυτές με 20% θόρυβο. Τέλος στην 3η και 4η γραμμή αντίστοιχα έχουμε συμπλήρωση ελλιπών τιμών με GAIN-FAST και GAIN-DEEP. Παρατηρούμε πως οι 2 τελευταίες γραμμές διαφέρουν αρκετά με την πρώτη, η οποία αντιστοιχεί στις πραγματικές εικόνες.



Εικόνα 15: Imputation από το Pneumonia dataset.

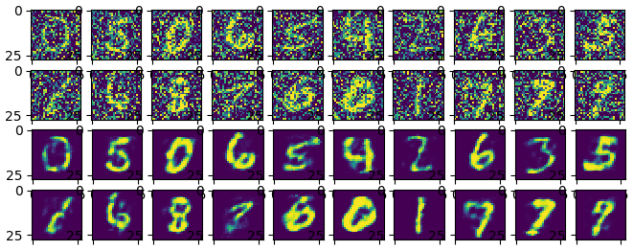
Τα αποτελέσματα του RMSE που λάβαμε στα MNIST και Pneumonia datasets, για όλες τις διαφορετικές αρχιτεκτονικές GAIN, παρουσιάζονται στον Πίνακα 3. Παρατηρούμε ότι τα μικρότερα RMSE επιτυγχάνονται

με τα GAIN-DEEP και GAIN-CNN. Ακολουθεί το GAIN-TD που περιγράψαμε στην ενότητα 5.2, και το GAIN-FAST στο οποίο μειώνονται σταδιακά οι διαστάσεις των στρωμάτων. Τέλος, το GAIN με την πρότυπη αρχιτεκτονική των Yoon et. al δίνει το μεγαλύτερο RMSE. Όλα αυτά τα αποτελέσματα ελήφθησαν με missing rate 0.2. Σημειώνουμε πως στο Pneumonia dataset, λόγω πολλών χαρακτηριστικών και έλλειψη μνήμης, δεν καταφέραμε να δοκιμάσουμε τα δίκτυα GAIN, GAIN-TD και GAIN-CNN.

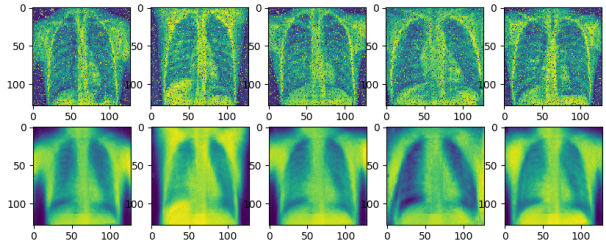
Πίνακας 3 RMSE scores στα σύνολα δεδομένων με εικόνες για 0.2 missing rate.

Architecture	GAIN	GAIN-TD	GAIN-FAST	GAIN-CNN	GAIN-DEEP
MNIST	0.2024	0.1260	0.1386	0.1189	0.1199
Pneumonia	-	-	0.1017	-	0.1007

Παρακάτω παραθέτουμε τα αποτελέσματα από τα 2 σύνολα δεδομένων με εικόνες έχοντας εισάγει 50% θόρυβο. Στις δύο πρώτες γραμμές της Εικόνας 16 βλέπουμε τα δεδομένα με θόρυβο, ενώ στις τελευταίες δύο τα δεδομένα μετά το imputation. Ομοίως και στην Εικόνα 17 η πρώτη γραμμή αναλογεί σε δεδομένα με θόρυβο ενώ η τελευταία στα δεδομένα μετά το imputation. Το imputation πραγματοποιήθηκε με την αρχιτεκτονική GAIN-FAST.



Εικόνα 16: Imputation με θόρυβο 50% στο MNIST dataset - RMSE=0.1630.



Εικόνα 17: Imputation με θόρυβο 50% στο Pneumonia dataset - RMSE=0.1014.

6 Συμπεράσματα/Μελλοντική Έρευνα

Στην παρούσα έρευνα, πραγματοποιείται μια αναπαραγωγή των αποτελεσμάτων του paper "GAIN: Missing Data Imputation using Generative Adversarial Nets" [1]. Επίσης, επεκτείνουμε αυτήν τη μεθοδολογία για συνολά δεδομένων που αφορούν εικόνες όπως το MNIST [9] και το Chest X-Ray Images (Pneumonia) [10]. Εν συνεχεία, προτείνουμε εναλλακτικές αρχιτεκτονικές για τη βελτίωση του GAIN, τις οποίες έχουμε ονοματίσει GAIN-TD, GAIN-CNN, GAIN-FAST και GAIN-DEEP.

Βάση των πειραμάτων μας, παρατηρήσαμε πως στα δεδομένα εικόνων οι αρχιτεκτονικές που λειτούργησαν καλύτερα ήταν τα GAIN-DEEP και GAIN-CNN. Στη συνέχεια για το σύνολο δεδομένων MNIST ακολουθεί το GAIN-TD, το GAIN-FAST και τη χειρότερη απόδοση είχε το GAIN. Επιπλέον, οι αρχιτεκτονικές με τους λιγότερους κρυφούς νευρώνες επιτρέπουν στη μεθοδολογία του GAIN να επεκταθεί και σε εικόνες με μεγαλύτερες διαστάσεις. Επίσης, να σημειώσουμε και πάλι ότι η αρχιτεκτονική GAIN-TD έβγαλε καλύτερα αποτελέσματα από το GAIN στα σύνολα δεδομένων της έρευνας των Yoon et al., καθώς παρουσιάζε και στα πέντε datasets μικρότερο RMSE.

Σε μελλοντική έρευνα θα μπορούσε να δοκιμαστεί μια αρχιτεκτονική που θα συνδυάζει το GAIN με τα Multi Generator GANS [30], αφότου προηγηθεί διαχωρισμός μεταβλητών (Variable Split) στο σύνολο δεδομένων σε κατηγορικές ή αριθμητικές μεταβλητές. Με αυτόν τον τρόπο θα μπορούσε να διερευνηθεί πώς οι generators μαθαίνουν τις κατανομές όταν έχουν να κάνουν αποκλειστικά με ένα είδος μεταβλητής. Με όλες τις αρχιτεκτονικές που ήδη δοκιμάσαμε, αλλά και με αυτήν που προτείναμε μόλις, η ύπαρξη παραπάνω υπολογιστικής δύναμης θα μας επέτρεπε να πειραματιστούμε περισσότερο πάνω σε σύνολα δεδομένων μεγάλων διαστάσεων και όγκου. Ακόμα, θα μπορούσε να διερευνηθεί περισσότερο η σχέση μεταξύ των loss functions και του hint vector, δεδομένων των αποτελεσμάτων που πήραμε με την αρχιτεκτονική GAIN-TD που εφαρμόσαμε. Αυτό θα έδινε πληροφορία για τη σχέση των δύο δικτύων, generator και discriminator, και κατά πως η ρύθμιση διαφόρων παραμέτρων που αφορούν το hint vector και το missing rate, την υπερπαράμετρο α και άλλων, συντελούν στην εκπαίδευσή τους.

Συνεισφορές

Τα πειράματα έτρεξαν στην εθνική υπερ-υπολογιστική υποδομή ARIS, η οποία είναι εγκατεστημένη στο κτίριο του Υπουργείου Παιδείας, Έρευνας και Θρησκευμάτων στο Μαρούσι. Έπίσης, έγινε χρήση του λογαριασμού του εργαστηρίου AILS (Artificial Intelligence and Learning Systems) για την πρόσβαση στην παραπάνω υποδομή. Κάθε μέλος της ομάδας συνεισέφερε τόσο στην ανάπτυξη του κώδικα όσο και στη συγγραφή του paper. Ο κώδικας για την αναπαραγωγή με όλα τα αποτελέσματα βρίσκεται στο github και είναι διαθέσιμος για βελτίωση και τροποποίησεις.

Αναφορές

[1] J. Yoon, J. Jordon, and M. van der Schaar, "GAIN: Missing Data Imputation using Generative Adversarial Nets", *CoRR*, vol. abs/1806.02920, 2018. arXiv: 1806.02920. [Online]. Available: http://arxiv.org/abs/1806.02920.

[2] T. D. Pigott, "A review of methods for missing data", *Educational research and evaluation*, vol. 7, no. 4, pp. 353–383, 2001.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets", in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[4] M. Hopkins, E. Reeber, G. Forman, and J. Suermondt, *UCI Machine Learning Repository*, 1999. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/spambase.

[5] ——, *UCI Machine Learning Repository*, 1991. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Letter+Recognition.

[6] C. Yeh and C.-h. Lie, *UCI Machine Learning Repository*, 2009. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients?fbclid=IwAR24zSxaMV6-JzPETXxGPmG3G7BQ_FIJ5jgCFK3DUyKSQ2ZdH1NXr64dDG4#.

[7] K. Fernandes, P. Vinagre, and P. Cortez, *UCI Machine Learning Repository*, 2015. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/online+news+popularity?fbclid=IwAR25BlNFWm5f3qPG8-9CahbrYwxl5zFuznD6sLa-8FJl6G2bWP-a17qB2qY.

[8] W. Wolberg, N. Street, and O. Mangasarian, *UCI Machine Learning Repository*, 1995. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic).

[9] [Online]. Available: http://yann.lecun.com/exdb/mnist/.

[10] D. Kermany, K. Zhang, and M. Goldbaum, *Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification*, 2018. [Online]. Available: http://dx.doi.org/10.17632/rscbjbr9sj.2.

[11] R. Camino, C. Hammerschmidt, and R. State, "Improving Missing Data Imputation with Deep Generative Models", Feb. 2019.

[12] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for DNA microarrays", *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.

[13] U. Jain, Z. Zhang, and A. G. Schwing, "Creativity: Generating diverse questions using variational autoencoders", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6485–6494.

[14] S. van Buuren and K. Groothuis-Oudshoorn, "MICE: Multivariate Imputation by Chained Equations in R Journal of Statistical Software, forthcoming, 2009", *URL http://CRAN. R-project. org/package mice*,

[15] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices", *The Journal of Machine Learning Research*, vol. 11, pp. 2287–2322, 2010.

[16] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: a review", *Neural Computing and Applications*, vol. 19, no. 2, pp. 263–282, 2010.

[17] A. Costa, M. Santos, J. Soares, and P. Henriques Abreu, "Missing Data Imputation via Denoising Autoencoders: The Untold Story: 17th International Symposium, IDA 2018, 's-Hertogenbosch, The Netherlands, October 24–26, 2018, Proceedings", in. Jan. 2018, pp. 87–98, ISBN: 978-3-030-01767-5. DOI: 10.1007/978-3-030-01768-2_8.

[18] S. C. Li, B. Jiang, and B. M. Marlin, "MisGAN: Learning from Incomplete Data with Generative Adversarial Networks", *CoRR*, vol. abs/1902.09599, 2019. arXiv: 1902.09599. [Online]. Available: http://arxiv.org/abs/1902.09599.

[19] C. Shang, A. Palmer, J. Sun, K. Chen, J. Lu, and J. Bi, "VIGAN: Missing view imputation with generative adversarial networks", in *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 766–775.

[20] D. Lee, J. Kim, W.-J. Moon, and J. C. Ye, "CollaGAN: Collaborative GAN for missing image data imputation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2487–2496.

[21] G. Bradski, "The OpenCV Library", *Dr. Dobb's Journal of Software Tools*, 2000.

[22] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009, ISBN: 1441412697.

[23] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from tensorflow.org, 2015. [Online]. Available: https://www.tensorflow.org/.

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library", in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[25] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: a structure for efficient numerical computation", *Computing in Science & Engineering*, vol. 13, no. 2, p. 22, 2011.

[26] 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.

[27] D. Charte, F. Charte, M. Jesus, and F. Herrera, *A Showcase of the Use of Autoencoders in Feature Learning Applications*, May 2020.

[28] A. Radhakrishnan, M. Belkin, and C. Uhler, *Downsampling leads to Image Memorization in Convolutional Autoencoders*, Oct. 2018.

[29] B. Hou and R. Yan, "Convolutional Auto-Encoder Based Deep Feature Learning for Finger-Vein Verification", Jun. 2018, pp. 1–5. DOI: 10.1109/MeMeA.2018.8438719.

[30] Q. Hoang, T. Nguyen, T. Le, and D. Phung, "Multi-Generator Gernerative Adversarial Nets", Aug. 2017.