ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΠΜΣ ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΥΠΟΛΟΓΙΣΤΩΝ

Μεταπτυχιακή εργασία

Δοκιμές αξιολόγησης συστημάτων ML:

Διαδικασίες, Μέθοδοι & Μετρικές

ΣΟΥΡΛΑΣ ΒΑΣΙΛΕΙΟΣ

Α.Μ.: 2022202002023

Τρίπολη | Μάιος 2024

Επιβλέπων Καθηγητής: Κώστας Βασιλάκης

Περίληψη

Η δοκιμή αξιολόγησης των συστημάτων ML είναι μια απαιτητική και δαπανηρή διαδικασία καθώς τα συστήματα αυτά παρουσιάζουν υψηλή πολυπλοκότητα, χρειάζονται μεγάλο όγκο δεδομένων για να πραγματοποιήσουν καλές προβλέψεις, και χρησιμοποιούν πολύπλοκους, πιθανολογικούς μη ντετερμινιστικούς αλγόριθμους. Η παρούσα διπλωματική εργασία επιδιώκει να δώσει μια ολοκληρωμένη εικόνα των σύγχρονων μεθοδολογιών που χρησιμοποιούνται για τις δοκιμές αξιολόγησης των συστημάτων ML, συμβάλλοντας στη βελτίωση της αξιοπιστίας και της απόδοσης τους.

Αρχικά, αναλύεται η διαδικασία ανάπτυξης ενός συστήματος ML, από τη συλλογή και την προεπεξεργασία των δεδομένων, έως και την παρακολούθηση της συμπεριφοράς και τρόπου λειτουργίας του συστήματος σε παραγωγικό περιβάλλον. Στη συνέχεια, αναλύεται η διαδικασία αξιολόγησης των συστημάτων AI, ενώ επίσης επισημαίνονται οι βασικές διαφορές των δοκιμών αξιολόγησης τους σε σύγκριση με τα συμβατικά συστήματα. Ιδιαίτερη έμφαση δίνεται στις μετρικές αξιολόγησης που χρησιμοποιούνται για την μέτρηση της απόδοσης των μοντέλων ML και περιγράφονται τα μειονεκτήματα και τα πλεονεκτήματα τους. Ακόμα, παρουσιάζονται οι μεθοδολογίες δοκιμών, όπως οι Differential, Metamorphic, Mutation και Combinatorial, οι οποίες βοηθούν τόσο στη δημιουργία δοκιμαστικών περιπτώσεων όσο και στην αντιμετώπιση του προβλήματος του test oracle. Τέλος, συζητούνται η επάρκεια των δοκιμών και οι τεχνικές ιεράρχησης και μείωσης των δοκιμαστικών περιπτώσεων για την αποδοτικότερη και αποτελεσματικότερη αξιολόγηση των ML συστημάτων.

Λέξεις-κλειδιά: δοκιμές αξιολόγησης, μηχανική μάθηση, τεχνητή νοημοσύνη, βαθεία μάθηση, μοντέλα, νευρωνικά δίκτυα, ακρίβεια, πληρότητα, κάλυψη

Abstract

Testing of ML systems is a demanding and expensive process, given that these systems exhibit high complexity, need a large amount of data to make good predictions, and use complex, probabilistic non-deterministic algorithms. This thesis seeks to present a comprehensive picture of the state-of-the-art methodologies used for the evaluation tests of ML systems, striving to improve their reliability and performance.

Initially, the development process of an ML system is analyzed, from the stages of collection and pre-processing of the data, to the phase of monitoring of the system's behavior and operation in a production environment. Then, the testing process of AI systems is analyzed, and the key differences between testing of AI systems conventional systems are highlighted. Special emphasis is given on the evaluation metrics used to measure the performance of ML models and their disadvantages and advantages are described. Furthermore, testing methodologies, such as Differential, Metamorphic, Mutation and Combinatorial, are presented, which help both in creating test cases and in dealing with the test oracle problem. Finally, test adequacy and test case prioritization and reduction techniques are discussed for more efficient and effective evaluation of ML systems.

Keywords: testing, machine learning, artificial intelligence, deep learning, models, neural networks, precision, adequacy, coverage

Πίνακας περιεχομένων

Πίνακας Εικόνων

1 Εισαγωγή

Η μηχανική μάθηση (ML) και η βαθιά μηχανική μάθηση (DL) χρησιμοποιείται όλο και περισσότερο σε κρίσιμα υπολογιστικά συστήματα, τα οποία επηρεάζουν άμεσα ή έμμεσα την καθημερινή μας ζωή. Παραδείγματα χρήσης των προσεγγίσεων αυτών μάθησης είναι:

α) σε Συστήματα Υγείας, για τη διάγνωση ασθενειών μέσω των ιατρικών εικόνων και την πρόβλεψη των αναγκών για πόρους σε νοσοκομεία,

β) σε Υποδομές Μεταφορών, για τον αυτόματο έλεγχο και τη διαχείριση της κίνησης, την πρόβλεψη της κυκλοφορίας, τη βελτιστοποίηση των δρομολογίων, και την ανάλυση των δεδομένων ασφάλειας,

γ) σε Ενεργειακές Υποδομές, για την πρόβλεψη της ζήτησης ενέργειας, την αυτόματη ρύθμιση της κατανομής ενέργειας, και τη βελτιστοποίηση των ενεργειακών πόρων,

δ) σε Χρηματοοικονομικές Υποδομές, για τον αυτόματο χειρισμό των χρηματοοικονομικών δεδομένων, την ανίχνευση απάτης, την πρόβλεψη των τιμών μετοχών και των αγορών, και για άλλες χρηματοοικονομικές αναλύσεις,

ε) στην Πρόβλεψη Καταστροφών, όπως σεισμοί, πλημμύρες ή πυρκαγιές, προκειμένου να ληφθούν προληπτικά μέτρα κ.ά.

Τα συστήματα αυτά, παρουσιάζουν αυξημένη πολυπλοκότητα, απαιτούν υψηλές υπολογιστικές δυνατότητες, χρειάζονται μεγάλο όγκο δεδομένων για να κάνουν εύστοχες προβλέψεις, και χρησιμοποιούν πολύπλοκους, πιθανοτικούς, μη ντετερμινιστικούς αλγόριθμους. Όπως γίνεται αντιληπτό, η διασφάλιση της ποιότητας αυτών των συστημάτων λογισμικού αποτελεί μια ανοιχτή πρόκληση για την ερευνητική κοινότητα. Ως εκ του, οι δοκιμές λογισμικού (testing) τέτοιου είδους συστημάτων είναι μια ιδιαίτερα απαιτητική και δαπανηρή διαδικασία.

Η ελλιπής ή πλημμελής αξιολόγηση των συστημάτων αυτών, μπορεί να δημιουργήσει σημαντικά προβλήματα, όπως θα μπορούσε να γίνει αντιληπτό από τις ενδεικτικές περιοχές εφαρμογών που αναφέρονται ανωτέρω. Μερικά χαρακτηριστικά παραδείγματα συνεπειών της εσφαλμένης λειτουργίας του λογισμικού είναι τα ακόλουθα: Το 2016, ένα αυτόνομο αυτοκίνητο της Google τράκαρε καθώς προσπαθούσε να αποφύγει σάκους άμμου, ενώ ένα αυτοκίνητο της Tesla συντρίφτηκε μη μπορώντας να αναγνωρίσει την καρότσα του προπορευόμενου οχήματος ως

εμπόδιο. Το 2017, ένας Παλαιστίνιος συνελήφθη όταν δημοσίευσε στο Facebook μια "καλημέρα" η οποία μεταφράστηκε λανθασμένα ως "επιτεθείτε τους". Τέτοιου είδους περιστατικά αναδεικνύουν πόσο σημαντικές είναι η δοκιμές αξιολόγησής των συστημάτων ML προτού γίνουν διαθέσιμα στο ευρύ κοινό.

1.1 Κατηγορίες μηχανικής μάθησης

Για την ανάπτυξη των συστημάτων μηχανικής μάθησης χρησιμοποιούνται αλγόριθμοι, οι οποίοι μπορούν να ταξινομηθούν σε τρείς βασικές κατηγορίες: α) στην επιβλεπόμενη μάθηση (Supervised learning) β) στην μη-επιβλεπόμενη μάθηση (Unsupervised learning) και γ) στην ενισχυτική μάθηση (Reinforcement learning). Οι κατηγορίες αυτές περιγράφονται στις ακόλουθες παραγράφους.

1.1.1 Επιβλεπόμενη μάθηση

Κατά την Επιβλεπόμενη μάθηση, ο αλγόριθμος δημιουργεί το μοντέλο ML κάνοντας χρήση ενός συνόλου επισημασμένων δεδομένων κατά τη φάση της εκπαίδευσης. Τα επισημασμένα δεδομένα υποδηλώνουν ότι κάθε σημείο εισόδου σχετίζεται με την αντίστοιχη ετικέτα (label) εξόδου. Πιο συγκεκριμένα, τα ζεύγη εισόδων (π.χ. η εικόνα μίας γάτας και η αντίστοιχη ετικέτα της – «γάτα») χρησιμοποιούνται από τον αλγόριθμο ώστε να συμπεράνει (infer) τη σχέση μεταξύ των δεδομένων εισόδου (π.χ. εικόνες από γάτες) και των ετικετών εξόδου (π.χ. «γάτα» και «σκύλος»).

Υπάρχουν δύο τύποι αλγορίθμων επιβλεπόμενης μηχανικής μάθησης:

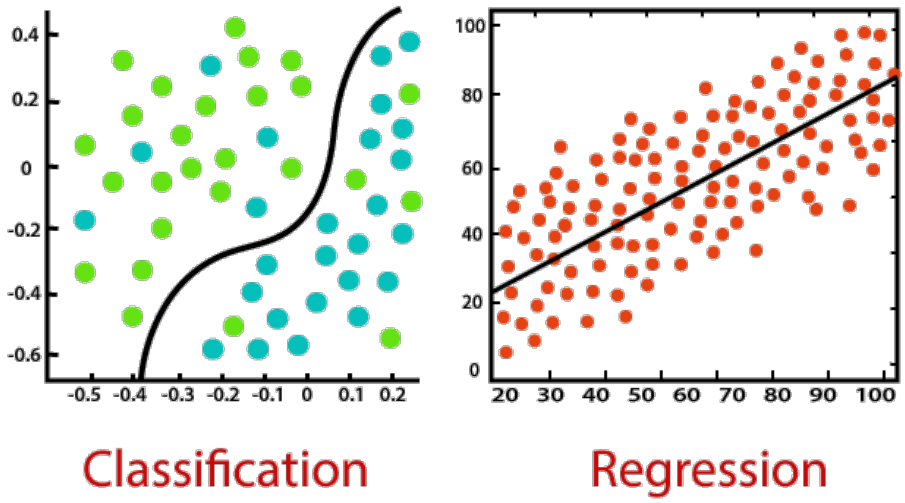
1. **Ταξινόμηση (Classification)** – χρησιμοποιείται όταν τα δεδομένα εισόδου πρέπει να ταξινομηθούν σε μια ή περισσότερες προκαθορισμένες κλάσεις (Riccio, et al., 2020). Χαρακτηριστικό παράδειγμα είναι η αναγνώριση αντικειμένων ή αναγνώριση προσώπων σε εικόνες. Υπάρχει πληθώρα αλγορίθμων που μπορούν να χρησιμοποιηθούν για την εκτέλεση της ταξινόμησης σε περιβάλλον επιβλεπόμενης μηχανικής μάθησης και οι πιο ευρέως διαδεδομένοι είναι οι ακόλουθοι:
    - Δένδρα αποφάσεων για Ταξινόμηση (Decision Tree Classification)
    - K- πλησιέστερου γείτονα (K-Nearest Neighbours)
    - Λογιστική παλινδρόμηση (Logistic Regression)
    - Απλοϊκή μέθοδος Bayes (Naïve Bayes)
    - Νευρωνικά δίκτυα (Neural Networks)

- Τυχαίος ταξινομητής δασών (Random Forest Classification)
- Μηχανές διανυσμάτων υποστήριξης (Support Vector Machines -SVM)

2. **Παλινδρόμηση (Regression) –** χρησιμοποιείται όταν το μοντέλο ML πρέπει να προβλέψει συνεχείς τιμές. Η βασική αρχή της παλινδρόμησης είναι η σύνδεση μίας ή περισσότερων ανεξάρτητων μεταβλητών με μια εξαρτημένη μεταβλητή, με τη χρήση της κατάλληλης γραμμικής συχνότητας (Riccio, et al., 2020). Χαρακτηριστικό παράδειγμα είναι η πρόβλεψη των μελλοντικών τιμών των μετοχών ή των ακινήτων, οι προβλέψεις εσόδων από πωλήσεις για μια επιχείρηση, καθώς και η πρόβλεψη της ηλικίας ενός ατόμου με βάση τις συνήθειες του.

   Και σε αυτή την περίπτωση υπάρχουν πολλαπλοί αλγόριθμοι για την υλοποίηση της προσέγγισης, με πιο διαδεδομένους τους εξής:

   - Δένδρα αποφάσεων για παλινδρόμηση (Decision Tree Regression)
   - Πολλαπλή γραμμική παλινδρόμηση (Multiple Linear Regression)
   - Νευρωνικά δίκτυα (Neural Networks, [1.2])
   - Πολυωνυμική παλινδρόμηση (Polynomial Regression)
   - Τυχαίος ταξινομητής δασών για παλινδρόμηση (Random Forest Regression)
   - Απλή γραμμική παλινδρόμηση (Simple Linear Regression)
   - Support Vector Regression



Εικόνα 1: Αριστερά απεικονίζεται επιβλεπόμενη μάθηση με Ταξινόμηση ενώ δεξιά με Παλινδρόμηση

1.1.2 Μη-επιβλεπόμενη μάθηση

Στη *μη-επιβλεπόμενη μάθηση*, ο αλγόριθμος δημιουργεί το μοντέλο ML από ένα σύνολο δεδομένων τα οποία δεν είναι επισημασμένα ούτε ταξινομημένα, κατά τη φάση της εκπαίδευσης. Τα δεδομένα αυτά χρησιμοποιούνται από τον αλγόριθμο για να εντοπίσει μοτίβα (patterns) στα δεδομένα εισόδου (Riccio, et al., 2020). Πιο συγκεκριμένα, διερευνά την ύπαρξη ή μη κλάσεων, και ποσοτικοποιεί τις τιμές των χαρακτηριστικών των υποσυνόλων (ως προς το συνόλου των προτύπων) που ανήκουν σε μια κλάση. Συνεπώς, η μη-εποπτευόμενη μάθηση μελετά πώς το σύστημα μπορεί να εντοπίσει αυτόματα μια συνάρτηση η οποία περιγράφει τις κρυφές δομές των μη επισημασμένων δεδομένων. Υπάρχουν δύο τύποι αλγορίθμων μη-επιβλεπόμενης μηχανικής μάθησης:

1. **Συσταδοποίησης/ Ομαδοποίησης (Clustering)**: χρησιμοποιείται για την ομαδοποίηση των δεδομένων που έχουν τα κοινά χαρακτηριστικά (characteristics) ή και ιδιότητες (attributes). Για τη λειτουργία του μοντέλου δεν είναι προαπαιτούμενο να είναι γνωστός ο αριθμός και το είδος των κατηγοριών. Για παράδειγμα, η συσταδοποίηση χρησιμοποιείται στο μάρκετινγκ, για την ομαδοποίηση διαφορετικών τύπων πελατών.

   Οι αλγόριθμοι που χρησιμοποιούνται σε αυτόν τον τύπο μη-επιβλεπόμενης μάθησης είναι:
   - K-means clustering algorithm
   - DBSCAN clustering algorithm (Density-based spatial clustering of applications with noise)
   - Gaussian Mixture Model algorithm
   - BIRCH algorithm (Balance Iterative Reducing and Clustering using Hierarchies)
   - Affinity Propagation clustering algorithm
   - Mean-Shift clustering algorithm
   - OPTICS algorithm (Ordering Points to Identify the Clustering Structure)
   - Agglomerative Hierarchy clustering algorithm
2. **Συσχέτισης (Association):** χρησιμοποιείται για τον εντοπισμό συσχετίσεων ή και εξαρτήσεων μεταξύ των ιδιοτήτων των δεδομένων. Επίσης, μέσω της συσχέτισης, το μοντέλο ML μπορεί να εξαγάγει πληροφορίες οι οποίες εμφανίζονται με την πάροδο του χρόνου (Sarker & Iqbal, 2021). Δεν θα πρέπει να παραλειφθεί, ότι ενίοτε χρησιμοποιείται και για να απεικονίσει συσχετίσεις ανάμεσα στους προγόνους (ancestors) και στους απογόνους

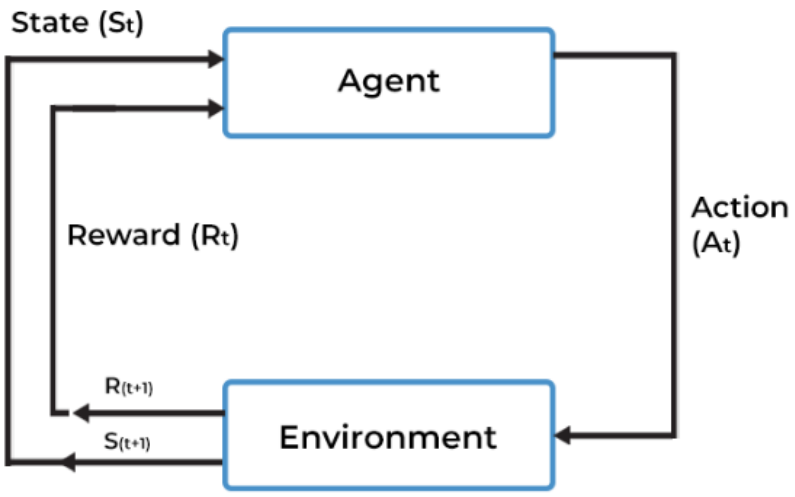
(descendants). Χαρακτηριστικό παράδειγμα εφαρμογής της συσχέτισης αποτελεί η χρήση της σε συστήματα σύστασης προϊόντων (recommender systems), όπου το μοντέλο εντοπίζει πιθανές συσχετίσεις μεταξύ των καταναλωτών και της αγοραστικής τους συμπεριφοράς. Οι αλγόριθμοι που χρησιμοποιούνται σε αυτόν τον τύπο μη-επιβλεπόμενης μάθησης είναι:

- Apriori
- Eclat (Equivalence Class Transformation)
- FP-Growth (Frequent Pattern-Growth)
- FPMax
- COBWEB (Categorization and Clustering Without Overlapping Boundaries)
- CARTE (Class Association Rule and Emerging Pattern Mining)
- Tertius

1.1.3 Ενισχυτική μάθηση (Reinforncemet Learning)

Η Ενισχυτική μάθηση διαφέρει ως προς τις προηγούμενες προσεγγίσεις καθώς για την εκπαίδευση του μοντέλου δεν απαιτούνται δεδομένα εισόδου και εξόδου. Η ενισχυτική μάθηση είναι μια μέθοδος όπου το σύστημα (ένας πράκτορας μάθησης - agent) μαθαίνει μέσω της αλληλεπίδρασης του με το περιβάλλον. Όπως φαίνεται στην *Εικόνα 2*, ο πράκτορας αλληλεπιδρά συνεχώς με το περιβάλλον λαμβάνοντας κάθε φορά μια ανταμοιβή (reward) για την εκτέλεση μιας ενέργειας (action). Το περιβάλλον, αποκρινόμενο σε αυτές τις ενέργειες, παρουσιάζει συνεχώς καινούριες καταστάσεις (states), καθώς επίσης ανταμείβει τον πράκτορα όταν παίρνει τις σωστές αποφάσεις ενώ τον «τιμωρεί» όταν οι ενέργειες δεν είναι ορθές. Επομένως, η θετική ενίσχυση του πράκτορα αυξάνει τη συχνότητα μιας συμπεριφοράς ενώ η αρνητική ενίσχυση την μειώνει. Επιπλέον, θα πρέπει να τονιστεί ότι ο πράκτορας δεν καθοδηγείται από κάποιο εξωτερικό παράγοντα, αλλά ανακαλύπτει μόνος του ποιες ενέργειες του αποφέρουν το μεγαλύτερο κέρδος.

Η δημιουργία του περιβάλλοντος, ο σχεδιασμός μιας συνάρτησης ανταμοιβής, και η επιλογή της καταλληλότερης στρατηγικής ώστε ο πράκτορας να επιτύχει τον επιθυμητό αποτέλεσμα αποτελούν βασικές προκλήσεις κατά την εφαρμογή της ενισχυτικής μάθησης. Η ενισχυτική μάθηση βρίσκει εφαρμογές, στην ρομποτική, στα αυτόνομα οχήματα, στα παιχνίδια κ.ά.



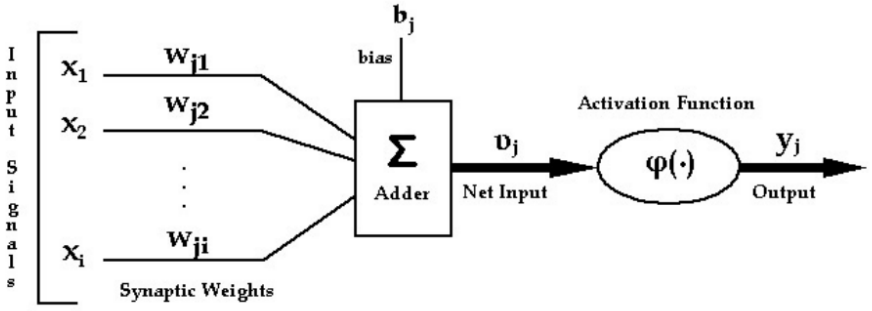
Εικόνα 2: Ενισχυτική μάθηση

1.2 Τεχνητά Νευρωνικά Δίκτυα

Τα Τεχνητά νευρωνικά δίκτυα (ΤΝΔ, Artificial Neural Networks), είναι αλγόριθμοι οι οποίοι χρησιμοποιούν υπολογιστικά συστήματα για να εκτελούν μαθηματικές πράξεις. Η αρχιτεκτονική τους είναι εμπνευσμένη από τη δομή και την λειτουργία των νευρώνων του ανθρώπινου εγκεφάλου. Πιο συγκεκριμένα, τα ανθρώπινα εγκεφαλικά κύτταρα - οι νευρώνες, σχηματίζουν ένα πολύπλοκο διασυνδεδεμένο δίκτυο μέσω του οποίου στέλνουν ηλεκτρικά σήματα ο ένας προς τον άλλο, προκυμμένου να μπορέσουν να επεξεργαστούν τις πληροφορίες που λαμβάνουν. Με τον ίδιο τρόπο τα ΤΝΔ, είναι φτιαγμένα από έναν αριθμό τεχνητών νευρώνων, οι οποίοι δουλεύουν όλοι μαζί, ώστε συνεργαζόμενοι να επιλύσουν ένα πρόβλημα.

Επομένως, κάθε νευρώνας δέχεται ως είσοδο δεδομένα (π.χ. τιμές των pixel), τα επεξεργάζεται και παράγει μια τιμή εξόδου. Οι είσοδοι μπορούν να διακριθούν, στο πρωταρχικό σήμα εισόδου του δικτύου καθώς και στις εξόδους από άλλους νευρώνες.

Έχουν αναπτυχθεί διάφοροι τύποι νευρώνων (π.χ. μη γραμμικοί, στοχαστικοί). Οι τύπος νευρώνα που επιλέγεται για να δομήσει ένα ΤΝΔ εξαρτάται από το είδος του εκάστοτε προβλήματος. Πολλές φορές συνδυάζονται παραπάνω από ένας τύποι, αλλά σε αρκετές περιπτώσεις χρησιμοποιείται ο μη-γραμμικός νευρώνας (Εικόνα 1*Εικόνα 3)*.



Εικόνα 3: Δομή μη-γραμμικού νευρώνα

Σε αυτόν το τύπο νευρώνα, οι πληροφορίες διαχέονται από την είσοδο (αριστερά στην Εικόνα 3) προς την έξοδο (δεξιά στην Εικόνα 3). Αρχικά, το σήμα εισόδου ($x_1, x_2, \dots x_i$) πολλαπλασιάζεται με το συναπτικό του βάρος ($w_{j1}, w_{j1}, \dots w_{ji}$). Το βάρος ουσιαστικά υποδηλώνει πόσο σημαντική είναι η συνεισφορά του εκάστοτε σήματος στη διαμόρφωση της δομής του δικτύου για τους διασυνδεδεμένους νευρώνες. Η συνεισφορά του σήματος είναι μεγάλη όταν το βάρος είναι μεγάλο, και το αντίστροφο. Το βάρος λαμβάνει συνήθως τιμές από -1 έως 1.

Στην συνέχεια, οι σταθμισμένες είσοδοι και ένας εξωτερικός παράγοντας – η *πόλωση* ή αλλιώς *κατώφλι* (b_j, bias) προστίθενται. Το άθροισμα αυτών ονομάζεται *τοπικό πεδίο* ($\upsilon_j, net\ input$) και δίδεται από τον τύπο:

$$\upsilon_j = \sum_{i=1}^{m} W_{ji}\, X_i$$

Τέλος, από την εφαρμογή της συνάρτησης ενεργοποίησης (activation function) στο τοπικό πεδίο, προκύπτει η έξοδος του νευρώνα ($y_j, output$).

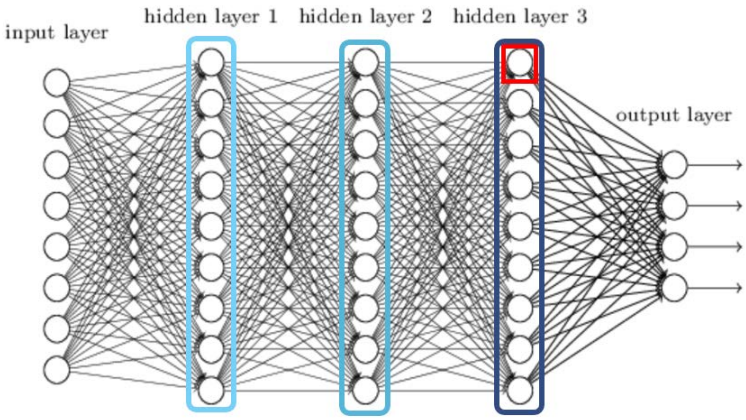
Οι πιο γνωστές μη γραμμικές συναρτήσεις για την υλοποίηση της συνάρτησης ενεργοποίησης είναι οι ακόλουθες:

- Σιγμοειδής συνάρτηση (sigmoid function): $\varphi(\chi) = \frac{1}{1+e^{-x}}$
- Συνάρτηση Softmax: $\varphi(\chi) = \sigma(\chi)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}$
- Συνάρτηση ReLu: $\varphi(\chi) = \max(x, 0)$
- Συνάρτηση υπερβολικής εφαπτομένης (tanh function): $\varphi(\chi) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$

Η χρήση των ανωτέρω συναρτήσεων ενεργοποίησης κάνει τους νευρώνες μη-γραμμικούς, και κατ' επέκταση αυτή η μη-γραμμικότητα μεταφέρεται και στα ΤΝΔ. Η μη-γραμμικότητα αποτελεί χαρακτηριστικό πλεονέκτημα των ΤΝΔ εν συγκρίσει με άλλες γνωστές μεθόδους, καθώς με αυτό τον τρόπο έχουν τη δυνατότητα να εντοπίσουν σύνθετα μοτίβα, να εξάγουν σχέσεις μεταξύ των δεδομένων, και να κάνουν προβλέψεις σε προβλήματα που παρουσιάζουν χαοτική συμπεριφορά. Σε αυτές τις περιπτώσεις, τα γραμμικά μοντέλα αδυνατούν να δώσουν την σωστή απάντηση.

Η επίλυση αυτών των χαοτικών προβλημάτων, δεν είναι δυνατή με χρήση ΤΝΔ που αποτελούνται από ένα μόνο νευρώνα, για το λόγο αυτό γίνεται χρήση πολλαπλών νευρώνων διατεταγμένων σε επίπεδα-στρώματα. Τα ΤΝΔ που αποτελούνται από περισσότερα από δύο στρώματα ονομάζονται *Βαθιά Νευρωνικά Δίκτυα* (Deep Neural Networks - ΒΝΔ). Τα επίπεδα των νευρώνων διακρίνονται σε τρείς κατηγορίες, όπως φαίνεται και στην *Εικόνα 4*.

- Στο επίπεδο εισόδου (Input layer).
- Στα ενδιάμεσα ή κρυφά επίπεδα (Hidden layers).
- Στο επίπεδο εξόδου (Output layer).



Εικόνα 4: Αρχιτεκτονική ένας Νευρωνικού Δικτύου

Τα ΒΝΔ λαμβάνουν τα δεδομένα από το επίπεδο εισόδου και, μέσω των διαθέσιμων κόμβων, τα αναλύουν και τα μεταβιβάζουν στο επόμενο επίπεδο. Στη συνέχεια, το κάθε κρυφό επίπεδο, επεξεργάζεται περαιτέρω την έξοδο του προηγούμενου επιπέδου και την μεταφέρει στο επόμενο. Σε ένα νευρωνικό δίκτυο, όταν κάθε νευρώνας συνδέεται με όλους τους νευρώνες που βρίσκονται στο προηγούμενο και στο επόμενο επίπεδο, τότε το νευρωνικό δίκτυο ονομάζεται πλήρως συνδεδεμένο (fully connected), ειδάλλως ονομάζεται μερικώς συνδεδεμένο (partially connected).

Με την σειρά του, το επίπεδό εξόδου, που αποτελεί το τελευταίο στρώμα του δικτύου, δίνει το αποτέλεσμα ένας επεξεργασίας των δεδομένων.

Θεωρητικά, τα ΒΝΔ μπορούν να αντιστοιχίσουν οποιονδήποτε τύπο εισόδου με οποιαδήποτε έξοδο, αλλά ο όγκος των δεδομένων που απαιτούνται για την εκπαίδευση ένας είναι ένας τάξεως των μερικών εκατομμυρίων, σε αντίθεση με ένας μεθόδους μηχανικής μάθησης όπου συνήθως δεν υπερβαίνει ένας ένας χιλιάδες.

Θα πρέπει να αναφερθεί ότι κατά τη φάση ένας εκπαίδευσης, ένα μοντέλο είναι πιθανό να εμφανίσει φαινόμενα υποπροσαρμογής (underfitting) ή υπερπροσαρμογής (overfitting). Η υποπροσαρμογή σημαίνει όταν το μοντέλο είναι υπερβολικά απλό για να έχει τη δυνατότητα να μάθει την πολυπλοκότητα των δεδομένων εκπαίδευσης, οδηγώντας το σε χαμηλή απόδοση. Αντίστροφα, η υπερπροσαρμογή συμβαίνει όταν το μοντέλο μάθησης μοντελοποιεί υπερβολικά τα δεδομένα εκπαίδευσης συμπεριλαμβανομένου των πιθανών θορύβων. Αυτό συνήθως οδηγεί σε υψηλή απόδοση όταν στην είσοδο εμφανίζονται δεδομένα που έχουν χρησιμοποιηθεί στην εκπαίδευση, αλλά σε χαμηλή απόδοση σε νέα δεδομένα που δεν έχουν χρησιμοποιηθεί κατά την εκπαίδευση.

Τα ΤΝΔ είναι συστήματα παράλληλων κατανεμημένων διεργασιών (parallel distributed processing), διότι οι εργασίες που καλούνται να επιτελέσουν μοιράζονται παράλληλα στα επιμέρους τμήματα του δικτύου, και εκτελούνται από όλους ένας νευρώνες, παρέχοντας έτσι μεγαλύτερες ταχύτητες. Η αρχιτεκτονική των ΤΝΔ διαφέρει από αυτήν των συμβατικών υπολογιστών, καθώς σύμφωνα με τον von Neuman οι κλασικοί υπολογιστές δουλεύουν σειριακά. Ο *Πίνακας* 1 απεικονίζει τις βασικές τους διάφορες.

No.	Νευρωνικά δίκτυα	Συμβατικός Υπολογιστής
1.	Σύγχρονος τρόπος λειτουργίας	Ασύγχρονος τρόπο λειτουργίας
2.	Παράλληλη επεξεργασία	Σειριακή επεξεργασία
3.	Εκπαιδεύονται με παραδείγματα, μεταβάλλοντας τα βάρη των συνδέσεών ένας	Προγραμματίζονται με εντολές (if–then)
4.	Η μνήμη, τα δίκτυα και οι μονάδες λειτουργίας συνυπάρχουν	Διαχωρισμός μνήμης και μονάδων επεξεργασίας ένας πληροφορίας
5.	Ανοχή στα σφάλματα	Καμία ανοχή στα σφάλματα
6.	Ένας–οργάνωση κατά τη διαδικασία ένας εκπαίδευσης	Εξαρτάται εξ ολοκλήρου από την υλοποίηση και κωδικοποίηση του λογισμικού
7.	Η πληροφορία αποθηκεύεται στα βάρη των συνδέσεων	Η πληροφορία αποθηκεύεται σε συγκεκριμένες διευθύνσεις μνήμης

Πίνακας 1: Βασικές διαφορές των ΤΝΔ και των υπολογιστών με την φιλοσοφία του von Neuman

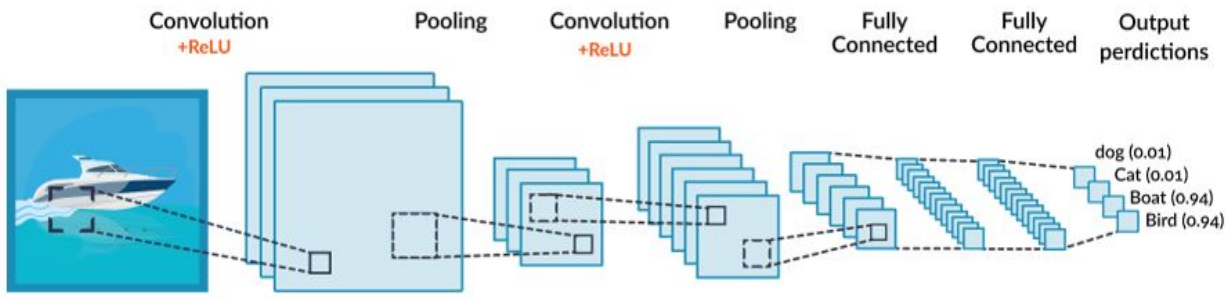
1.2.1 Τύποι Νευρωνικών δικτύων

Ανάλογα με το με το είδος του προβλήματος που ζητείται να επιλυθεί, χρησιμοποιείται διαφορετικός τύπος νευρωνικών δικτύων. Τροποποιώντας την αρχιτεκτονική ένας και αυξάνοντας τον αριθμό των κρυφών στρωμάτων, το μοντέλο μπορεί να εξάγει καλύτερα χαρακτηριστικά από τα δεδομένα εισαγωγής. Ωστόσο, όσο αυξάνεται το βάθος του νευρωνικού δικτύου, τόσο αυξάνεται και η υπολογιστική του πολυπλοκότητα. Στη βιβλιογραφία, τα ΒΝΔ αναφέρονται ως *μαύρα κουτιά* (black boxes), διότι δεν είναι εύκολο να κατανοηθεί από έναν εξωτερικό παρατηρητή (τον χρήστη), πώς το δίκτυο λαμβάνει ένας αποφάσεις του. Στο κεφάλαιο αυτό παρουσιάζονται μερικοί από ένας πιο ευρέως χρησιμοποιούμενους τύπους.

1.2.1.1 Συνελικτικό νευρωνικό δίκτυο (CNN)

Τα συνελικτικό νευρωνικό δίκτυο **(**CNN, Convolutional Neural Network) είναι ένας τύπος δικτύου που έχει αποδειχθεί πολύ αποτελεσματικός για εργασίες ανάλυσης και επεξεργασίας εικόνων, καθώς, ο τρόπος λειτουργίας του επιτρέπει τον σαφή εντοπισμό μοτίβων όπως γραμμές, κλίσεις και κύκλους χωρίς να αγνοεί σημαντικά χαρακτηριστικά, τα οποία θεωρούνται απαραίτητα ώστε το μοντέλο να κάνει σωστές προβλέψεις. Ένα ακόμα μέτρο της αποτελεσματικότητας τους, μπορεί να θεωρηθεί και η ευρεία χρήση τους από μεγάλες εταιρίες για τέτοιου είδους εργασίες.

Όπως φαίνεται και στην *Εικόνα 5*, το CNN αποτελείται συνήθως από το επίπεδο συνέλιξης (convolutional layers), τη μη γραμμική συνάρτηση ενεργοποίησης (Rectified Linear Unit ReLU), το επίπεδο συγκέντρωσης (pooling layer) και το πλήρως συνδεδεμένο επίπεδο (fully connected layer). Πρακτικά, το κάθε κρυφό επίπεδο επεξεργάζεται διαφορετικά χαρακτηριστικά, όπως τις ακμές, το χρώμα, και το βάθος. Τα επίπεδα αυτά εφαρμόζονται μια ή περισσότερες φορές ανάλογα την αρχιτεκτονική του εκάστοτε CNN.



Εικόνα 5: Απεικόνιση ενός απλού CNN, όπου περιλαμβάνει 3 επίπεδα (επίπεδο συνέλιξης, επίπεδο υποδειγματολείψιας, fully connected επίπεδο) ορισμένα από τα οποία επαναλαμβάνονται παραπάνω από μία φορές (Choudhari, 2020)

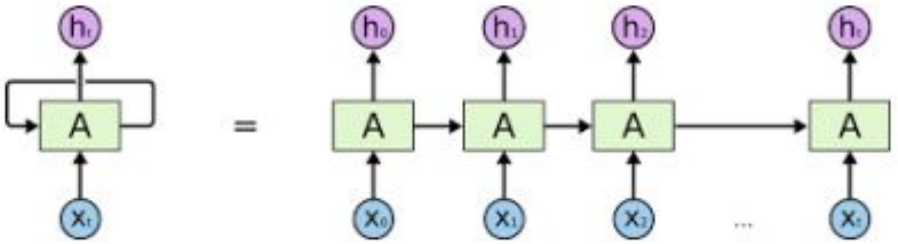
Οι πιο γνωστές αρχιτεκτονικές CNN είναι: Το AlexNet (Alex Krizhevsky, 2012) που αναπτύχθηκε το 2012, το ZFNet που νίκησε στο διαγωνισμό LSVRC το 2013, το GoogleNet δημιουργήθηκε από τη Google το 2014, το VGGNet (Karen Simonyan, 2015) και το ResNet (Kaiming He, 2016) νίκησε στον διαγωνισμό ILSVRC 2015.

1.2.1.2 Αναδρομικό – Ανατροφοδοτούμενο νευρωνικά δίκτυο (RNN)

Τα Ανατροφοδοτούμενα νευρωνικά δίκτυα (RNN - Recurrent neural network) είναι πολύ αποτελεσματικά στην επεξεργασία δεδομένων αλληλουχίας, διότι ο μηχανισμός τους έχει σχεδιαστεί με βάσει την ακολουθιακή μνήμη (sequential memory) του ανθρώπινου εγκεφάλου. Πρακτικά, η ακολουθιακή μνήμη επιτρέπει στον εγκέφαλο να εντοπίζει μοτίβα ακολουθίας ευκολότερα.

Η δομή ενός RNN φαίνεται στην Εικόνα 6. Αναλυτικότερα, τα RNN μπορούν να αντιλαμβάνονται εξαρτήσεις μεταξύ διαφορετικών εισόδων, επειδή η έξοδος κάθε βήματος τους εξαρτάται από την είσοδο των προηγούμενων βημάτων, και όχι μόνο από την είσοδο στο συγκεκριμένο βήμα.

Σε ένα RNN οι πίνακες των βαρών που χρησιμοποιούνται είναι οι ίδιοι σε κάθε βήμα με αποτέλεσμα ο αριθμός των παραμέτρων που καλείται να μάθει το δίκτυο να είναι ανεξάρτητος από το μέγεθος της εισόδου. Άρα το μέγεθος του μοντέλου παραμένει ανεξάρτητο του μήκους των εισόδων, καθιστώντας τέτοιου τύπου ΤΝΔ ικανά να διαχειρίζονται τα δεδομένα εισόδου ως χρονοσειρές.



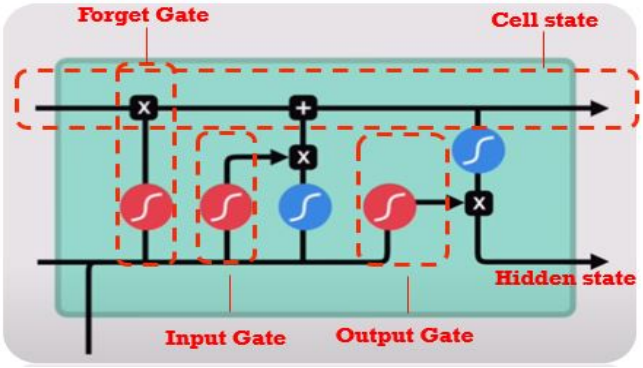
Εικόνα 6: Απεικόνιση δομής ενός RNN (Donges, 2019)

Παρ' όλα αυτά, τα RNN έχουν ορισμένα μειονεκτήματα: Πρώτον, δεν είναι εφικτή η παραλληλοποίηση τους, αφού η κατάσταση του δικτύου σε κάθε βήμα εξαρτάται από τις προηγούμενες καταστάσεις του (σειριακός υπολογισμός). Δεύτερον, εμφανίζουν το πρόβλημα του *vanishing gradient* (εξαφανιζόμενων κλίσεων) που παρουσιάζεται όταν η διαβάθμιση (gradient) λαμβάνει μικρές τιμές που σταδιακά εξαφανίζονται προς τα τελικά βήματα, με συνέπεια τα αποτελέσματα αυτών των βημάτων να μην λαμβάνονται υπόψιν. Αντίστοιχα, το πρόβλημα του exploding gradient (υπερμεγεθυνόμενων κλίσεων) εμφανίζεται όταν οι διαβαθμίσεις παίρνουν πολύ μεγάλες τιμές και γιγαντώνονται όσο διαδίδονται σε επόμενα βήματα, με αποτέλεσμα το δίκτυο να δίνει μεγαλύτερη βαρύτητα στις παλαιότερες καταστάσεις του και να αγνοεί τις νεότερες.

1.2.1.3 Long-Short Term Memory – LSTM

Αν και θεωρητικά τα RNN είναι ικανά να εντοπίζουν εξαρτήσεις μεγάλων αποστάσεων, στην πράξη αυτό επιτυγχάνεται δύσκολα. Για τον λόγο αυτό, προτάθηκε το Long-Short Term Memory, ως εναλλακτική του RNN.

Ένα κλασικό LSTM (Εικόνα 7) δίκτυο αποτελείται από ένα κελί - μνήμη (cell), μια θύρα εισόδου (input gate), μια θύρα εξόδου (output gate) και μία θύρα λήθης (forget gate). Μέσω των θυρών ελέγχεται η ροή των πληροφοριών. Το κελί είναι υπεύθυνο για την παρακολούθηση των εξαρτήσεων μεταξύ των στοιχείων στα δεδομένα εισόδου. Η θύρα εισόδου ελέγχει μέχρι ποιο σημείο η νέα τιμή θα εισχωρήσει στο κελί, ενώ η θύρα λήθης για πόσο θα παραμείνει η τιμή στο κελί. Τέλος, η θύρα εξόδου ελέγχει ποια τιμή από το κελί θα χρησιμοποιηθεί για να υπολογισθεί η έξοδος ενεργοποίησης (Phi, 2018).



Εικόνα 7: Κλασική αρχιτεκτονική ενός LSTM δικτύου (Phi, 2018)

Σε ένα πρόβλημα κατηγοριοποίησης κειμένου, το LSTM δέχεται τις λέξεις του συνόλου δεδομένων στη σειρά ως διάνυσμα. Τις επεξεργάζεται σειριακά κρίνοντας ποια χαρακτηριστικά θεωρούνται σημαντικά και ποια όχι, "αποθηκεύοντας" ή "διαγράφοντας" τα χαρακτηριστικά αυτά από το κελί-μνήμη. Τέλος, το παραγόμενο διάνυσμα είναι η περίληψη όλης της πληροφορίας του εγγράφου. Το LSTM, σε σύγκριση με άλλες αρχιτεκτονικές νευρωνικών δικτύων, θεωρείται κατάλληλο για την ανίχνευση μη εύκολα αντιληπτών συσχετίσεων στο κείμενο.

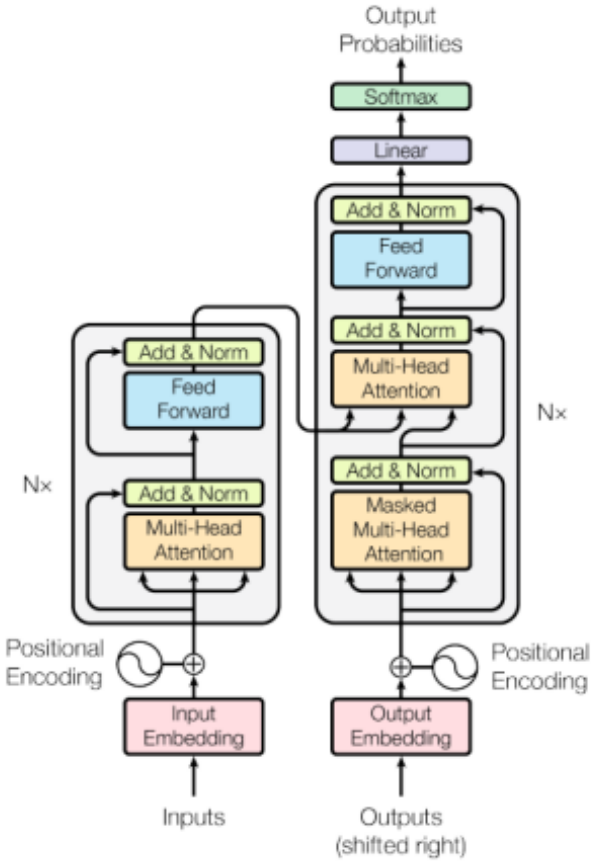
1.2.1.4 Transformer

Το Transformer (*Εικόνα 8*) προτάθηκε το 2017 από ερευνητές της Google και θεωρείται από τις πιο επιδραστικές αρχιτεκτονικές που έχουν επινοηθεί μέχρι σήμερα (Vaswani, et al., 2017). Η αρχιτεκτονική αυτή, παρόλο που απαιτεί λιγότερα δεδομένα για την εκπαίδευση του μοντέλου, του επιτρέπει να κάνει καλύτερες προβλέψεις σε σχέση με τις υπόλοιπες αρχιτεκτονικές. Αναλυτικότερα, για τον υπολογισμό των αναπαραστάσεων εισόδου και εξόδου βασίζεται στον μηχανισμό προσοχής (self-attention), ο οποίος επιτρέπει στο μοντέλο να εντοπίζει εξαρτήσεις μεταξύ των διαφορετικών στοιχείων (ακόμα κι αν είναι μακριά το ένα από το άλλο) καθώς και να αξιολογεί την σημασία τους, βελτιώνοντας επομένως την κατανόηση του περιβάλλοντος του.
Τα κύρια μέρη της αρχιτεκτονικής Trasformers είναι:

- **Στοίβα κωδικοποίησης (Encoder) και αποκωδικοποίησης (Decoder):** Ο κωδικοποιητής λαμβάνει την είσοδο και τη μετατρέπει σε μια σειρά από διανύσματα (vectors) που αντιπροσωπεύουν τη σημασιολογική πληροφορία της εισόδου. Ο αποκωδικοποιητής χρησιμοποιεί αυτά τα διανύσματα για να παράγει την έξοδο.
- **Multi-Head Self-Attention Mechanism:** Αυτός ο μηχανισμός επιτρέπει στο μοντέλο να εξετάζει και να αξιολογεί το κάθε στοιχείο (π.χ.: τη σημασία κάθε λέξης σε σχέση με άλλες

λέξεις στην ίδια πρόταση ταυτόχρονα), με πολλούς διαφορετικούς τρόπους, εξ ου και η χρήση του όρου «multi-head».

- **Positional Encoding:** Χρησιμοποιείται για να δώσει πληροφορίες σχετικά με τη θέση κάθε λέξης στην πρόταση.



Εικόνα 8: Αρχιτεκτονική μοντέλου Trasformers (Vaswani, et al., 2017)

2 Δοκιμές επαλήθευσης και αξιολόγησης συστημάτων ML

Οι δοκιμές επαλήθευσης και αξιολόγησης συστημάτων ML αποτελούν μέρος της διαδικασίας ανάπτυξης των συστημάτων αυτών, όπως συμβαίνει σε κάθε ανάπτυξη συστήματος. Στη συνέχεια περιγράφεται αναλυτικότερα η διαδικασία ανάπτυξης συστημάτων ML, καθώς και οι ιδιαιτερότητές τους σε σχέση με τα λοιπά συστήματα λογισμικού, με έμφαση στα χαρακτηριστικά εκείνα που επηρεάζουν τις δοκιμές επαλήθευσης.

2.1 Διαδικασία ανάπτυξης ενός συστήματος ML

Η διαδικασία ανάπτυξης ενός ML συστήματος περιλαμβάνει τα ακόλουθα στάδια:

1. **Σκοπός της δημιουργίας του μοντέλου AI (Understand the Objectives):** Αρχικά, θα πρέπει να κατανοηθεί από όλα τα εμπλεκόμενα μέρη (π.χ. Μηχανικοί ML, Testers, Προγραμματιστές, Business Analysts κ.λπ.), ποιο πρόβλημα επιχειρείται να λύσει η ανάπτυξη του συστήματος ML. Αφού προσδιοριστεί ο σκοπός του, θα πρέπει να συμφωνηθούν τα ελάχιστα κριτήρια αποδοχής (acceptance criteria) για την ανάπτυξη του, συμπεριλαμβανομένων των μετρικών αξιολόγησης του (πρβλ. Κεφάλαιο 3 «Μετρικές Αξιολόγησης»).
2. **Επιλογή του Framework (Select the framework):** Στη συνέχεια θα πρέπει να γίνει η επιλογή του κατάλληλου framework που θα χρησιμοποιηθεί για την ανάπτυξη του μοντέλου ML.

   Τα τελευταία χρόνια έχουν αναπτυχθεί πολλά frameworks, τα οποία μπορούν να συμβάλουν στις ακόλουθες διαδικασίες: α) την επεξεργασία των δεδομένων, β) την επιλογή του κατάλληλου αλγορίθμου, γ) τη μεταγλώττιση των μοντέλων ώστε να μπορούν εκτελεστούν σε διαφορετικούς τύπους επεξεργαστών (π.χ.: CPUs , GPUs και TPUs[1]).

   Τα πιο γνωστά open-source frameworks είναι: το TensorFlow[2] και το Sciti-learn[3] που αναπτύχθηκαν από την Google, το CNTK[4] που αναπτύχθηκε από την Microsoft, το Apache MxNet[5] που αναπτύχθηκε από τον Carlos Guestrin και το πανεπιστήμιο του Washington

[1] Tensor Processing Units, συνιστώσες που επιταχύνουν τις εργασίες ML.
[2] https://www.tensorflow.org/
[3] https://scikit-learn.org/stable/
[4] https://github.com/microsoft/CNTK
[5] https://mxnet.apache.org/

και χρησιμοποιείται ευρέως από την AWS, το IBM Watson Studio[6] που περιέχει μια σουίτα από εργαλεία που υποστηρίζουν την ανάπτυξη AI συστημάτων, το KERAS[7] που αναπτύχθηκε από ερευνητές στο πλαίσιο της ερευνητικής προσπάθειας του έργου ONEIROS, καθώς και το PyTorch[8] που αναπτύχθηκε από την Facebook.

Επομένως, για την επιλογή του κατάλληλου Framework, θα πρέπει να ληφθούν υπ' όψη ο σκοπός χρήσης του, τα κριτήρια αποδοχής, η γλώσσα προγραμματισμού που θα χρησιμοποιηθεί για την ανάπτυξη του μοντέλου και οι επιχειρηματικές ανάγκες που έχουν τεθεί από τον πελάτη.

3. **Επιλογή και δημιουργία του αλγορίθμου (Select and build the algorithm):** Για την επιλογή του αλγορίθμου ML, των υπερπαραμέτρων (hyperparameters_ του μοντέλου ML καθώς και των ρυθμίσεών του, θα πρέπει να ληφθούν υπόψιν οι ακόλουθοι παράγοντες:
    - Ο σκοπός λειτουργίας του (π.χ.: Ταξινόμηση ή Πρόβλεψη)
    - Τα ελάχιστα απαιτούμενα ποιοτικά του χαρακτηριστικά:
        - Απαιτήσεις στην μνήμη,
        - Ο απαιτούμενος χρόνος εκπαίδευσης – επανεκπαίδευσης του μοντέλου,
        - Η ταχύτητα πρόβλεψης και ο τρόπος παραγωγής των προβλέψεων, π.χ.: μερικά μοντέλα κάνουν προβλέψεις τμηματικά (batch predictions),
        - Πιθανές απαιτήσεις που σχετίζονται με τη Διαφάνεια (Transparency), την Ερμηνευσιμότητα (Interpretability) και τη Δυνατότητα Εξήγησης (Explainability) του μοντέλου.
        - Η ακρίβεια (accuracy) των προβλέψεων.
    - Το είδος των διαθέσιμων δεδομένων, π.χ.: εικόνες, αρχεία κειμένων.
    - Ο όγκος των διαθέσιμων δεδομένων, καθώς μερικά μοντέλα απαιτούν μεγάλο όγκο δεδομένων για την εκπαίδευση τους.
    - Ο αριθμός των χαρακτηριστικών (features) που θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου. Για παράδειγμα η ταχύτητα των προβλέψεων, ο χρόνος

[6] https://www.ibm.com/products/watson-studio
[7] https://keras.io/
[8] https://pytorch.org/

εκπαίδευσης και η ακρίβεια των προβλέψεων εξαρτώνται άμεσά από τον αριθμό των επιλεγμένων χαρακτηριστικών.

- Ο αναμενόμενος αριθμών των κλάσεων κατά τη διαδικασία της συσταδοποίησης, καθώς κάποια μοντέλα είναι κατάλληλα μόνο για περιπτώσεις με δύο αναμενόμενες κλάσεις ή -γενικότερα- έχουν διαφορετική απόδοση, ανάλογα με τον αριθμό των κλάσεων.

  Επομένως, για την επιλογή του κατάλληλου αλγορίθμου, θα πρέπει να ληφθούν υπ' όψη ο σκοπός χρήσης του μοντέλου, τα κριτήρια αποδοχής καθώς και τα υπάρχοντα σύνολα δεδομένων.

4. **Επεξεργασία και αξιολόγηση των δεδομένων (Prepare and Test Data):** Τα δεδομένα που θα χρησιμοποιηθούν για την εκπαίδευση, την αρχική αξιολόγηση (evaluation), καθώς και για τη βελτιστοποίηση/λεπτομερή ρύθμιση (tuning) του μοντέλου ML, θα πρέπει να είναι αντιπροσωπευτικά σε σχέση με τα δεδομένα που θα δεχθεί το μοντέλο ML κατά τη λειτουργία του σε παραγωγικό περιβάλλον.

   Μερικές φορές, για την εκπαίδευση του μοντέλου είναι δυνατό να χρησιμοποιηθούν προϋπάρχοντα σύνολα δεδομένων, κατάλληλα προ-επεξεργασμένα. Εάν δεν είναι εφικτό, τα πρωτογενή (raw) σύνολα δεδομένων που θα χρησιμοποιηθούν κατά τη φάση «Model generation and Test» (Εικόνα 9) φάση θα πρέπει να υποστούν κατάλληλη επεξεργασία. Η επεξεργασία των δεδομένων αποτελείται από τρείς βασικές διαδικασίες α) την ανάκτηση των δεδομένων (data acquisition), β) την προ-επεξεργασία των δεδομένων, και γ) τη μηχανική χαρακτηριστικών (feature engineering). Σημειώνεται, ότι πολλές φορές παράλληλα με την επεξεργασία των δεδομένων πραγματοποιείται διερευνητική ανάλυση δεδομένων (Exploratory data analysis, EDA).

   Σε αυτό το στάδιο η επεξεργασία των δεδομένων θα πρέπει να αυτοματοποιηθεί, εάν αυτό είναι εφικτό, και στη συνέχεια τα δεδομένα θα πρέπει να αξιολογηθούν (πρβλ. ενότητα 2.3).

5. **Εκπαίδευση του Μοντέλου (Train the model):** Σε αυτή τη φάση, τα δεδομένα χρησιμοποιούνται από τον αλγόριθμο για την εκπαίδευση του μοντέλου ML.

Για παράδειγμα, οι αλγόριθμοι που χρησιμοποιούνται για τη δημιουργία ενός NN απαιτείται να διαβάσουν τα δεδομένα εκπαιδεύσης πολλές φορές. Ο αριθμός των φορών που απαιτείται να διαβαστούν τα δεδομένα καλείται «εποχές» («epoch»).

Επομένως, για να εκπαιδευτεί το μοντέλο, θα πρέπει πρώτα να έχουν προσδιοριστεί τα ακόλουθα στον αλγόριθμο:

- Οι παράμετροι που καθορίζουν τη δομή του μοντέλου ML (π.χ.: ο αριθμός των επιπέδων ενός NN) και ονομάζονται «model hyperparameters»
- Οι παράμετροι που καθορίζουν την εκπαίδευση του μοντέλου (π.χ.: ο αριθμός των εποχών) και ονομάζονται «algorithm hyperparameters».

6. **Αξιολόγηση του μοντέλου (Evaluate the model):** Το μοντέλο αξιολογείται με βάση τις Μετρικές αξιολόγησης που ορίστηκαν κατά το αρχικό στάδιο. Για τον σκοπό αυτό, γίνεται χρήση των δεδομένων επαλήθευσης (validation dataset). Στη συνέχεια, τα αποτελέσματα της αξιολόγησης χρησιμοποιούνται για να πραγματοποιηθούν προσαρμογές και βελτιώσεις στο μοντέλο. Η διαδικασία αυτή είναι γνωστή ως «Tuning» και θα πρέπει να διεξάγεται προσεκτικά, ως επιστημονικό πείραμα, κάτω από ελεγχόμενες συνθήκες.

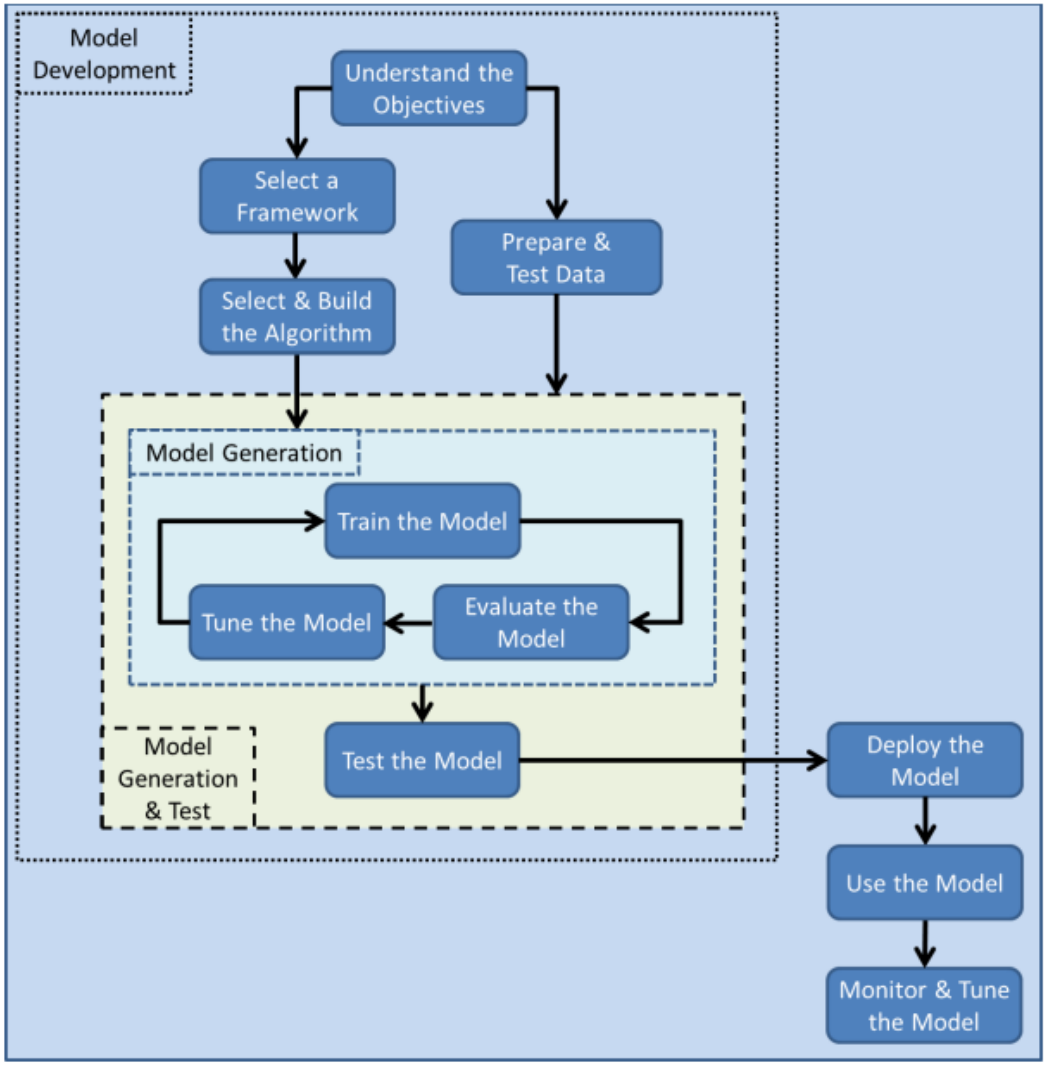
   Επίσης, θα πρέπει να σημειωθεί, ότι στην πράξη αναπτύσσονται πολλά μοντέλα ML για να επιλύσουν το πρόβλημα, καθένα εκ των οποίων βασίζεται σε διαφορετικό αλγόριθμο (π.χ. SVM, Deep Neural Network - DNN), και στη συνέχεια επιλέγεται το μοντέλο που παρουσιάζει τα καλύτερα αποτελέσματα στις Μετρικές.

7. **Βελτιστοποίηση/λεπτομερής ρύθμιση του μοντέλου (Tune the model):** Τα αποτελέσματα της φάσης «Αξιολόγηση του μοντέλου» χρησιμοποιούνται για να βελτιώσουν την απόδοση του μοντέλου ML. Αυτό επιτυγχάνεται είτε τροποποιώντας τα χαρακτηριστικά του μοντέλου (π.χ. αλλάζοντας τον αριθμό των νευρώνων ενός NN) είτε αλλάζοντας τον τρόπο εκπαίδευσης του (π.χ.: αύξηση του όγκου των δεδομένων που χρησιμοποιούνται για εκπαίδευση).
8. **Δομική αξιολόγηση του μοντέλου (Test the model):** Μόλις δημιουργηθεί το μοντέλο ML, γίνεται χρήση των δεδομένων δοκιμών (test dataset) προκειμένου να αξιολογηθεί εάν οι τιμές των Μετρικών αξιολόγησης είναι ικανοποιητικές (πάνω από το κατώτατο ορισμένο όριο αποδοχής). Επίσης, τα αποτελέσματα των Μετρικών αξιολόγησης που

προέκυψαν από τα δεδομένα δοκιμών συγκρίνονται με τα αποτελέσματα που προέκυψαν από τα δεδομένα επαλήθευσης, προκειμένου να διαπιστωθεί εάν η απόδοση του μοντέλου στα δεδομένα δοκιμών είναι χαμηλότερη από την απόδοση του στα δεδομένα επαλήθευσης. Σε αυτή την περίπτωση, μπορεί να είναι αναγκαία η επιλογή ενός διαφορετικού μοντέλου.

Σε αυτό το στάδιο, θα πρέπει να αξιολογηθούν επίσης τα μη-λειτουργικά χαρακτηριστικά του συστήματος ML, για παράδειγμα η ταχύτητα των προβλέψεων του μοντέλου, οι πόροι (resources) που απαιτούνται για την λειτουργία του και ο χρόνος εκπαίδευσης του μοντέλου.

9. **Ανάπτυξη του μοντέλου (Deploy the model):** Στη συνέχεια, το μοντέλο αναπτύσσεται (deployed) στο Clould ή σε ένα Ενσωματωμένο σύστημα (Embedded system) μέσω του framework. Οι χρήστες συνήθως μπορούν να χρησιμοποιήσουν το μοντέλο μέσω ενός web API ή μέσω κατάλληλης εφαρμογής.
10. **Χρήση του μοντέλου:** Τα μοντέλα AI, αναλόγως του τρόπου σχεδιασμού τους, μπορούν είτε να απαντούν άμεσα (πραγματοποίηση προβλέψεων σε πραγματικό χρόνο) είτε να απαντούν σε καθορισμένα χρονικά διαστήματα (πραγματοποίηση προβλέψεων με μαζικό τρόπο (batch predictions).
11. **Παρακολούθηση και βελτιστοποίηση του μοντέλου (Monitor &Tune the Model):** Είναι σημαντικό, μετά την ανάπτυξη του μοντέλου ML στο παραγωγικό περιβάλλον, να παρακολουθείται η συμπεριφορά του προκειμένου να διασφαλιστεί ότι το μοντέλο διατηρεί την αναμενόμενη συμπεριφορά του και δεν παρεκκλίνει από αυτή. Σε περίπτωση που διαπιστωθεί ότι το μοντέλο παρεκκλίνει, θα πρέπει να ρυθμιστούν οι παράμετροι του μοντέλου ή να επανεκπαιδευτεί, προκειμένου να συνεχίζει να κάνει προβλέψεις με ακρίβεια.



Εικόνα 9: Διαδικασία ανάπτυξης ενός ML συστήματος (Dussa-Zieger, et al., 2021)

2.2 Διαφορές στις δοκιμές επαλήθευσης και αξιολόγησης των συστημάτων ML έναντι των συμβατικών συστημάτων

Η δομική επαλήθευση και αξιολόγηση των συστημάτων ML παρουσιάζει αρκετές διαφορές σε σχέση με την επαλήθευση και αξιολόγηση των συμβατικών συστημάτων, οι κυριότερες εκ των οποίων είναι:

1. **Οι συνιστώσες (components) του συστήματος:** Κατά τη δοκιμή αξιολόγησης, η μη σωστή λειτουργία του ενός συμβατικού συστήματος οφείλεται συνήθως σε προβλήματα που υπάρχουν στοn κώδικα, ενώ η τα αντίστοιχα ζητήματα στη συμπεριφορά ενός συστήματος ML οφείλονται σε διαφορετικές αιτίες που συνήθως ανάγονται:

- Στα δεδομένα (π.χ. σφάλματα ή προβλήματα στα δεδομένα που χρησιμοποιούνται για την εκπαίδευση του μοντέλου),
- Πρόγραμμα Μάθησης (π.χ. σφάλματα στον ίδιο τον αλγόριθμο ή στον τρόπο που μαθαίνει το μοντέλο),
- Framework: (π.χ. σφάλματα στο framework που χρησιμοποιείται για την ανάπτυξη του μοντέλου ML).

2. **Η συμπεριφορά του συστήματος:** Ο τρόπος λειτουργίας των συμβατικών συστημάτων ορίζεται συνήθως λεπτομερώς μέσω των απαιτήσεων (requirements), ωστόσο η συμπεριφορά των συστημάτων ML δεν μπορεί πλήρως να καθοριστεί, καθώς η συμπεριφορά τους αλλάζει κάθε φορά που το μοντέλο επανεκπαιδεύεται σε νέα δεδομένα.
3. **Τα δεδομένα εισόδου:** Στα συμβατικά συστήματα, τα δεδομένα που χρησιμοποιούνται για τη δοκιμή επαλήθευσης και αξιολόγησης του συστήματος, περιορίζονται συνήθως στα δεδομένα που μπορεί να δεχθεί ο κώδικας ως είσοδο. Αντίθετα, για την αξιολόγηση των συστημάτων ML χρησιμοποιούνται πολλά και διαφορετικά δεδομένα προκειμένου να προσδιοριστεί η συμπεριφορά τόσο του ML μοντέλου όσο και του ML συστήματος.
4. **Test oracle:** Στα συμβατικά συστήματα οι τιμές εξόδου του συστήματος είναι καθορισμένες (στις απαιτήσεις), και επομένως ο Αξιολογητής (π.χ.: Προγραμματιστής, Ελεγκτής/Tester) μπορεί να τις επαληθεύσει τις εξόδου του προγράμματος με τις αναμενόμενες τιμές, για να διασφαλιστεί η σωστή λειτουργεία του συστήματος.

   Αντίθετα, οι αναμενόμενες έξοδοι των συστημάτων ML είναι δύσκολο να οριστούν εξ αρχής – αυτό είναι γνωστό ως το πρόβλημα του *test oracle*. Ο προσδιορισμός του test-oracle είναι μια επίπονη και χρονοβόρα διαδικασία, επειδή απαιτείται εξειδικευμένη γνώση του τομέα εφαρμογής του μοντέλου και επειδή είναι δύσκολο να οριστούν «οι σωστές παράμετροι». Για να αντιμετωπιστεί αυτό το πρόβλημα, έχουν αναπτυχθεί διάφοροι τύποι test oracle, μερικοί εκ των οποίων είναι η Metamorphic Relations as Test Oracles και το Cross-Referencing as Test Oracles.

5. **Κριτήρια επάρκειας δοκιμών (Test adequacy criteria):** Τα κριτήρια επάρκειας αξιολογούν ποσοτικά την ποιότητα των δοκιμών αξιολόγησης. Για τα συμβατικά συστήματα, έχουν αναπτυχθεί πολλές μετρικές, μερικές εκ των οποίων είναι το dataflow coverage, το branch coverage, και το line coverage. Παρόλα αυτά, δεδομένου ότι ο τρόπος

ανάπτυξης ενός ML συστήματος διαφέρει σημαντικά ως προς τα συμβατικά, έχουν αναπτυχθεί νέες μετρικές και τρόποι αξιολόγησης (πρβλ. Κεφάλαια 3, 4), καθώς οι υφιστάμενες μετρικές και μέθοδοι δεν επαρκούν.

6. **Ψευδώς αναγνωρισμένα σφάλματα:** Δεδομένου ότι στα συστήματα ML είναι δύσκολο να προσδιοριστεί η αναμενόμενη τιμή εξόδου, εξ αίτιας του προβλήματος test-oracle, εμφανίζεται το φαινόμενο να αναφέρονται συνήθως περισσότερα ψευδώς θετικά προβλήματα (false positive bugs).

2.3 Στάδια δοκιμών επαλήθευσης και αξιολόγησης των συστημάτων AI

Τα συστήματα AI αποτελούνται από συνιστώσες (components) που βασίζονται στο AI, όσο και από συμβατικές (not-AI) συνιστώσες (Dussa-Zieger, et al., 2021). Τα στάδια των δοκιμών αξιολόγησης ενός συστήματος ML είναι τα ακόλουθα:

1. **Δοκιμές αξιολόγησης των δεδομένων εισόδου** Οι δοκιμές αξιολόγησης στα δεδομένα εισόδου έχουν ως στόχο τη διασφάλιση της ποιότητας των δεδομένων εκπαίδευσης. Αυτό μπορεί να επιτευχθεί εφαρμόζοντάς τις ακόλουθες τεχνικές:
   - Επιθεωρήσεις (Reviews).
   - Διερευνητική ανάλυση δεδομένων (exploratory data analysis, EDA) στα δεδομένα εκπαίδευσης.
   - Στατικές τεχνικές (π.χ.: έλεγχος δεδομένων για συστηματικά σφάλματα (bias))
   - Στατικές και δυναμικές δοκιμές αξιολόγησης της σωλήνωσης επεξεργασίας δεδομένων (data processing pipeline).
2. **Δοκιμές αξιολόγησης του μοντέλου ML**: Σε αυτό το στάδιο επιχειρείται να διασφαλιστεί ότι η απόδοση (performance) του μοντέλου ML ικανοποιεί τα κριτήρια αποδοχής (Acceptance criteria). Για τον σκοπό αυτό, θα πρέπει να αξιολογηθούν:
   - Τα αποτελέσματα των μετρικών αξιολόγησης (**πρβλ.** Κεφάλαιο 3**)**
   - Οι μη-λειτουργικές απαιτήσεις του συστήματος (π.χ.: χρόνος εκπαίδευσης του μοντέλου, ταχύτητα των προβλέψεων, απαιτούμενοι υπολογιστικοί πόροι)
   - Το framework, ο αλγόριθμος που έχει χρησιμοποιηθεί και οι ρυθμίσεις του μοντέλου και των υπερπαραμέτρων του, έτσι ώστε να ελεγχθεί ότι έχουν βελτιστοποιηθεί.
3. **Δοκιμές αξιολόγησης των συνιστωσών:** Αποτελεί συμβατική μέθοδος αξιολόγησης, και μπορεί να εφαρμοστεί για την αξιολόγηση όλων των συνιστωσών (π.χ. οι συνιστώσες που

χρησιμοποιούνται για την επικοινωνία των συστημάτων, οι συνιστώσες UI και άλλα), εκτός των AI.

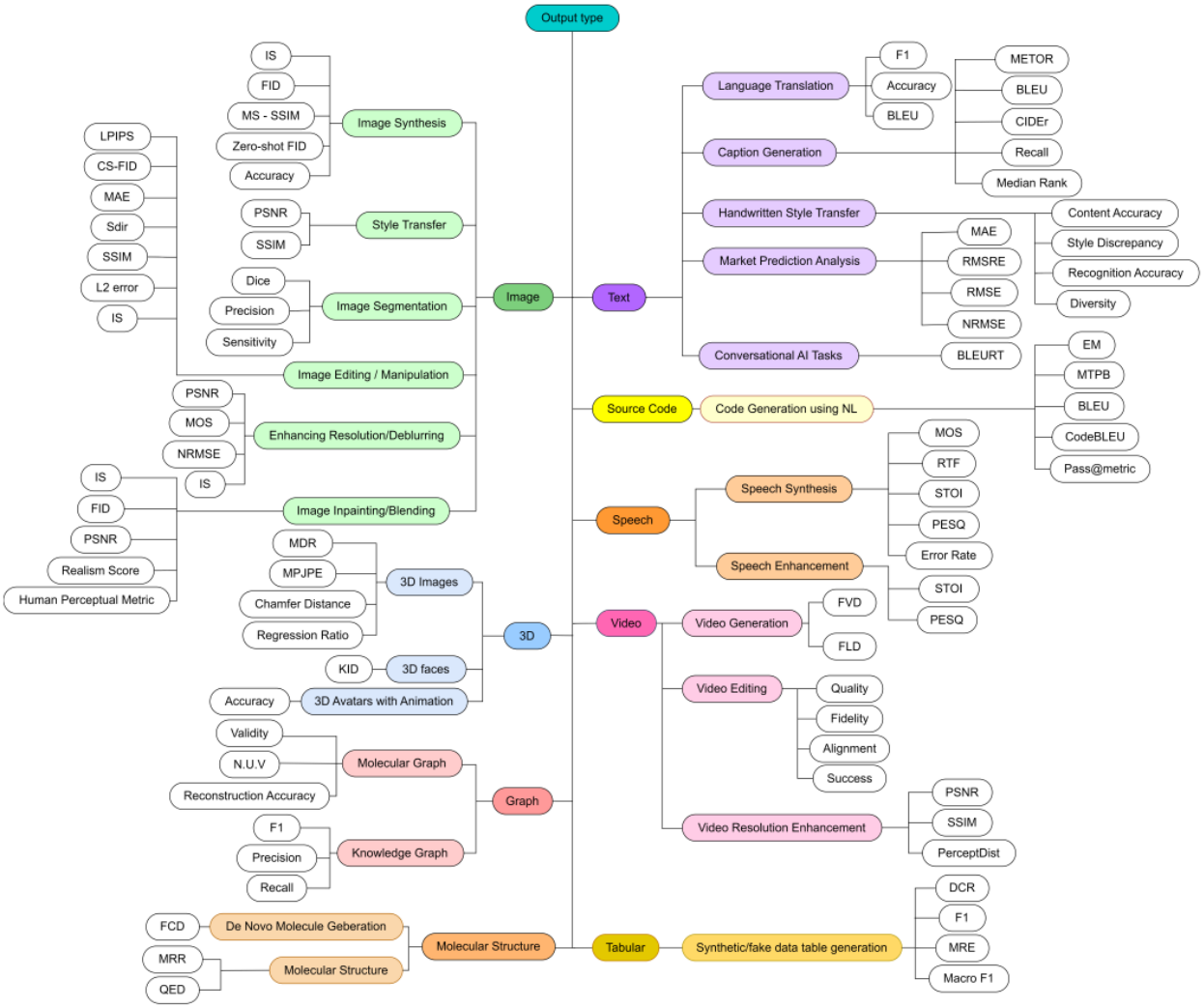
4. **Δοκιμές αξιολόγησης της Ολοκλήρωσης των Συνιστωσών (Component Integration testing):** Μόλις ολοκληρωθούν οι δοκιμές αξιολόγησης των συνιστωσών, αξιολογείται εάν τα δεδομένα διαδίδονται σωστά από τη μία συνιστώσα στην άλλη. Αναλυτικότερα, στα συστήματα AI, μπορεί να εκλεχθεί εάν η σωλήνωση δεδομένων (data pipeline) διαδίδει ορθά τα δεδομένα εισόδου στο μοντέλο (π.χ.: κατάλληλα επεξεργασμένα, μη ελλιπή, σωστού τύπου) και στη συνέχεια, να αξιολογηθεί εάν οι παραγόμενες από το μοντέλο προβλέψεις απεικονίζονται σωστά στο UI (Breck , Cai, & Nielsen, 2017). Στις περιπτώσεις που το μοντέλο δίδει τις προβλέψει χρησιμοποιώντας υπηρεσίες διαδικτύου (web services), στα πλαίσια της αξιολόγησης της ολοκλήρωσης των συστημάτων, μπορεί να πραγματοποιηθούν δοκιμές API.
5. **Δοκιμές επαλήθευσης και αξιολόγησης του Συστήματος:** Σε αυτή τη φάση, διεξάγονται οι δοκιμές επαλήθευσης και αξιολόγησης στο δοκιμαστικό περιβάλλον προκειμένου να διασφαλιστεί ότι όλα τα στοιχεία του συστήματος λειτουργούν. Στην περίπτωση που οι περιπτώσεις δοκιμών είναι δύσκολο ή επικίνδυνο να εκτελεστούν στο πραγματικό περιβάλλον, η αξιολόγηση τους γίνεται σε περιβάλλον προσομοίωσης.

   Επίσης, κατά τη δοκιμή αξιολόγησης του Συστήματος, αξιολογούνται και οι μη λειτουργικές απαιτήσεις του συστήματος όπως είναι η ευρωστία (robustness) μέσω του adversarial testing, και η δυνατότητα επεξήγησης (explainability). Ακόμα, σε αυτό το στάδιο, θα πρέπει να επανεξεταστούν οι τιμές των μετρικών αξιολόγησης, ώστε να διασφαλιστεί ότι δεν έχουν επηρεαστεί από την ανάπτυξη (deployment) του μοντέλου. Η πρακτική αυτή εφαρμόζεται κυρίως όταν οι συνιστώσες AI τροποποιούνται. Χαρακτηριστικό παράδειγμα τροποποίησης αποτελεί η συμπίεση του DNN προκειμένου να μειωθεί το μέγεθος του (Dussa-Zieger, et al., 2021).

6. **Έλεγχος αποδοχής (Acceptance Testing):** Αποτελεί συμβατική μέθοδος αξιολόγησης και διεξάγεται από τον πελάτη προκειμένου να διασφαλίσει το σύστημα λειτουργεί με βάση τις προκαθορισμένες απαιτήσεις.

3 Μετρικές Αξιολόγησης

Οι μετρικές αξιολόγησης αποτελούν κρίσιμο στοιχείο για την αποτίμηση της απόδοσης των συστημάτων AI και συμβάλλουν στην ποσοτική εκτίμηση της αποτελεσματικότητας, της ακρίβειας και της αξιοπιστίας των μοντέλων. Στη συνέχεια, οι μετρικές αναλύονται με βάση την αναμενόμενη έξοδο (π.χ.: εικόνα, κείμενο, βίντεο) όπως φαίνεται στην Εικόνα 10.



Εικόνα 10: Κατηγοριοποίηση των μετρικών αξιολόγησης με βάση την αναμενόμενη έξοδο του μοντέλου

3.1 Μετρικές απόδοσης της ταξινόμησης (Classification)

3.1.1 Πίνακας σύγχυσης

Για τον προσδιορισμό της απόδοσης μιας διαδικασίας ταξινόμησης χρησιμοποιείται ευρέως ο πίνακας σύγχυσης (Confusion matrix). Ο πίνακας δημιουργείται από τον προσδιορισμό των ακολούθων τιμών true positives (TP), true negatives (TN), false positives (FP), false negative (FN) (Naidu, Zuva, & Sibanda, 2023). Πιο συγκεκριμένα, ο πίνακας σύγχυσης έχει τη δομή που παρουσιάζει ο Πίνακας 2:

Πίνακας 2. Δομή του πίνακα σύγχυσης

		Actual	
		Positive	**Negative**
Predicted	**Positive**	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Ειδικότερα:

- TP: Εκφράζει τον συνολικό αριθμό των αποφάσεων που έχουν ταξινομηθεί σωστά στην θετική κλάση από τον αλγόριθμο,
- TN: Εκφράζει τον συνολικό αριθμό των αποφάσεων που έχουν ταξινομηθεί σωστά στην αρνητική κλάση από τον αλγόριθμο,
- FP: Εκφράζει τον συνολικό αριθμό των αποφάσεων που έχουν ταξινομηθεί λανθασμένα στην θετική κλάση από τον αλγόριθμο,
- FN: Εκφράζει τον συνολικό αριθμό των αποφάσεων που έχουν ταξινομηθεί λανθασμένα στην αρνητική κλάση από τον αλγόριθμο.

3.1.2 Ορθότητα

Η ορθότητα (accuracy) είναι από τις πιο συχνά χρησιμοποιημένες μετρικές για τα προβλήματα ταξινόμησης. Εκφράζει το ποσοστό όλων των σωστών ταξινομήσεων και μαθηματικά μπορεί να οριστεί ως:

$$Accuracy = \frac{Number\ of\ Correctly\ Classified\ Samples}{Total\ Number\ of\ Samples} * 100\%$$

Έχοντας ως βάση τον πίνακα σύγχυσης, η ορθότητα μπορεί να εκφραστεί ως:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} * 100\%$$

Η ορθότητα δεν θα πρέπει να χρησιμοποιείται όταν ο αριθμός των δειγμάτων δεν είναι ισομερώς κατανεμημένος σε όλες τις κλάσεις, διότι έχει την τάση να ευνοεί την κλάση που περιέχει τα περισσότερα δείγματα (Naidu, Zuva, & Sibanda, 2023). Για παράδειγμα, έστω ότι ο σκοπός ενός μοντέλου είναι να εντοπίσει την ύπαρξη ή μη του καρκίνου στα κύτταρα. Αν μόνο ένα από τα 100 δείγματα πράγματι περιέχει καρκίνο, το μοντέλο θα έχει ορθότητα 99% αλλά δεν θα είναι χρήσιμο καθώς δεν βοηθά στον εντοπισμό των πραγματικών περιπτώσεων καρκίνου. Σε αυτές τις περιπτώσεις, ο συνδυασμός των μετρικών ακρίβειας, ανάκλησης (recall), και F1-score θεωρείται καταλληλότερος, καθώς λαμβάνει υπόψη τόσο τις αληθινά θετικές προβλέψεις όσο και τις λανθασμένες (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023).

3.1.3 Ακρίβεια

Ως ακρίβεια (precision) ορίζεται ως ο λόγος του συνόλου των αποφάσεων που έχουν ταξινομηθεί σωστά στην θετική κλάση προς το σύνολο των αποφάσεων τα οποία ταξινομήθηκαν στη θετική κλάση (Naidu, Zuva, & Sibanda, 2023).

$$\text{Precision } = \frac{TP}{TP + FP} * 100\%$$

Υψηλή ακρίβεια υποδεικνύει ότι το πλήθος των ψευδώς θετικών προβλέψεων είναι μικρό, και κατά συνέπεια οι αποφάσεις είναι αξιόπιστες (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023). Ωστόσο, για την αξιολόγηση της απόδοσης ενός μοντέλου δεν πρέπει να λαμβάνεται υπόψη μόνο η ακρίβεια, καθώς υψηλή ακρίβεια μπορεί να επιτευχθεί και από μοντέλα

που παράγουν ελάχιστες θετικές προβλέψεις (π.χ. μόνον όταν το μοντέλο είναι εξαιρετικά πιθανό η πραγματική τιμή να είναι θετική), χαρακτηρίζοντας αρνητικές τις υπόλοιπες. Μοντέλα ωστόσο με αυτή τη συμπεριφορά θα έχουν πολύ χαμηλή ανάκληση (recall).

3.1.4 Ανάκληση ή ρυθμός αληθώς θετικών ή ευαισθησία

Η ανάκληση (recall) καλείται επίσης ρυθμός αληθώς θετικών (True Positive Rate) ή ευαισθησία (Sensitivity). Εκφράζει τον λόγο των αποφάσεων που έχουν ταξινομηθεί σωστά στην θετική κλάση προς τον σύνολο τον συνολικό αριθμό των ορθών αποφάσεων (Naidu, Zuva, & Sibanda, 2023). Ουσιαστικά μετρά την ικανότητα του μοντέλου να αναγνωρίσει όλες τις θετικές περιπτώσεις στο σύνολο δεδομένων. Υπολογίζεται μαθηματικά από τον ακόλουθο τύπο:

$$Recall = \frac{TP}{TP + FN} * 100\%$$

Υψηλό recall υποδεικνύει ότι το μοντέλο κάνει λίγες ψευδώς αρνητικές (FN) προβλέψεις και επομένως εντοπίζει τις περισσότερες θετικές περιπτώσεις.

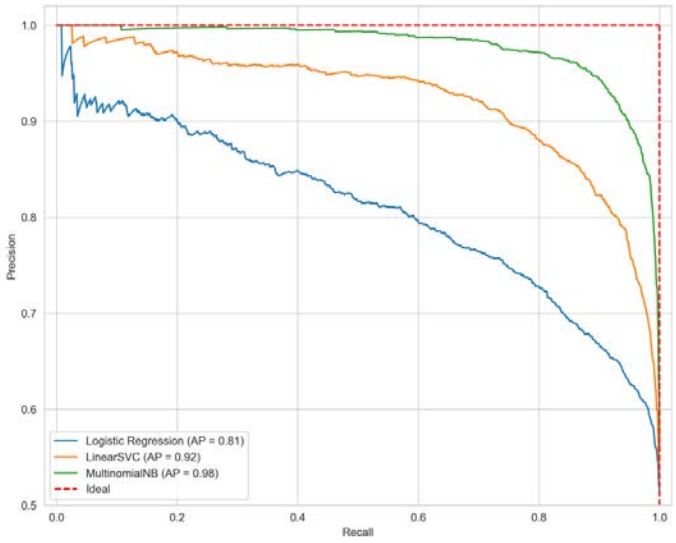
3.1.5 Αντιστάθμισμα μεταξύ ακρίβειας και ανάκλησης

Η δημιουργία μοντέλων ML που ταυτόχρονα να διαθέτουν υψηλή ακρίβεια και ανάκληση δεν είναι πάντοτε εφικτή. Εάν η μετρική της ακρίβειας αυξηθεί, η ανάκληση θα μειωθεί και το αντίστροφο. Αυτή η σχέση μεταξύ των μετρικών, καλείται *αντιστάθμισμα μεταξύ ακρίβειας και ανάκλησης* (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023). Επομένως, με βάση τις απαιτήσεις (requirements) και τους περιορισμούς της εκάστοτε εφαρμογής, θα πρέπει να προσδιορίζεται η βέλτιστη σχέση μεταξύ των δύο μετρικών.

Παραδείγματος χάρη, έστω ότι ο σκοπός ενός μοντέλου ML είναι να εντοπίζει τα ανεπιθύμητα (spam) emails. Εάν το μοντέλο έχει ρυθμιστεί ώστε να επιτυγχάνει υψηλή ακρίβεια, τότε θα μαρκάρει τα emails ως ανεπιθύμητα μόνο όταν είναι απολύτως βέβαιο, ωστόσο δεν θα χαρακτηρίζει ως spam πολλά ανεπιθύμητα emails (λόγω της χαμηλής βεβαιότητας), οδηγώντας το σε χαμηλή τιμή για τη μετρική της ανάκλησης. Αντίθετα, αν το μοντέλο είναι ρυθμισμένο να επιτυγχάνει υψηλή τιμή ανάκλησης, πιθανότατα θα επισημαίνει σωστά τα περισσότερα ανεπιθύμητα emails. Ωστόσο, αυτή η προσέγγιση θα οδηγεί το μοντέλο να αναγνωρίζει εσφαλμένα πολλά μη ανεπιθύμητα emails ως ανεπιθύμητα, οδηγώντας σε χαμηλή ακρίβεια.

Καθίσταται κατανοητό ότι η εύρεση της βέλτιστης σχέσης μεταξύ της ακρίβειας και της ανάκλησης είναι σημαντική για την σωστή ρύθμιση των μοντέλων ML.

Στη *Εικόνα 11* αναπαρίσταται γραφικά το Precision ως προς την ανάκληση για τρία μοντέλα που εκπαιδεύτηκαν στο ίδιο σύνολο δεδομένων, ενώ η διακεκομμένη γραμμή αναπαριστά την ιδανική απόδοση ενός μοντέλου. Όσο πιο κοντά είναι η καμπύλη του μοντέλου στην πάνω-δεξιά γωνία, τόσο πιο ισορροπημένη είναι η απόδοση του σε ανάκληση και ακρίβεια για τις διάφορες τιμές κατωφλίου. Επίσης, στην *Εικόνα 11* παρατηρείται ότι η καμπύλη του μοντέλου MultinomialNB (πράσινη γραμμή), βρίσκεται εξ ολοκλήρου πάνω από την καμπύλη του μοντέλου LinearSVC (πορτοκαλί γραμμή), και του μοντέλου Logistic Regression (μπλε γραμμή). Αυτό σημαίνει ότι η απόδοση του είναι καλύτερη σε σχέση με τα υπόλοιπα μοντέλα.



Εικόνα 11: Διάγραμμα Precision-Recall τριών μοντέλων που έχουν εκπαιδευτεί στο ίδιο σύνολο δεδομένων.

Επίσης, το εμβαδόν κάτω από την κάτω από την καμπύλη (Area Under Curve-AUC) του γραφήματος ακρίβειας-ανάκλησης αποτελεί μέτρο αξιολόγησης του ταξινομητή. Ένα μοντέλο θεωρείται καλό όταν το AUC είναι μεγαλύτερο από 0,5 ενώ όταν το AUC-PR είναι 1, σημαίνει ότι ο ταξινομητής είναι τέλειος.

3.1.6 F1-Score

Το F1-Score ορίζεται ως ο αρμονικός μέσος της ακρίβειας και της ανάκλησης. Μαθηματικά ορίζεται από τον τύπο ως:

$$\text{F1-Score} = 2 * \frac{\text{Precision} * Recall}{Precision + Recall} * 100\%$$

Όπως φαίνεται και από τον μαθηματικό τύπο, το F1-Score λαμβάνει υπόψη τόσο την ικανότητα του μοντέλου να εντοπίζει όλες τις πραγματικές θετικές περιπτώσεις (Recall) όσο και την ικανότητα του μοντέλου να προβλέπει σωστά τις θετικές περιπτώσεις (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023). Χρησιμοποιείται κυρίως όταν η κατανομή των δεδομένων στις κλάσεις δεν είναι ισομερώς κατανεμημένη (imbalance) ή όταν η ανάκληση και η ακρίβεια είναι εξίσου σημαντικές. Χαμηλό F1-Score υποδηλώνει ότι το μοντέλο έχει είτε χαμηλή ανάκληση είτε χαμηλή ακρίβεια (είτε χαμηλή επίδοση και στις δύο μετρικές). Αντίθετα, υψηλή τιμή για το F1-Score υποδηλώνει ότι το μοντέλο ML έχει καλή ισορροπία μεταξύ της ανάκλησης και της ακρίβειας.

3.1.7 F2-score

Το F2-score αποτελεί παραλλαγή του F1-score. Αυτή η μετρική δίνει μεγαλύτερο βάρος στην ανάκληση, γεγονός που καθιστά αυτή τη μετρική χρήσιμη όταν είναι σημαντικό να ελαχιστοποιηθούν τα ψευδώς αρνητικά αποτελέσματα (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023).

$$\text{F2-Score} = (1 + 2^2) * \frac{\text{Precision} * Recall}{2^2 * Recall + Recall} * 100\%$$

Η τιμή του F2-score κυμαίνεται από 0 έως 1. Υψηλή τιμή υποδηλώνει ότι το μοντέλο έχει καλύτερη απόδοση.

3.1.8 Εξειδίκευση ή ρυθμός αληθώς αρνητικών

Η εξειδίκευση (specificity) καλείται επίσης ρυθμός αληθώς αρνητικών (True Negative rate) και εκφράζει την ικανότητα του μοντέλου να αναγνωρίζει τις πραγματικές αρνητικές περιπτώσεις στο σύνολο δεδομένων. Υπολογίζεται μαθηματικά από τον ακόλουθο τύπο:

$$Specificity = \frac{TN}{TN + FP} * 100\%$$

Αυτή η μετρική χρησιμοποιείται ευρέως στις ιατρικές διαγνωστικές εξετάσεις, διότι ελαχιστοποιώντας τον αριθμός των ψευδών θετικών προβλέψεων, αποφεύγονται οι περιττές

θεραπείες ή η περεταίρω διενέργεια εξετάσεων (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023).

Υψηλή εξειδίκευση υποδεικνύει ότι το μοντέλο κάνει λίγες ψευδώς θετικές (FP) προβλέψεις και επομένως εντοπίζει τις περισσότερες αρνητικές περιπτώσεις. Για την καλύτερη αξιολόγηση της συνολικής απόδοσης ενός μοντέλου ταξινόμησης ML, η μετρική αυτή χρησιμοποιείται μαζί με την ανάκληση, ή την αποτίμηση των σφαλμάτων τύπου I/ρυθμού ψευδώς θετικών (βλ. παράγραφο 3.1.9).

3.1.9 Ρυθμός ψευδών θετικών

Ο ρυθμός ψευδών θετικών (False Positive Rate, FPR) καλείται επίσης ρυθμός σφαλμάτων τύπου I (Type I Error rate) και υπολογίζεται ως ο λόγος των αποφάσεων που έχουν ταξινομηθεί εσφαλμένα στην θετική κλάση προς τον συνολικό αριθμό των αρνητικών περιπτώσεων, και δίδεται από τον ακόλουθο τύπο:

$$FPR = \frac{FP}{FP + TN} * 100\%$$

Το FPR συνδέεται άμεσα με το κατώφλι που καθορίζει τις περιπτώσεις ως θετικές ή αρνητικές. Όταν το όριο του κατωφλίου ορίζεται χαμηλά, ο αριθμός των ψευδών θετικών αυξάνεται και άρα το FPR αυξάνεται (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023). Αντίθετα, όσο υψηλότερα τίθεται το όριο του κατωφλίου, τόσο μειώνεται ο αριθμός των ψευδών θετικών και άρα το FPR.

Στην πράξη, το FPR συνήθως απεικονίζεται στον άξονα x ενός γραφήματος ROC (receiver operating characteristic), προκειμένου να δείξει την ισορροπία μεταξύ του ποσοστού ανίχνευσης θετικών περιπτώσεων (TPR) και του ποσοστού λανθασμένων θετικών προβλέψεων (FPR) για διαφορετικά κατώφλια ταξινόμησης.

3.1.10 Αρνητική προγνωστική αξία

Η αρνητική προγνωστική αξία (Negative Predictive Value, NPV) εκφράζει το ποσοστό των αρνητικών περιπτώσεων που αναγνωρίζονται σωστά ως αρνητικές και χρησιμοποιείται όταν το κόστος των ψευδώς αρνητικών προβλέψεων είναι υψηλό (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023). Υπολογίζεται από τον ακόλουθο τύπο:

$$NPV = \frac{TN}{TN + FN} * 100\%$$

Θα πρέπει να σημειωθεί ότι η NPV αποτελεί καλή μετρική για την αξιολόγηση της απόδοση του ML μοντέλου ακόμα και όταν η κατανομή των δεδομένων στις κλάσεις είναι ανισομερής.

Το NPV μπορεί να υπολογιστεί και από τον ακόλουθο τύπο:

$$NPV = 1 - FPR$$

3.1.11 True Discovery Rate (TDR)

Το True Discovery Rate (TDR) καλείται επίσης και θετική προγνωστική αξία (Positive Predictive Value, PPV) ή ακρίβεια της θετικής κλάσης (precision of the positive class). Το TDR εκφράζει το ποσοστό των αληθών θετικών προβλέψεων ως προς το συνολικό αριθμό των θετικών προβλέψεων (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023).

$$TDR = \frac{TP}{TP + FP} * 100\%$$

Η TDR χρησιμοποιείται ιδίως όταν ο αριθμός των αληθών θετικών προβλέψεων είναι χαμηλός, και ο αριθμός των ψευδώς θετικών προβλέψεων υψηλός. Επίσης, η μετρική αυτή είναι χρήσιμη όταν το σύνολα των δεδομένων είναι υψηλών διαστάσεων, δηλαδή όταν ο αριθμός των χαρακτηριστικών είναι μεγάλος, και ο αριθμός των θετικών παρατηρήσεων είναι χαμηλός.

Το TDR είναι υψηλό όταν το Recall είναι χαμηλό, και το αντίστροφο. Επομένως, για την αξιολόγηση της απόδοση ενός μοντέλου ML είναι σημαντικό να λαμβάνεται υπόψη τόσο η μετρική TDR, όσο και η μετρική Recall.

3.1.12 False Discovery Rate (FDR)

Το FDR εκφράζει το ποσοστό των ψευδών θετικών προβλέψεων ως προς το συνολικό αριθμό των θετικών προβλέψεων. Υπολογίζεται από τον ακόλουθο τύπο:

$$FDR = \frac{FP}{TP + FP} * 100\%$$

Χρησιμοποιείται όταν ο αριθμός των ψευδώς θετικών προβλέψεων (FP) είναι πιο σημαντικός από τον αριθμό των ψευδών αρνητικών (FN). Χαμηλή τιμή FDR υποδηλώνει ότι το μοντέλο κάνει

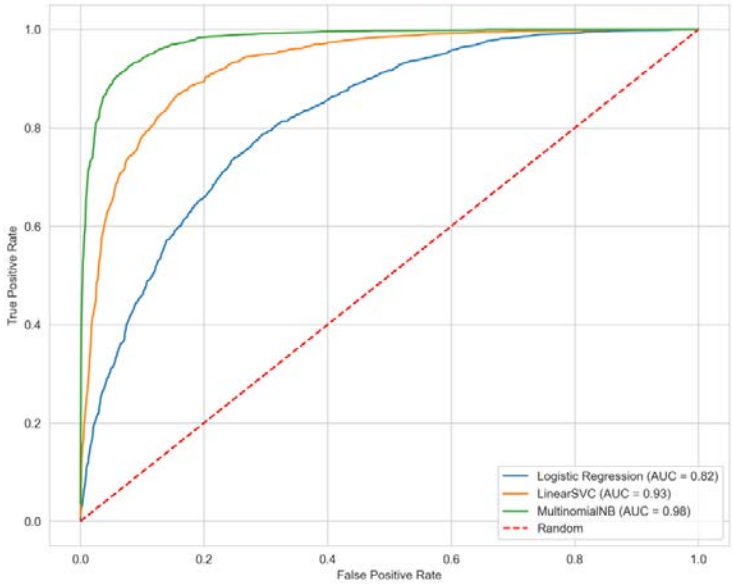
λίγες ψευδώς θετικές προβλέψεις (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023). Παράδειγμα χρήσης της μετρικής FDR αποτελεί η εφαρμογή της για την αξιολόγηση μοντέλων Μηχανικής Μάθησης σε ιατρικές εξετάσεις ή την ανίχνευση απάτης.

3.1.13 Area Under the Receiver Operation Characteristics curve (AUC-ROC)

Όπως έχει ήδη αναφερθεί, για την αξιολόγηση της απόδοσης ενός ταξινομητή δεν αρκεί ο υπολογισμός μόνο της ακρίβειας, καθώς αυτή η μετρική δεν λαμβάνει υπόψιν την κατανομή των κλάσεων (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023). Το πρόβλημα αυτό μπορεί να επιλυθεί από τις καμπύλες ROC, οι οποίες πρωτοεμφανίστηκαν το 1940.

Στην *Εικόνα 12* αναπαρίσταται η καμπύλη ROC που οποία δημιουργείται από την αντιστοίχιση μεταξύ TPR στο FPR, για τις διάφορες τιμές κατωφλίου και για διαφορετικούς αλγόριθμους. Στον κάθετο άξονα τοποθετείται το TPR και στον οριζόντιο το FPR. Η διακεκομμένη κόκκινη γραμμή ορίζει την τυχαιότητα (η απόφαση λαμβάνεται τυχαία). Αν η καμπύλη ενός μοντέλου ML βρίσκεται κάτω από τη διαγώνιο, το μοντέλο έχει χαμηλή ακρίβεια και επομένως ο ταξινομητής κάνει τυχαίες προβλέψεις.

Η περιοχής κάτω από την καμπύλη ROC (AUC-ROC) χρησιμοποιείται ευρέως για την μέτρηση της απόδοσης ενός ταξινομητή. Όταν το AUC-ROC είναι μεγαλύτερο από 0,5 θεωρείται ότι το μοντέλο δεν κάνει τυχαίες προβλέψεις, ενώ όταν το AUC- ROC τείνει στο 1 ο ταξινομητής είναι τέλειος.



Εικόνα 12: Διάγραμμα ROC τριών διαφορετικών μοντέλων που έχουν εκπαιδευτεί στο ίδιο σύνολο δεδομένων

3.2 Μετρικές απόδοσης της παλινδρόμηση (Regression)

3.2.1 Μέσο απόλυτο σφάλμα

Το μέσο απόλυτο σφάλμα (Mean Absolute Error, MAE) καλείται επίσης L1-loss, και μετρά την μέση τιμή, των απόλυτων διαφορών των πραγματικών τιμών (y_i) από τις προβλεπόμενες ($\hat{y}_i$). Υπολογίζεται από τον ακόλουθο τύπο:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Όσο μεγαλύτερη είναι η τιμή του MAE, τόσο μεγαλύτερες είναι οι αποκλίσεις μεταξύ των προβλεπόμενων και των πραγματικών τιμών, ενώ όταν το MAE τείνει στο 0, το μοντέλο κάνει καλές προβλέψεις. Σημειώνεται ότι η θεώρηση της τιμής του MAE ως υψηλής ή χαμηλής εξαρτάται από το πεδίο τιμών των μετρήσεων: αν π.χ. οι μετρήσεις λαμβάνουν τιμές στην περιοχή [10^5, 10^6] τιμές MAE της τάξης του 5 θα πρέπει να θεωρούνται πολύ ικανοποιητικές, ενώ αν μετρήσεις λαμβάνουν τιμές στην περιοχή [0, 10], τιμές MAE της τάξης του 5 θεωρούνται ιδιαίτερα υψηλές.

Όπως φαίνεται από τον τύπο, η MAE είναι γραμμική συνάρτηση των σφαλμάτων πρόβλεψης (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023). Η ιδιότητα αυτή, την καθιστά λιγότερο ευαίσθητη στις ακραίες τιμές σφαλμάτων σε σχέση με το μέσο τετραγωνικό σφάλμα (πρβλ. ενότητα 3.2.2), καθιστώντας την κατάλληλή για εφαρμογές όπου αναμένεται η παρουσία ακραίων τιμών σφαλμάτων, που ενίοτε αποδίδεται σε έκτοπα δεδομένα (outlier data) ή η όπου η κατανομή των σφαλμάτων δεν είναι συμμετρική.

3.2.2 Μέσο τετραγωνικό σφάλμα

Το Μέσο τετραγωνικό σφάλμα (Mean Square Error, MSE) καλείται επίσης L2-loss και υπολογίζεται ως ο μέσος όρος των τετραγώνων των αποκλίσεων μεταξύ των πραγματικών (y_i) και των προβλεπόμενων τιμών ($\hat{y}_i$):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Όταν το MSE είναι 0, οι προβλεπόμενες και οι πραγματικές τιμές ταυτίζονται, και επομένως η ακρίβεια του μοντέλου είναι υψηλή (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023). Αντίθετα, υψηλές τιμές του MSE, υποδηλώνουν μεγαλύτερες αποκλίσεις μεταξύ των προβλεπόμενων και των πραγματικών τιμών, δηλαδή μεγαλύτερα σφάλματα στις προβλέψεις του μοντέλου.

Όπως φαίνεται από τον τύπο, η MSE είναι μια τετραγωνική συνάρτηση των σφαλμάτων πρόβλεψης και λόγω αυτής της ιδιότητας της, δεν είναι ανθεκτική σε αποκλίσεις με ακραίες τιμές (οι οποίες ενίοτε προκαλούνται από έκτοπα δεδομένα - outlier data), και άρα, μπορεί να οδηγήσει στη δημιουργία μοντέλων που δίνουν προτεραιότητα στη μείωση των μεγάλων σφαλμάτων παρά στων μικρότερων. Επομένως, αν τα δεδομένα περιλαμβάνουν ακραίες τιμές, είναι καλύτερο να χρησιμοποιηθεί η συνάρτηση Mean Absolute Error (MAE) για την αξιολόγηση του μοντέλου.

3.2.3 Root Mean Squared Error (RMSE)

Το Root Mean Squared Error (RMSE) ορίζεται μαθηματικά ως η τετραγωνική ρίζα του MSE:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Η μετρική RMSE μετρά τη μέση απόκλιση των προβλεπόμενων (y_i) από τις πραγματικές τιμές (y_i) και είναι ευαίσθητη στις ακραίες τιμές, όπως και η MSE (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023). Όταν η μετρική MSE τείνει στο 0, το μοντέλο έχει καλή απόδοση.

3.2.4 Μέσο απόλυτο ποσοστιαίο σφάλμα

Η μετρική του μέσου απόλυτου ποσοστιαίου σφάλματος (Mean Absolute Percentage Error, MAPE) υπολογίζει το μέσο ποσοστιαίο σφάλμα των προβλέψεων ($\hat{y}_i$) του μοντέλου ως προς τις πραγματικές τιμές (y_i).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} * 100\%$$

Θα πρέπει να σημειωθεί ότι όταν η πραγματική τιμή (y_i) είναι μηδέν, η MAPE παράγει μη ορισμένα αποτελέσματα, καθώς επίσης, είναι ευαίσθητη στις ακραίες τιμές (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023). Ένα χαρακτηριστικό πλεονέκτημα του εν λόγω μέτρου είναι η ανεξαρτησία του από την κλίμακα των δεδομένων που προβλέπονται. Αυτό σημαίνει ότι μπορεί να χρησιμοποιηθεί για τη σύγκριση μοντέλων σε περιπτώσεις όπου η εξαρτημένη μεταβλητή παρουσιάζει διαφορετικές κλίμακες μέτρησης.

3.2.5 Συμμετρικό μέσο απόλυτο ποσοστιαίο σφάλμα

Το συμμετρικό μέσο απόλυτο ποσοστιαίο σφάλμα (Symmetric Mean Absolute Percentage Error, SMAPE) χρησιμοποιείται ευρέως για την αξιολόγηση της ακρίβειας των προβλέψεων σε χρονοσειρές (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023). Υπολογίζεται από τον ακόλουθο τύπο:

$$SMAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} * 100\%$$

Τα μειονεκτήματα του SMAPE είναι ότι παράγει μη ορισμένα αποτελέσματα όταν τόσο η πραγματική (y_i) όσο και η προβλεπόμενη τιμή (y_i) είναι μηδέν, καθώς επίσης παρουσιάζει ευαισθησία στις ακραίες τιμές των δεδομένων.

Ένα από τα κύρια πλεονεκτήματα του SMAPE είναι η συμμετρική του φύση, καθώς αποδίδει ίση σημασία στο σφάλμα της υπερεκτίμησης και της υπο-εκτίμησης. Αυτή η ιδιότητα είναι σημαντική κατά την ανάλυση χρονοσειρών, καθώς οι συνέπειες των ανωτέρων σφαλμάτων απαιτούν ισορροπημένη αντιμετώπιση. Η χρήση του SMAPE εξασφαλίζει ότι το μοντέλο δέχεται ισότιμη αρνητική ανατροφοδότηση σε κάθε περίπτωση, συμβάλλοντας έτσι στη βελτίωση της απόδοσής του.

3.2.6 Coefficient of Determination R^2

Το Coefficient of Determination R^2 ορίζεται ως το ποσοστό της μεταβολής της εξαρτημένης (προβλέψιμης) μεταβλητής ως προς τις ανεξάρτητες μεταβλητές (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023). Όταν η τιμή του R^2 τείνει στο 1, το μοντέλο προσαρμόζεται καλά στα δεδομένα. Η μετρική Coefficient of Determination ορίζεται με τον τύπο

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

όπου $\bar{y}$ είναι η μέση τιμή των προβλέψεων y_i.

3.2.7 Adjusted R^2

Το Adjusted R^2 βασίζεται στο Coefficient of Determination R^2. Έχει τροποποιηθεί ώστε να λαμβάνει υπόψη τον αριθμό των προβλέψεων στο μοντέλο, και δίδεται από το κάτωθι τύπο:

$$\text{Adjusted } R^2 = 1 - \left(\frac{(1 - R^2)(n - 1)}{n - k - 1}\right)$$

Όπου n είναι ο αριθμός των παρατηρήσεων και k ο αριθμός των ανεξάρτητων μεταβλητών του μοντέλου. Η τιμή του Adjusted R^2 υποδηλώνει πόσο καλά η ανεξάρτητη μεταβλητή ερμηνεύει το μοντέλο: επί παραδείγματι, αν η τιμή του Adjusted R^2 είναι 70%, αυτό σημαίνει ότι η ανεξάρτητη μεταβλητή ερμηνεύει το μοντέλο σε ποσοστό 70%. Με άλλα λόγια, η τιμή της μετρικής αυξάνεται μόνον εάν η ενσωμάτωση νέων όρων βελτιώνει το μοντέλο, ενώ μειώνεται εάν ένας προγνωστικός παράγοντας δεν προσθέτει αξία στο μοντέλο. Θα πρέπει να σημειωθεί ότι όταν το Adjusted R^2 λαμβάνει αρνητική τιμή, υποδεικνύει ότι το μοντέλο δεν προσαρμόζεται καλά στα δεδομένα (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023).

Αυτή η προσέγγιση αποδεικνύεται ιδιαίτερα χρήσιμη σε μοντέλα πολλαπλής παλινδρόμησης (multiple regression), όπου γίνεται ταυτόχρονη χρήση πολλών προγνωστικών παραγόντων.

Το Adjusted R^2 σε αντίθεση με το Coefficient of Determination R^2, χρησιμοποιείται ευρέως για τη σύγκριση μοντέλων με διαφορετικά πλήθη μεταβλητών, καθώς η τιμή του δεν αυξάνεται με την προσθήκη επιπρόσθετων μεταβλητών στο μοντέλο.

3.3 Μετρικές απόδοσης για εργασίες επεξεργασίας εικόνας

3.3.1 Inception Score (IS)

Η μετρική Inception Score (IS) χρησιμοποιείται ευρέως για την αξιολόγηση της ποιότητας των εικόνων που δημιουργούνται από μοντέλα AI, όπως τα Generative Adversarial Networks (GANs). Μετρά ποσοτικά τόσο την ποικιλομορφία όσο και την ποιότητα (αληθοφάνεια) των παραγόμενων εικόνων.

Για το υπολογισμό του **IS,** οι παραγόμενες εικόνες τροφοδοτούνται στον ταξινομητή Inception-v3 ο οποίος έχει εκπαιδευτεί στο σύνολο δεδομένων ImageNet. Ο Inception-v3 προσδιορίζει με υψηλή ακρίβεια, την πιθανότητα της εκάστοτε εικόνας να ανήκει σε μια κατηγορία του ImageNet. Αφότου εκτιμηθούν οι πιθανότητες της εκάστοτε εικόνας να ανήκει στην κάθε μία από τις κατηγορίες, αυτές συναθροίζονται με τη χρήση της Softmax, μετατρέποντας το διάνυσμα των πιθανοτήτων ένταξης σε κατηγορίες σε κατανομή πιθανότητας K πιθανών αποτελεσμάτων.

Στη συνέχεια, για κάθε εικόνα, υπολογίζεται ο μέσος όρος των πιθανοτήτων Softmax για όλες τις κατηγορίες. Ο μέσος όρος των πιθανοτήτων ονομάζεται "Οριακή κατανομή" (marginal distribution). Οι συνολικές βαθμολογίες καθορίζουν την ποιότητα των εικόνων, ενώ η εντροπία αυτών των βαθμολογιών προσδιορίζει την ποικιλομορφία των παραγόμενων εικόνων (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023).

Το IS δίδεται από τον ακόλουθο τύπο:

$$IS\ (G) = \exp[E_{x\sim p_{g(x)}}\, D_{KL}\big(\, p(y|x)\big) \,\|\, p(y))]$$

Όπου το $p_{g(x)}$ είναι η κατανομή των παραγόμενων εικόνων, $p(y|x)$ είναι η υπό συνθήκη κατανομή των κλάσεων, και $p(y)$ είναι η οριακή πιθανότητα της κάθε κλάσης.

Το βασικά πλεονεκτήματα της μετρικής IS είναι τα ακόλουθα: α) Λαμβάνει υπόψη δύο σημαντικούς παράγοντες για την αξιολόγηση της ποιότητας των παραγόμενων εικόνων, την ποιότητα και την ποικιλομορφία. β) Μπορεί να υπολογιστεί με τη χρήση προ-εκπαιδευμένων μοντέλων Inception που δεν απαιτούν τη γνώση της πραγματικής κατανομής των δεδομένων ως είσοδο.

Παρόλα αυτά, η μετρική IS έχει επίσης ορισμένα αρκετά μειονεκτήματα:

1. Η υψηλή εξάρτηση του IS με τον ταξινομητή Inception-v3 περιορίζει την χρησιμότητα του σε τομείς που έχουν σημαντικές διαφορές από τα περιεχόμενα του ImageNet.
2. Ο συντελεστής IS μπορεί να είναι υψηλός ακόμα και εάν οι εικόνες δεν είναι αληθοφανείς, αρκεί οι παραγόμενες εικόνες να έχουν υψηλή ποικιλομορφία.
3. Το Inception Score μπορεί να ευνοήσει μοντέλα που παράγουν παρόμοιες εικόνες.
4. Το IS αδυνατεί να διακρίνει με ακρίβεια τη συχνότητα εμφάνισης των διαφορετικών χαρακτηριστικών που παράγονται από ένα μοντέλο γενετικής αρχιτεκτονικής, με αποτέλεσμα η τιμή του IS να παραμένει υψηλή και να οδηγήσει σε παραπλανητική αξιολόγηση του μοντέλου.

3.3.2 Structural Similarity Index (SSIM)

Η μετρική Structural Similarity Index (SSIM) μετράει το ποσοστό ομοιομορφίας της εικόνας που παράγεται από το μοντέλο εικόνας ως προς μία εικόνα αναφοράς.

Σε αντίθεση με άλλες μεθόδους που εστιάζουν στα σφάλματα μεταξύ της εξεταζόμενης εικόνας και της εικόνας αναφοράς, ο SSIM θεωρεί τις αλλαγές στις δομικές πληροφορίες ως το κύριο στοιχείο που επηρεάζει την οπτική ποιότητα (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023), ως εκ τούτου βασίζεται στου ακόλουθους παράγοντες:

- Φωτεινότητα (luminance): Η μέση φωτεινότητα των pixel στις εικόνες.
- Αντίθεση (contrast): Η διαφορά φωτεινότητας μεταξύ γειτονικών pixel.
- Δομή (structure): Η ομοιότητα στη διάταξη των pixel στις εικόνες.

Το SSIM λαμβάνει τιμές από -1 έως 1. Το 1 υποδηλώνει τέλεια ομοιότητα, το 0 καμία ομοιότητα, και το -1 υποδηλώνει ότι οι εικόνες είναι αντίθετες.

Για τον υπολογισμό του δείκτη SSIM, θα πρέπει πρώτα να υπολογιστούν η μέση τιμή και η τυπική απόκλιση κάθε εικόνας, η διασταυρούμενη συνδιακύμανση μεταξύ των δύο εικόνων καθώς και το γινόμενο των τυπικών αποκλίσεων. Ο SSIM δίδεται από τον ακόλουθο τύπο:

$$SSIM(X,Y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Όπου μ_x, μ_y είναι οι μέσες τιμές των εικόνων x και y, σ_x, σ_y είναι οι τυπικές αποκλίσεις των εικόνων x και y, και σ_{xy} η διασταυρούμενη συν-διακύμανση μεταξύ των εικόνων x και y, ενώ C_1, C_2 είναι σταθερές που χρησιμοποιούνται για την αποφυγή αστάθειας.

Ο SSIM θεωρείται πιο αξιόπιστη επιλογή από την Peak Signal-to-Noise Ratio (PSNR) και την Mean Squared Error (MSE), καθώς είναι πιο ανθεκτική στις αλλαγές της φωτεινότητας και της αντίθεσης. Παρόλα αυτά, το SSIM θα πρέπει να εφαρμόζεται σε συνδυασμό με άλλες μετρικές, όπως το Inception Score (IS) και η Fréchet Inception Distance (FID), καθώς σκοπός του μοντέλου δεν είναι να δημιουργήσει ακριβές αντίγραφο της εικόνας εισόδου, αλλά να κατανοήσει την υποκείμενη κατανομή δεδομένων.

Αυτή η μετρική έχει βρει εφαρμογές σε μοντέλα τόσο σε μοντέλα επεξεργασίας όσο και σε μοντέλα δημιουργίας εικόνας και βίντεο, όπως το DiffusionCLIP , VSRResFeatGAN, mDCSRN, alignDRAW, και το SRGAN.

3.3.3 Fréchet Inception Distance (FID)

Η μετρική Fréchet Inception Distance (FID) χρησιμοποιείται για την αξιολόγηση της ποιότητας των εικόνων που δημιουργούνται από ένα μοντέλο, όπως το GAN, συγκρίνοντας τις στατιστικές κατανομές των πραγματικών και των συνθετικών εικόνων.

Πιο συγκεκριμένα, μέσω ενός προ-εκπαιδευμένου μοντέλου νευρωνικού δικτύου, όπως το Inception v3, εξάγονται τα χαρακτηριστικά (features) του συνόλου των πραγματικών εικόνων καθώς του συνόλου των εικόνων που έχουν παραχθεί από το μοντέλο (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023). Μόλις εξαχθούν τα χαρακτηριστικά, υπολογίζεται η πολυμεταβλητή (multivariate) Gaussian κατανομή για τα χαρακτηριστικά του κάθε συνόλου εικόνων. Τέλος μετράται η απόσταση Fréchet των κατανομών μέσω του ακόλουθου τύπου:

$$FID\ (X', Y') = \left\|\mu_x - \mu_y\right\|^2 + Tr(\Sigma_x + \Sigma_y - 2(\Sigma_x\Sigma_y)^{1/2})$$

Όπου το X είναι το σύνολο των πραγματικών εικόνων, Y είναι το σύνολο των συνθετικών εικόνων, X', Y' είναι η πολύ-μεταβλητή Gaussian κατανομή για τις πραγματικές και τις συνθετικές εικόνες, με μέση τιμή μ_x, μ_y, αντίστοιχα, ενώ και Σ_x, Σ_y είναι οι πίνακες συν-διακύμανσης. Επιπρόσθετα, Tr είναι το ίχνος του πίνακα και $\left(\Sigma_x\Sigma_y\right)^{1/2}$ είναι η τετραγωνική ρίζα του γινομένου των πινάκων συνδιακύμανσης.

Όσο χαμηλότερη είναι η τιμή της FID, τόσο πιο κοντά βρίσκονται οι δύο κατανομές, και επομένως τόσο καλύτερο είναι το μοντέλο GAN.

Το πλεονέκτημα της FID είναι ότι λαμβάνει υπόψιν τόσο την μέση τιμή όσο και τη συνδιακύμανση, καθιστώντας την καλύτερη μετρική σε σχέση με τις υπόλοιπες, καθώς οι λοιπές βασίζονται μόνο σε μια παράμετρο. Ωστόσο, το κύριο μειονέκτημά της είναι η εξάρτησή της από το δίκτυο Inception, καθιστώντας την αποτελεσματική μόνο σε εργασίες που αντιστοιχούν στο σύνολο δεδομένων με το οποίο έχει εκπαιδευτεί το δίκτυο αυτό (δηλ. το ImageNet).

Η FID έχει χρησιμοποιηθεί για την αξιολόγηση μοντέλων που σχετίζονται με τις εικόνες όπως το DDPM, BigGAN και Muse, καθώς και για την αξιολόγηση μοντέλων δημιουργίας βίντεο όπως το PHENAKI.

3.3.4 Zero-Shot FID (Fréchet Inception Distance)

Η Zero-Shot FID (Fréchet inception distance) αποτελεί παραλλαγή της FID. Χρησιμοποιείται για την αξιολόγηση της ποιότητας των εικόνων που παράγονται από διαφορετικά σύνολα δεδομένων, χωρίς να απαιτείται στάδιο προ-εκπαίδευσης, εξ ου και ο όρος «zero-shot» (Bandi, Pydi, & Yudu, 2023). Όπως γίνεται αντιληπτό, οι παραγόμενες εικόνες δεν ανήκουν σε προκαθορισμένες κλάσεις εκπαίδευσης, αλλά δημιουργούνται με βάση περιγραφές κειμένου ή άλλες μορφές εισόδου που διαφέρουν από τα δεδομένα εκπαίδευσης. Χαρακτηριστικό παράδειγμα εφαρμογής αυτής της μετρικής είναι το μοντέλο GLIDE, το οποίο χρησιμοποιείται για τη δημιουργία εικόνων, και το μοντέλο IMAGEN Video, που χρησιμοποιείται για τη σύνθεση βίντεο.

3.3.5 Multi-Scale Structural Similarity Index Measure (MS-SSIM)

Η μετρική multi-scale structural similarity index measure (MS-SSIM) μετρά την ομοιότητα δύο εικόνων, και όπως δηλώνει και το όνομα της, αποτελεί παραλλαγή της μετρικής SSIM. Χρησιμοποιείται ευρέως για την αξιολόγηση ποιότητας των εικόνων και βασίζεται στην υποβάθμιση των δομικών τους χαρακτηριστικών.

Η βασική διαφορά του SSIM σε σχέση με την MS-SSIM, είναι ότι η SSIM συγκρίνει τη θετική εικόνα με την εικόνα αναφοράς μόνο μια φορά (Bandi, Pydi, & Yudu, 2023), ενώ η MS-SSIM συγκρίνει τις εικόνες περισσότερες από δυο φορές στα διάφορα επίπεδα (scales) για να εξάγει το τελικό αποτέλεσμα, παρέχοντας έτσι πιο αξιόπιστες μετρήσεις. Ένα ακόμα πλεονέκτημα της SSIM είναι ότι μπορεί να χρησιμοποιηθεί για να συγκρίνει εικόνες που έχουν διαφορετικό μέγεθος (size), ανάλυση (resolution), συνθήκες θέασης (viewing conditions).

Πιο συγκεκριμένα, η μετρική MS-SSIM λειτουργεί λαμβάνοντας ως είσοδο την εικόνα αναφοράς και την εικόνα-στόχο (μπορεί να είναι παραμορφωμένη εικόνα). Αρχικά, ο αλγόριθμος εφαρμόζει επαναληπτικά ένα φίλτρο χαμηλής συχνότητας για να μειώσει την ανάλυση της φιλτραρισμένης εικόνας κατά παράγοντα 2 (Nilsson & Akenine-Möller, 2020). Στη συνέχεια, στην εικόνα αναφοράς δίδεται το επίπεδο Scale 1, ενώ το υψηλότερο επίπεδο είναι το «Scale M», το οποίο προκύπτει επαναλαμβάνοντας την παραπάνω διαδικασία M-1 φορές. Τέλος, σε κάθε κλίμακα, υπολογίζεται η αντίθεση $c_j(x,y)$ και η δομική ομοιότητα $s_j(x,y)$, ενώ η σύγκριση της φωτεινότητας υπολογίζεται μόνο στην κλίμακα «Scale M» και συμβολίζεται με $l_M(x,y)$. Το MS-SSIM προκύπτει από τον συνδυασμό των μετρήσεων που λήφθηκαν από όλα τα στάδια του αλγορίθμου και δίδεται από τον ακόλουθο τύπο:

$$\text{MS-SSIM}(x,y) = [l_M(x,y)]^{a_M} \prod_{j=1}^{M} [c_j(x,y)]^{\beta_j} [s_j(x,y)]^{y_j}$$

Όπου ο δείκτης M αντιστοιχεί στη χαμηλότερη ανάλυση που θεωρήθηκε και ο δείκτης 1 στην υψηλότερη, και για κάθε j ισχύει $\alpha_j = \beta_j = \gamma_j$

Το MS-SSIM παίρνει τιμές από -1 εώς 1, το 1 υποδηλώνει ότι η ομοιότητα είναι τέλεια, το 0 υποδηλώνει ότι δεν υπάρχει καμία ομοιότητα ανάμεσα στις εικόνες και το -1 υποδηλώνει αντίστροφη ομοιότητα.

Η μετρική έχει χρησιμοποιηθεί για να αξιολογήσει το μοντέλο δημιουργίας εικόνων TAC-GAN.

3.3.6 Learned Perceptual Image Patch Similarity (LPIPS)

Οι παραδοσιακές μετρικές όπως η SSIM και η PSNR μετρούν αποτελεσματικά την ομοιότητα των εικόνων, αλλά δεν λαμβάνουν υπόψη τους τον τρόπο με τον οποίο οι άνθρωποι βλέπουν τις εικόνες (Bandi, Pydi, & Yudu, 2023). Για την αντιμετώπιση αυτού του προβλήματος, οι ερευνητές εισήγαγαν την μετρική LPIPS μέσω του οποίου δύναται να μοντελοποιηθεί έως ένα βαθμό η οπτική αντίληψη του ανθρώπου.

Αναλυτικότερα, η Learned Perceptual Image Patch Similarity (LPIPS) αρχικά εξάγει τα χαρακτηριστικά των εικόνων, κάνοντας χρήση βαθιών νευρωνικών δικτύων τα οποία είναι εκπαιδευμένα σε ένα μεγάλο σύνολο εικόνων, όπως το AlexNet και το VGG. Τα χαρακτηριστικά αυτά δίνουν πληροφορίες για τις οπτικές δομές και τα μοτίβα που εντοπίζονται στις εικόνες και πρακτικά αντικατοπτρίζουν τον τρόπο με τον οποίο το οπτικό σύστημα του ανθρώπου επεξεργάζεται τις εικόνες. Στη συνέχεια, συγκρίνονται τα εξαγμένα χαρακτηριστικά των εικόνων, λαμβάνοντας υπόψη τόσο τοπικές όσο και καθολικές διαφορές.

Όταν το LPIPS τείνει στο 0, τα τμήματα της εικόνας (ή εικονίδια, image patches) είναι παρόμοια, ενώ όταν η τιμή του LPIPS είναι υψηλή οι εικόνες θεωρούνται ανόμοιες.

Παρά την υψηλή της αποτελεσματικότητα, η LPIPS παρουσιάζει τα ακόλουθα μειονεκτήματα:

- **Υπολογιστικό κόστος:** Η εκτέλεση της LPIPS μπορεί να είναι χρονοβόρα, ιδιαίτερα για εικόνες υψηλής ανάλυσης.
- **Εξάρτηση από τα δεδομένα εκπαίδευσης**: Η ακρίβεια της LPIPS επηρεάζεται από τα δεδομένα εκπαίδευσης που χρησιμοποιήθηκαν κατά την εκπαίδευση του νευρωνικού δικτύ.
- **Περιορισμένη γενίκευση:** Η LPIPS δεν είναι αποτελεσματική σε εξειδικευμένες εφαρμογές, όπως η ανίχνευση αλλαγών σε ιατρικές εικόνες.

Η μετρική έχει χρησιμοποιηθεί για να εκτιμηθεί η απόδοση μοντέλων επεξεργασίας εικόνων μέσω οδηγιών (prompt), όπως είναι το μοντέλο DiffusionCLIP και το DIFFEDIT.

3.3.7 Directional CLIP Similarity (Sdir)

Η μετρική Directional CLIP Similarity χρησιμοποιείται για να αξιολογήσει την ομοιότητα δύο εικόνων οι οποίες έχουν υποβληθεί σε επεξεργασία μέσω εντολών κειμένου (text-driven image manipulation). Η μετρική Sdir βασίζεται στο cross-modal μοντέλο ανάκτησης CLIP (Contrastive Language-Image Pretraining), μέσω του οποίου κατανοεί τη σχέση μεταξύ εικόνων και των περιγραφών τους (Bandi, Pydi, & Yudu, 2023). Αξίζει να σημειωθεί ότι το μοντέλο αυτό έχει εκπαιδευτεί σε δεδομένα που περιλαμβάνουν ζεύγη (εικόνα, λεζάντα) τα οποία συλλέχθηκαν από το διαδίκτυο, και έχει συνολικό μέγεθος 400M.

Υψηλή βαθμολογία Sdir υποδεικνύει ότι οι δύο εικόνες παρουσιάζουν σημαντική ομοιότητα και ότι οι οδηγίες που δόθηκαν (υπό την μορφή κειμένου), για την επεξεργασία την εικόνας εφαρμόστηκαν με επιτυχία από το μοντέλο. Η Sdir μετρική αξιολόγησης έχει χρησιμοποιηθεί στο μοντέλο DiffusionCLIP.

3.3.8 Dice Loss ή Dice similarity coefficient

Η μετρική Dice Loss, χρησιμοποιείται (μεταξύ άλλων) για την αξιολόγηση της ομοιότητας μεταξύ της προβλεπόμενης μάσκας τμηματοποίησης (segmentation mask) και της μάσκας αναφοράς, η οποία θα αποτελεί την «απόλυτη αλήθεια» (ground truth) (Bandi, Pydi, & Yudu, 2023). Η μετρική Dice Loss υπολογίζεται από τον ακόλουθο τύπο:

$$L = 1 - \frac{2 \cdot intersection\,(pred, gt)}{|pred| + |gt|}$$

όπου $pred$ είναι η προβλεπόμενη μάσκα τμηματοποίησης, gt η μάσκα που έχει οριστεί ως ground truth, $intersection\,(pred, gt)$ είναι ο συνολικός αριθμός των pixel που βρίσκονται στην τομή της μάσκας που προβλέπεται από το μοντέλο και μάσκας αναφοράς, $|pred|$ είναι ο συνολικός αριθμός των pixel των προβλεπόμενων μασκών τμηματοποίησης, και $|gt|$ είναι ο συνολικός αριθμός των pixel των μασκών που έχουν οριστεί ως μάσκα αναφοράς (Rainio, Teuho, & Klén, 2024).

Όταν η μετρική είναι ίση με 1, η τμηματοποίηση είναι τέλεια, ενώ όταν η τιμή του Dice είναι 0, οι μάσκες δεν επικαλύπτονται και επομένως η τμηματοποίηση είναι κακή.

Το Dice similarity coefficient χρησιμοποιείται για την αξιολόγηση generative AI μοντέλων. Επίσης, εφαρμόζεται ευρέως στην ιατρική απεικόνιση, όπου είναι αναγκαία η κατάτμηση δομών σε εικόνες με υψηλή ακρίβεια. Χαρακτηριστικό παράδειγμα αποτελεί το SegAN, όπου έχει εκπαιδευτεί να τμηματοποιεί τις ακτινογραφίες των ασθενών.

3.3.9 Peak Signal-to-Noise Ratio (PSNR)

Η μετρική Peak Signal-to-Noise Ratio χρησιμοποιείται συχνά για την αξιολόγηση της ποιότητας της εικόνας αναφοράς ως προς την επεξεργασμένη εικόνα, από την οποία το μοντέλο έχει αφαιρέσει τον θόρυβο (Terven, Córdova-Esparza, Ramírez-Pedraza, & Chávez-Urbiola, 2023).

Η μετρική PSNR βασίζεται στο Mean Square Error (MSE). Το MSE για δύο μονόχρωμες εικόνες I (αρχική θορυβώδης εικόνα) και K (η εικόνας της οποίας ο θόρυβος έχει αφαιρεθεί) οι οποίες έχουν διαστάσεις m × n, ορίζεται ως:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

Το PSNR υπολογίζεται με χρήση του MSE από το ακόλουθο τύπο:

$$PSNR = 10 \cdot log_{10}\left(\frac{(MAX_{\mathrm{I}})^2}{MSE}\right)$$

Όπου το MAX_{I} μεγίστη δυνατή τιμή των pixel μιας εικόνας. Για παράδειγμα, σε μια ασπρόμαυρη εικόνα 8bit, το MAX_{I} είναι 255.

Μονάδα μέτρησης του PSNR είναι το dB. Υψηλότερο PSNR συνήθως υποδηλώνει ότι η ανακατασκευασμένη εικόνα είναι υψηλής ποιότητας. Θα πρέπει να σημειωθεί ότι σε ορισμένες περιπτώσεις, παρόλο που συγκρινόμενες εικόνες είναι παρόμοιες και ο θόρυβος έχει αφαιρεθεί, το PSNR μπορεί να είναι χαμηλό, καθώς μικρά σφάλματα στα pixel είναι πιθανόν να μην γίνουν αντιληπτά από τη δομική ομοιότητα.

Η μετρική έχει εφαρμοστεί σε μοντέλα που αφαιρούν το θόρυβο από τις εικόνες, όπως το DeblurGAN και στο DeblurGAN-v2, έτσι ώστε να αξιολογηθεί η ποιότητα ανακατασκευασμένης εικόνας.

3.3.10 Normalized Root Mean Square Error (NRMSE) ή scatter index

Το Normalized Root Mean Square Error, ή scatter index χρησιμοποιείται συνήθως για να αξιολογήσει την ακρίβεια των προβλέψεων ενός μοντέλου, όταν η έξοδος του (output) είναι μια αριθμητική τιμή (Bandi, Pydi, & Yudu, 2023).

Πιο αναλυτικά, η NRMSE κανονικοποιεί την μετρική RMSE, αυξάνοντας την αντιπροσωπευτικότητά της για τις περιπτώσεις που απαιτείται να συγκριθούν διαφορετικά σύνολα δεδομένων. Το NRMSE υπολογίζεται συνήθως από τον λόγο του RMSE προς την τυπική απόκλιση των πραγματικών τιμών (σ), και εκφράζεται ως ποσοστό.

$$NRMSE = \frac{RMSE}{\sigma}$$

Θα πρέπει να σημειωθεί, ότι στη βιβλιογραφία υπάρχουν διάφοροι τύποι για την κανονικοποίηση του RMSE. Η μετρική RMSE μπορεί να κανονικοποιηθεί με διάφορους τρόπους, συμπεριλαμβάνοντας:

- Τη μέση τιμή ($\bar{y}$):

$$NRMSE = \frac{RMSE}{\bar{y}}$$

- Τη διαφορά μεταξύ μέγιστης και ελάχιστης τιμής:

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}}$$

- Το διατεταρτημοριακό εύρος (interquartile range)

$$NRMSE = \frac{RMSE}{Q_1 - Q_2}$$

Εάν δεν υπάρχουν πολλές ακραίες τιμές στις μεταβλητές απόκρισης, τότε ενδείκνυται η χρήση του διατεταρτημοριακού εύρους. Μικρή τιμή NRMSE, υποδηλώνει ότι οι προβλέψεις του μοντέλου είναι κοντά στις πραγματικές τιμές. Τέλος, το NRMSE έχει χρησιμοποιηθεί για να αξιολογήσει εικόνες υψηλής ανάλυσης οι οποίες έχουν παραχθεί από το μοντέλο mDCSRN.

3.3.11 Mean Opinion Score (MOS)

Το Mean Opinion Score είναι μια υποκειμενική μετρική. Χρησιμοποιείται συνήθως για την αξιολόγηση εικόνων, των οποίων ο θόρυβος έχει αφαιρεθεί (deblurring) ή ανάλυση τους έχει βελτιωθεί από ένα μοντέλο generative AI (Bandi, Pydi, & Yudu, 2023). Η αξιολόγηση των εικόνων δεν γίνεται με βάση κάποιον τύπο, αλλά βασίζεται σε μια ομάδα αξιολογητών, οι οποίοι αναθέτουν σε κάθε εικόνα μια τιμή, 1 έως 5. Όταν η τιμή της εικόνας είναι ίση με 1, η ποιότητα της είναι κακή, ενώ όταν η βαθμολογία της είναι 5, η εικόνα είναι τέλεια. Η MOS μετρική έχει χρησιμοποιηθεί στο μοντέλο SRGAN προκειμένου να αξιολογηθούν οι παραγόμενες εικόνες υψηλής ανάλυσης (HR).

3.3.12 Fully Convolutional Network Score (FCN-Score)

Η μετρική FCN-Score χρησιμοποιείται για την αξιολόγηση της ποιότητας των εικόνων που δημιουργούνται από μοντέλα cGAN (Conditional Generative Adversarial Network).

Το FCN-Score βασίζεται στο μοντέλο FCN, το οποίο δέχεται ως είσοδο τον **χάρτη τμηματοποίηση** (segmentation map) της εικόνας αναφοράς και της παραγόμενης εικόνας από το μοντέλο cGAN (Bandi, Pydi, & Yudu, 2023). Πρακτικά, ο χάρτης τμηματοποίησης καθορίζει το είδους των αντικειμένων πού πρέπει να εμφανιστούν στην τελική εικόνα (π.χ. λευκό για φόντο, κίτρινο για ήλιο, καφέ για γάτα). Στη συνέχεια, το μοντέλο FCN συγκρίνοντας τα χαρακτηριστικά των εικόνων, μέτρα τις διαφορές στις κατανομές τους.

Χαμηλή τιμή FCN υποδηλώνει ότι το μοντέλο έχει καλή απόδοση, και επομένως οι παραγόμενες εικόνες μοιάζουν πολύ με τις πραγματικές. Αντίθετα, όταν το FCN έχει υψηλή τιμή, υποδεικνύεται ότι το μοντέλο είναι λιγότερο αποτελεσματικό και οι επομένως υπάρχουν μεγάλες αποκλίσεις μεταξύ των συγκρινόμενων εικόνων.

3.3.13 Realism Score

Η μετρική «Realism Score» χρησιμοποιείται για την αξιολόγηση της ποιότητας των εικόνων που έχουν υποστεί επεξεργασία από ένα μοντέλο AI, για παράδειγμα, είτε μέσω Image inpainting ή Image Blending (Bandi, Pydi, & Yudu, 2023). Πιο συγκεκριμένα, αξιολογεί το κατά πόσο η τροποποιημένη εικόνα διαφέρει από την εικόνα ή τις εικόνες αναφοράς. Όταν η τιμή της μετρικής είναι υψηλή, οι τροποποιήσεις έχουν ενσωματωθεί καλά στην εικόνα, και είναι σχεδόν αδύνατο

να εντοπιστούν διαφορές μεταξύ της τροποποιημένης και της αρχικής εικόνας. Η μετρική Realism score έχει εφαρμοστεί στο GP-GAN για να αξιολογηθούν οι εικόνες εξόδου.

3.4 Μετρικές απόδοσης εργασιών NLP

3.4.1 BLEU (Bilingual Evaluation Understudy)

Η μετρική BLEU αξιολογεί την ομοιότητα της πρότασης που παράγεται ή μεταφράζεται από το μοντέλο ως προς την πρόταση αναφοράς (Google, 2024). Η BLEU παίρνει τιμές από 0 έως 1. Εάν η μετρική τείνει στο 0, υποδηλώνει ότι μετάφραση (του μοντέλου) δεν έχει καμία επικάλυψη (overlap) με τη μετάφραση αναφοράς και επομένως το μοντέλο δεν είναι αποτελεσματικό. Αντίθετα, όταν η τιμή του BLEU τείνει στο 1, ότι υποδηλώνει ότι μεταφράσεις επικαλύπτονται και επομένως το μοντέλο είναι αποτελεσματικό. Θα πρέπει να σημειωθεί, ότι ακόμη και μεταφράσεις που γίνονται από ανθρώπους δεν επιτυγχάνουν τέλεια βαθμολογία (ίση με 1,0).

Μαθηματικά η μετρική BLEU υπολογίζεται από τον ακόλουθο τύπο:

$$BLUE = \underbrace{min\left(1, exp\left(1 - \frac{\text{reference-length}}{\text{output-length}}\right)\right)}_{\text{Brevity penalty}} \underbrace{\left(\prod_{i=1}^{4} precision_i\right)^{1/4}}_{\text{n-gram overlap}}$$

όπου η ποσότητα precision ορίζεται ως:

$$precision_i = \frac{\sum_{snt \in \text{Cand-Corpus}} \sum_{i \in snt} \min\left(m^i_{cand}, m^i_{ref}\right)}{w^i_t = \sum_{snt \in \text{Cand-Corpus}} \sum_{i' \in snt'} m^{i'}_{cand}}$$

όπου:

- m^i_{cand}: είναι ο συνολικός αριθμός (i-gram) των μεταφρασμένων γραμμάτων που είναι ίδια με τη μετάφραση αναφοράς
- m^i_{ref}: είναι ο συνολικός αριθμός (i-gram) των γραμμάτων της μετάφραση αναφοράς.
- w^i_t: είναι ο συνολικός αριθμός (i-gram) των μεταφρασμένων από μοντέλο γραμμάτων.

Η μετρική BLEU αποτελείται από δύο μέρη:

- **Brevity penalty:** Η μετρική BLEU μέσω του όρου *brevity penalty*, τιμωρεί τις μεταφράσεις που έχουν πολύ μικρότερο μήκος σε σχέση με την μετάφραση αναφοράς
- **n-gram overlap:** Η μετρική BLEU μέσο του όρου "n-gram overlap" υπολογίζει τον συνολικό αριθμό των unigrams, bigrams, trigrams, και four-grams (i=1, ..., 4) και στη συνέχεια τα συγκρίνει ως προς τον αριθμό των n-gram που υπάρχουν στη μετάφραση αναφοράς.
- Το **n-gram overlap** δρα ως μετρική ακρίβειας στην συνάρτηση. Τα unigrams αντιπροσωπεύουν την επάρκεια, ενώ n-gram που αποτελούνται πολλές λέξεις, αντιπροσωπεύουν την ευχέρεια της μετάφρασης. Ο αριθμός των n-gram έχει περιοριστεί στις 4 λέξεις, για να αποφευχθεί η υπερβολική καταμέτρηση (overcounting).

Η μετρική BLEU έχει χρησιμοποιηθεί για την αξιολόγηση των ακόλουθων μοντέλων AI: T5, WT5, MusCaps, RTT-GAN, Seq2Seq, "Show and Tell: Neural Image Captioning" και ConvS2S.

3.4.2 Metric for Evaluation of Translation with Explicit Ordering (METEOR)

Η μετρική METEOR χρησιμοποιείται για την αξιολόγηση της ποιότητας της μετάφρασης ενός κειμένου. Η μετρική δεν περιορίζεται απλώς στην επικάλυψη λέξεων πρόβλεψης και αναφοράς, όπως η BLEU, αλλά λαμβάνει επίσης υπόψη της και άλλα χαρακτηριστικά όπως είναι η σημασία των λέξεων (synonym matching), τη σειρά των λέξεων στην πρόταση, καθώς και το μήκος τους (stemming) (Bandi, Pydi, & Yudu, 2023).

Η METEOR αρχικά υπολογίζει τον αρμονικό μέσο όρο του Precision και του Recall, δίνοντας μεγαλύτερη βαρύτητα στο Recall σε σχέση με το Precision, και στη συνέχεια υπολογίζεται το Word Order Penalty (Lee, et al., 2023).

$$METEOR = \underbrace{FMean}_{\text{Harmonic Mean of Unigram Precision / Recall}} * \underbrace{(1 - Penalty)}_{\text{Word Order Penalty}}$$

Το FMean δίδεται από τον τύπο:

$$\text{FMean} = \frac{10PR}{R + 9P}$$

Όπου το P είναι το Precision των unigrams, και το R είναι το Recall των unigrams.

Όπως μπορεί να συναχθεί και από τον τύπο, ο όρος FMean βασίζεται μόνο στο Precision και το Recall των unigrams. Για να έχει τη δυνατότητα η METEOR να λάβει υπόψιν και μεγαλύτερες ακολουθίες, χρησιμοποίει την συνάρτηση *Word Order Penalty*, η οποία μετριάζει αυτή την αδυναμία, και ταυτόχρονα της επιτρέπει να κατανοήσει τη σειρά των λέξεων. Δίδεται από τον τύπο:

$$\text{Penalty } = 0.5 * \frac{\#of\ Chunks}{\#\ of\ Unigram\ Matched}$$

Για παράδειγμα, έστω ότι η πρόταση αναφοράς είναι «the president then spoke to the audience», και το μοντέλο έχει δημιουργήσει την ακόλουθη πρόταση (candidate sentence) «the president spoke to the audience». Τότε ο αριθμός των κομματιών (number of chunks) θα ήταν ίσος με 2, ("the president" και το "spoke to the audience") και αριθμός των unigrams θα είναι ίσος με 6 (Lee, et al., 2023). Μπορεί να γίνει αντιληπτό, ότι όσο μειώνονται ο αριθμός των chunk, τόσο μειώνεται και η ποινή, γεγονός που οδηγεί σε υψηλότερη βαθμολογία METEOR.

Η METEOR χρησιμοποιείται τόσο για την αξιολόγηση μοντέλων που αλληλοεπιδρούν με τους ανθρώπους, με τρόπο παρόμοιο με τις ανθρώπινες συνομιλίες, όπως είναι το GPT-3 και το T5, τόσο και σε image captioning μοντέλα, όπως είναι το ‘Show and Tell: Neural Image Captioning’, GTR-LSTM και το MusCaps.

3.4.3 CIDEr (Consensus-based Image Description Evaluation)

Η μετρική CIDEr χρησιμοποιείται για την αξιολόγηση της ποιότητας αναφορικά με τις λεζάντες των εικόνων οι οποίες παράγονται από μοντέλα AI, όπως είναι το RTT-GAN και το MusCaps. Η μετρική βασίζεται στο πρωτόκολλο consensus, το οποίο θεωρεί ότι οι καλές λεζάντες πρέπει να χρησιμοποιούν παρόμοιες λέξεις και γραμματική, καθώς και να έχουν παρόμοιο νόημα και περιεχόμενο ως προς τις λεζάντες αναφοράς (Vedantam, Zitnick, & Parikh, 2015).

Αρχικά, για κάθε εικόνα παρέχεται ένα σύνολο λεζάντων αναφοράς, το οποίο αποτελεί την «απόλυτη αλήθεια» (ground truth). Στη συνέχεια κάθε παραγόμενη λεζάντα συγκρίνεται το οικείο σύνολο από λεζάντες αναφοράς, χρησιμοποιώντας το σκορ BLEU (Bilingual Evaluation Understudy) για να αποτιμήσει την επικάλυψη τους. Οι βαθμολογίες που προκύπτουν από την

BLEU τροποποιούνται μέσω των βαρών IDF (Inverse Document Frequency). Σκοπός της IDF είναι να αυξήσει το βάρος των λέξεων που χρησιμοποιούνται σπάνια στις λεζάντες αναφοράς, αλλά εμπεριέχονται στις παραγόμενες λεζάντες. Το βάρος $g_k(s_{ij})$ για κάθε n-gram ω_k υπολογίζεται με τον ακόλουθο τύπο:

$$g_k(s_{ij}) = \frac{h_k(s_{ij})}{\sum_{\omega_l \in \Omega} h_l(s_{ij})} log\left(\frac{|I|}{\sum_{I_p \in I} min\left(1, \sum_q h_k(s_{pq})\right)}\right)$$

όπου το πλήθος των φορών που εμφανίζεται ένα n-gram ω_k σε μία πρόταση s συμβολίζεται με h(s).

Τέλος, υπολογίζεται η μετρική CIDEr. Ο υπολογισμός της μετρικής δίδεται από τον ακόλουθο τύπο:

$$CIDEr_n(c_i, S_i) = \frac{1}{m} \sum_j \frac{g^n(c_i) \cdot g^n(s_{ij})}{\|g^n(c_i)\| \|g^n(s_{ij})\|}$$

όπου $g^n(c_i)$ είναι το διάνυσμα που σχηματίζεται από το $g_k(c_i)$, το οποίο αντιστοιχεί σε όλα τα n-grams που έχουν μήκος n, και $\|g^n(c_i)\|$ το μέγεθος του του διανύσματος $g^n(c_i)$ (Vedantam, Zitnick, & Parikh, 2015).

Για να αποτυπωθούν οι γραμματικές ιδιότητες, καθώς και η σημασιολογία των προτάσεων θα πρέπει να χρησιμοποιηθούν n-gram υψηλότερης τάξης. Ο υπολογισμός του μπορεί να γίνει κάνοντας χρήση του ακόλουθου τύπου:

$$CIREr(c_i, S_i) = \sum_{n=1}^{N} w_n \, CIDEr_n(c_i, S_i)$$

Όπου w_n είναι το βάρος που αποδίδεται στα n-grams μήκους n. Οι (Vedantam, Zitnick, & Parikh, 2015) αναφέρουν ότι με βάση τα πειράματα που διεξήγαγαν, τα καλύτερα αποτελέσματα επιτυγχάνονται όταν χρησιμοποιούνται ομοιόμορφα βάρη, δηλ. w_n=1/N.

3.4.4 Median Rank (MdR)

Η μετρική Median Rank αποτελεί βασική μετρική αξιολόγησης των συστημάτων ανάκτησης πληροφορίας (information retrieval tasks). Ο υπολογισμός της μετρικής γίνεται σε τρία βήματα:

1. Αρχικά, το σύστημα λαμβάνει ένα ερώτημα (query) από τον χρήστη, και επιστρέφει μια λίστα με αντικείμενα, τα οποία ταξινομούνται με βάση τη συνάφειά τους ως προς το υποβληθέν ερώτημα (Bandi, Pydi, & Yudu, 2023). Το βήμα αυτό επαναλαμβάνεται πολλαπλές φορές.
2. Στη συνέχεια, για κάθε ερώτημα, εντοπίζεται και καταγράφεται η θέση (rank) του πιο συναφούς αντικείμενου της λίστας.
3. Αφού συλλεχθούν οι κατατάξεις (ranks) των αντικειμένων που έχουν την μεγαλύτερη συνάφεια για όλα τα ερωτήματα που χρησιμοποιήθηκαν στην αξιολόγηση, γίνεται ο υπολογισμός του Median Rank.

Όσο μικρότερη είναι η τιμή του Median Rank, τόσο καλύτερη είναι η απόδοση του συστήματος ανάκτησης.

3.4.5 EM-Diff (Exact Match Difference)

Η μετρική Exact Match Difference χρησιμοποιείται κυρίως για την αξιολόγηση συστημάτων question-answering (Bandi, Pydi, & Yudu, 2023). Βασίζεται στην ικανότητα των μοντέλων να παράγουν απαντήσεις που ταιριάζουν απόλυτα με την αναμενόμενη απάντηση (ground truth) .

Πιο συγκεκριμένα, για ένα ορισμένο σύνολο δεδομένων, η μετρική EM-Diff συγκρίνει τα σκορ ακριβείας (exact match) δύο διαφορετικών μοντέλων Conversational AI. Υπολογίζοντας τη διαφορά μεταξύ αυτών των σκορ, η μετρική φανερώνει πιο μοντέλο αποδίδει καλύτερα. Όταν το EM-Diff έχει υψηλή τιμή, υποδηλώνει ότι το μοντέλο έχει υψηλή απόδοση, ενώ όταν η τιμή του είναι μικρή, το μοντέλο έχει χαμηλή απόδοση.

Το κύριο μειονεκτήματα της EM-Diff είναι ότι λαμβάνει υπόψη μόνο τις απαντήσεις που ταυτίζονται (exact match) με της απαντήσεις αναφοράς (ground truth), αγνοώντας το εάν οι απαντήσεις είναι σημασιολογικά σωστές. Η μετρική αξιολόγησης EM-Diff έχει χρησιμοποιηθεί στο Conversational μοντέλο PEER.

3.4.6 BLEURT (Bilingual Evaluation Understudy for Natural Language Understanding in Translation)

Η μετρική BLEURT (Bilingual Evaluation Understudy for Natural Language Understanding in Translation) χρησιμοποιείται για την αξιολόγηση της ποιότητας των κειμένων/προτάσεων που

έχουν μεταφραστεί από ένα μοντέλο, όπως το InstructGPT. Το BLEURT μετρά κατά πόσο ομοιάζουν η μετάφραση αναφοράς (η οποία έχει δημιουργηθεί από άνθρωπο) με τη μετάφραση που έχει παραχθεί από το μοντέλο (Bandi, Pydi, & Yudu, 2023). Για να μπορέσει να μετρήσει την ομοιομορφία, το BLEURT βασίζεται σε μοντέλα Transformers.

Πιο συγκεκριμένα, η εκπαίδευση του BLEURT γίνεται σε τρία βήματα. Αρχικά, το μοντέλο μετασχηματισμού BERT, το οποίο αποτελεί και την βάση BLEURT, θα πρέπει να προ-εκπαιδευτεί σε ένα μεγάλο σύνολο δεδομένων κειμένου. Στη συνέχεια, πραγματοποιείται προ-εκπαίδευση του μοντέλου σε ένα μεγάλο όγκο συνθετικών δεδομένων, για να βελτιστοποιηθεί η γενική κατανομή του μοντέλου.

Τέλος, το μοντέλο βελτιστοποιείται (fine-tuning) σε δεδομένα που είναι ειδικά σχεδιασμένα για την εκάστοτε εργασία που θα εφαρμοστεί η μετρική BLEURT. Αυτό το βήμα επιτρέπει στο μοντέλο να προσαρμοστεί καλύτερα στις απαιτήσεις της εκάστοτε εργασίας που θα αξιολογήσει. Όσο πιο υψηλή είναι η τιμή BLEURT, τόσο καλύτερα το μοντέλο αποδίδει την μετάφραση.

3.5 Μετρικές απόδοσης εργασιών αναγνώρισης χαρακτήρων

3.5.1 Content Accuracy

Η μετρική Content accuracy χρησιμοποιείται για να αξιολογήσει την ποιότητα των χειρόγραφων χαρακτήρων που έχουν δημιουργηθεί από ένα μοντέλο AI.

Πιο συγκεκριμένα, για να μετρηθεί το Content accuracy χρησιμοποιείται το μοντέλο HCCR-GoogLeNet[9], το οποίο έχει εκπαιδευτεί αντλώντας δεδομένα από τη βάση CASIA-HWDB[10]. Η συγκεκριμένη βάση δεδομένων περιέχει κινέζικους χειρόγραφους χαρακτήρες, που έχουν συλλεχθεί από 1.020 συγγραφείς (Bandi, Pydi, & Yudu, 2023). Το Content accuracy, βασίζεται στην ακόλουθη λογική: εάν οι παραγόμενοι από το μοντέλο χειρόγραφοι χαρακτήρες είναι ρεαλιστικοί, τότε το μοντέλο HCCR-GoogLeNet θα μπορεί να τους κατηγοριοποιήσει σωστά. Επομένως, σωστή κατηγοριοποίηση των παραγόμενων χαρακτήρων, υποδηλώνει ότι το μοντέλο γένεσης χαρακτήρων είναι υψηλής ποιότητας. Η μετρική αυτή έχει χρησιμοποιηθεί για να για να

[9] https://github.com/zhongzhuoyao/HCCR-GoogLeNet
[10] https://nlpr.ia.ac.cn/databases/handwriting/Touching_Characters_Databases.html

αξιολογήσει την ποιότητα των χειρόγραφων κινέζικων χαρακτήρων που έχουν δημιουργηθεί από το μοντέλο DenseNet CycleGAN.

3.5.2 Style Discrepancy

Η μετρική Style Discrepancy, μετράει το κατά πόσο διαφέρει το στυλ των χαρακτήρων αναφοράς από το στυλ των χαρακτήρων που έχουν δημιουργηθεί από ένα μοντέλο AI, όπως το DenseNet CycleGAN (Bandi, Pydi, & Yudu, 2023).

Μαθηματικά ορίζεται ως η τετραγωνική ρίζα του μέσου τετραγωνικού σφάλματος (RMSE) μεταξύ του στυλ των χαρακτήρων αναφοράς και των παραγόμενων χαρακτήρων. Η συσχέτιση των χαρακτηριστικών τους δίνεται από τον πίνακα $G^l \in R^{N_l x N_l}$, όπου το N_l είναι ο αριθμός των φίλτρων στο επίπεδο υπ' αριθμόν l (l-th layer), και G_{ij}^l είναι το εσωτερικό γινόμενο μεταξύ των διανυσμάτων i και j στο επίπεδο l.

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l$$

Χαμηλή απόκλιση υποδηλώνει ότι το μοντέλο δημιουργεί χαρακτήρων οι οποίοι έχουν καλό στυλ.

3.5.3 Recognition Accuracy

Το Recognition Accuracy αξιολογεί το κατά πόσο οι χειρόγραφοι χαρακτήρες που παράγονται από ένα μοντέλο AI μοιάζουν με τους χαρακτήρες αναφοράς (Bandi, Pydi, & Yudu, 2023). Η αξιολόγηση πραγματοποιείται μετρώντας τον συνολικό αριθμό των παραχθέντων χαρακτήρων οι οποίοι μπορούν να αναγνωριστούν από ένα σύστημα OCR (Optical Character Recognition).

3.5.4 Diversity

Η μετρική Diversity μετρά το εύρος (range) και την ποικιλία (variation) των χειρόγραφων χαρακτήρων που έχουν δημιουργηθεί από ένα AI μοντέλο (Bandi, Pydi, & Yudu, 2023). Πρακτικά, το Divercity αξιολογεί την ικανότητα του μοντέλου να παράγει τόσο διακριτούς όσο και ποικιλόμορφους χαρακτήρες για μία ορισμένη γραμματοσειράς -στόχο, αποφεύγοντας τις επαναλήψεις ή τις υπερβολικές ομοιότητες μεταξύ των γραμμάτων.

3.6 Μετρικές απόδοσης μοντέλων δημιουργίας κώδικα

3.6.1 CodeBLEU

Η μετρική CodeBLEU χρησιμοποιείται για να αξιολογήσει την ποιότητα του κώδικα που έχει δημιουργήσει ένα μοντέλο γένεσης κώδικα βάσει NLP (NLP-to-code generation). Η CodeBLEU έχει βασιστεί στην μετρική BLEU, και πρακτικά μετράει την ομοιότητα μεταξύ του παραγόμενου κώδικα και του κώδικα αναφοράς (Ren, et al., 2020).

Μαθηματικά το CodeBLUE ορίζεται ως ο σταθμισμένος μέσος όρος επί μέρους μετρικών, βάσει του ακόλουθου τύπου:

$$\text{CodeBLEU} = \alpha \cdot BLEU + \beta \cdot BLEU_{weight} + \gamma \cdot BLEU_{ast} + \delta \cdot Match_{df}$$

Όπου η μεταβλητή BLEU υπολογίζεται από την κλασσική συνάρτηση BLEU (Papineni, Roukos, Ward, & Zhu, 2002), το $BLUE_{weight}$ αποτελεί μετρική που αφορά σταθμισμένα n-grams που είναι κοινά τόσο στον παραγόμενο κώδικα όσο και στον κώδικα αναφοράς, το $BLEU_{ast}$ είναι η συντακτική αντιστοιχία των αφηρημένων συντακτικών δένδρων (abstract syntax trees, AST), μέσω της οποίας ανιχνεύεται η συντακτική αντιστοιχία μεταξύ κωδίκων, και το $Match_{df}$ αποτελεί την αντιστοιχία ροής δεδομένων. Το $Match_{df}$ λαμβάνει υπόψη του την σημασιολογική ομοιότητα μεταξύ του παραγόμενου κώδικα και κώδικα αναφοράς. Τα σταθμισμένα (weighted) n-gram και το AST χρησιμοποιούνται για να μετρήσουν τη συντακτική ορθότητα του κώδικα, ενώ το $Match_{df}$ χρησιμοποιείται για να υπολογίσει την λογική του ορθότητα.

Η μετρική BLEU έχει χρησιμοποιηθεί για την αξιολόγηση του κώδικα που παράγεται από τα μοντέλα όπως το CodeBERT και CodeT5.

3.6.2 Exact Match (EM)

Η μετρική Exact Match χρησιμοποιείται για την αξιολόγηση μοντέλων παραγωγής κώδικα (Bandi, Pydi, & Yudu, 2023). Αναλυτικότερα, η μετρική ελέγχει εάν ο παραγόμενος κώδικας ταυτίζεται με τον κώδικα αναφοράς. Το κύριο μειονεκτήματά της είναι ότι δεν λαμβάνει υπόψη παράγοντες όπως η καθαρότητα, η συντομία, ή η αποδοτικότητα του κώδικα.

3.6.3 Pass@k

Η μετρική Pass@k χρησιμοποιείται ευρέως για την αξιολόγηση του κώδικα που παράγεται είτε από ένα γλωσσικό μοντέλο (language model) είτε από ένα εξειδικευμένο μοντέλο δημιουργίας κώδικα, όταν αυτό δέχεται ως είσοδο μια περιγραφή σε φυσική γλώσσα (prompt).

Πιο συγκεκριμένα, η μετρική αξιολογεί την λειτουργική ορθότητα του κώδικα (functional correctness). Βασίζεται στην ακόλουθη αρχή: Για κάθε ορισμένο (προς επίλυση) πρόβλημα, το AI μοντέλο παράγει k δείγματα κώδικα. Η Pass@k θεωρεί ότι το πρόβλημα έχει επιλυθεί, εάν τουλάχιστον μια από τις παρεχόμενες λύσεις περνάει με επιτυχία όλα τα unit test. Το κλάσμα των προβλημάτων που έχουν επιλυθεί από το μοντέλο προσδιορίζεται μαθηματικά από τον ακόλουθο τύπο:

$$\text{pass@k} := E_{Problems}\left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}\right]$$

Όπου n είναι ο συνολικός αριθμός των δειγμάτων, c ο συνολικός αριθμός των σωστών δειγμάτων, και k είναι ο αριθμός των δειγμάτων κώδικα ανά πρόβλημα. Το n θα πρέπει να είναι μεγαλύτερο ή ίσο του k.

Συνοψίζοντάς, η μέτρηση παρέχει πολύτιμες πληροφορίες τόσο για την ικανότητα του μοντέλου να παράγει κώδικα που είναι λειτουργικά σωστός, όσο και για το εάν πληροί τις προ-καθορισμένες απαιτήσεις. Έχει χρησιμοποιηθεί για την αξιολόγηση μοντέλων όπως το Codex της GPT, το οποίο βασίζεται στο μοντέλο Transformers, καθώς και στο Alphacode, προκειμένου να προσδιορίσουν το ποσοστό επιτυχίας τους, στην παραγωγή λειτουργικού κώδικα (Chen, et al., 2021).

3.6.4 Multi-Turn Programming Benchmark (MTPB)

Η μετρική MTPB χρησιμοποιείται για την αξιολόγηση της απόδοσης συστημάτων που παράγουν κώδικα μέσω της διαδικασίας “multi-turn program synthesis” (Bandi, Pydi, & Yudu, 2023). Στην προκειμένη περίπτωση, η ανάπτυξη του επιθυμητού κώδικα γίνεται σταδιακά, μέσω της συνεχούς αλληλεπίδρασης μεταξύ ενός ανθρώπου και του συστήματος σύνθεσης κώδικα. Κατά τη διαδικασία αυτή ο άνθρωπος δίδει ερωτήματα ή οδηγίες σε φυσική γλώσσα (prompt) και το σύστημα παράγει τον κώδικα βασιζόμενο στις κατευθυντήριες οδηγίες. Στη συνέχεια, η ποιότητα

του παραγόμενου κώδικα αξιολογείται με βάση την ορθότητα (correctness), την αποδοτικότητα (efficiency) καθώς και την τήρηση των καθορισμένων απαιτήσεων.

3.7 Μετρικές απόδοσης μοντέλων δημιουργίας γράφων

3.7.1 Validity Metric (Validity constraint)

Η μετρική Validity χρησιμοποιείται για την αξιολόγηση των γράφων που παράγονται από συστήματα AI. Αναλυτικότερα, η μετρική προσδιορίζει κατά πόσο οι παραγόμενοι γράφοι τηρούν τους κανόνες/περιορισμούς που τίθενται από τον τομέα εφαρμογής τους (Bandi, Pydi, & Yudu, 2023). Για παράδειγμα, στο πλαίσιο της δημιουργίας γραφημάτων μορίων, η μετρική διασφαλίζει ότι οι δομές που έχουν δημιουργηθεί ακολουθούν τόσο τους κανόνες χημικής σύνδεσης όσο και τις λοιπές ιδιότητες των μορίων.

Η μετρική Validity έχει χρησιμοποιηθεί για την αξιολόγηση του μοντέλου JT-VAE (Junction Tree Variational Autoencoder)[11]. Το μοντέλο JT-VAE, χρησιμοποιεί την αναπαράσταση δένδρου διασταυρώσεων (Junction Tree, η οποία είναι γνωστή και ως «Clique Tree»), καθώς και έναν μεταβλητό αυτοκωδικοποιήτη (VAE[12]) για να δημιουργήσει γραφήματα μορίων που είναι δομικά έγκυρα και συνεπή ως προς το χημικό πλαίσιο στο οποίο λειτουργούν.

3.7.2 Reconstruction Accuracy

Η μετρική Reconstruction Accuracy αξιολογεί κατά πόσο οι γράφοι οι οποίοι έχουν δημιουργηθεί από ένα μοντέλο AI αναπαριστούν πιστά τα δεδομένα εισόδου ή τον γράφο αναφοράς. Αναλυτικότερα, μέσω της μετρικής, αξιολογείται η ικανότητα του μοντέλου να αποτυπώνει με ακρίβεια τόσο τα βασικά χαρακτηριστικά όσο και τις ιδιαιτερότητες των δεδομένων εισόδου (Bandi, Pydi, & Yudu, 2023).

Το μοντέλο JT-VAE (το οποίο αναφέρθηκε προηγουμένως), επικεντρώνεται όχι μόνο στη δημιουργία έγκυρων (valid) μοριακών γραφημάτων, αλλά και στην ακριβή (accurate) ανακατασκευή των μοριακών δομών εισόδου. Επομένως, το μοντέλο θα επιτύχει υψηλό

[11] https://github.com/kamikaze0923/jtvae

[12] Η VAE είναι μια αρχιτεκτονική νευρωνικών δικτύων που χρησιμοποιείται για ανάλυση και τη δημιουργία νέων δεδομένων. Έχει σχεδιαστεί για να αντιμετωπίζει το προβλήματα αβεβαιότητα κατά την επεξεργασία πληροφοριών

Reconstruction Accuracy εάν καταφέρει να αποτυπώσει τα μοριακά χαρακτηριστικά και τις ιδιότητες τους με υψηλή ακρίβεια.

3.7.3 N.U.V. (Novel, Unique, and Valid Molecules)

Η μετρική N.U.V. αξιολογεί τα μόρια που δημιουργούνται από ένα μοντέλο AI. Η αξιολόγηση γίνεται με βάση τα ακόλουθα χαρακτηριστικά (Bandi, Pydi, & Yudu, 2023):

- Καινοτομία (**Novel**): Κατά πόσο οι παραγόμενες μοριακές δομές διαφέρουν από τα υπάρχοντα μόρια (ενός συνόλου δεδομένων)
- Μοναδικότητα (**Unique**): Αξιολογεί εάν τα παραγόμενα μόρια είναι ιδιαίτερα χαρακτηριστικά ή αν είναι παρόμοια με άλλα παραγόμενα μόρια
- Εγκυρότητα (**Valid**): Αξιολογεί εάν τα παραγόμενα μόρια τηρούν τους κανόνες και περιορισμούς της χημείας.

Η N.U.V. έχει χρησιμοποιηθεί για την αξιολόγηση του μοντέλου δημιουργίας μοριακών γραφών MoFlow.

3.8 Μέτρα απόδοσης μοντέλων δημιουργίας συνθετικών δεδομένων σε πίνακες

3.8.1 DCR (Distance to the Closest Record)

Η μετρική Distance to the Closest Record (distance) χρησιμοποιείται για την αξιολόγηση των συνθετικών πινάκων που δημιουργούνται από ένα μοντέλο AI (Bandi, Pydi, & Yudu, 2023). Οι συνθετικοί πίνακες θα πρέπει να είναι στατιστικά παρόμοιοι με τους πίνακες αναφοράς. Πιο συγκεκριμένα, η DCR μετρά την Ευκλείδεια απόσταση μεταξύ των παραγόμενων συνθετικών εγγραφών και των πλησιέστερων αντίστοιχων εγγραφών από τον πίνακες αναφοράς. Θα πρέπει να σημειωθεί ότι όσο υψηλότερη είναι η τιμή του DCR, τόσο μικρότερη είναι η πιθανότητα τα υπάρξει κίνδυνος διαρροής εμπιστευτικών (private) δεδομένων.

Η μετρική DCR έχει χρησιμοποιηθεί στο μοντέλο table-GAN, προκειμένου να αξιολογήσει την εγγύτητα και ομοιότητα των συνθετικών εγγραφών ως προς τις εγγραφές αναφοράς.

3.8.2 Macro-F1

Η μετρική Macro-F1 αξιολογεί την απόδοση (performance) και την ομοιότητα (similarity) των παραγόμενων συνθετικών δεδομένα ως προς τα δεδομένα αναφοράς (Bandi, Pydi, & Yudu, 2023). Για ένα σύνολο δεδομένων, το Macro-F1 υπολογίζεται ως ο μέσος όρος του F1-score κάθε κλάσης. Θα πρέπει να τονιστεί ότι, η μετρική Macro-F1 χρησιμοποιείται αντί της μετρικής της ορθότητας (Accuracy), διότι σε ένα σύνολο δεδομένων, ο αριθμός των δειγμάτων δεν είναι ισομερώς κατανεμημένος σε όλες τις κλάσεις. Παράδειγμα χρήσης της μετρικής, αποτελεί το μοντέλο Tabular GAN.

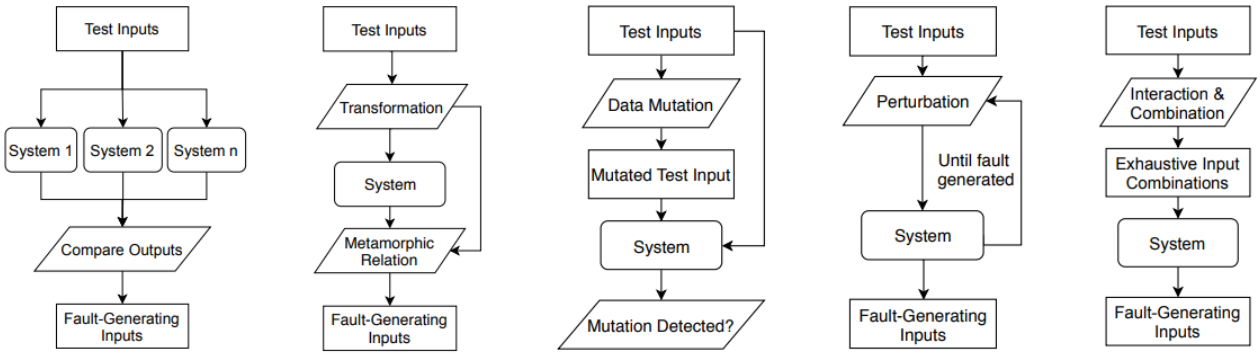
3.8.3 Mean Relative Error (MRE)

Η μετρική Mean relative error (MRE) χρησιμοποιείται για την αξιολόγηση των συνθετικών πινάκων που δημιουργούνται από ένα μοντέλο AI. Η MRE ποσοτικοποιεί το μέσο σχετικό σφάλμα (average relative error) μεταξύ των παραγόμενων συνθετικών δεδομένων και των δεδομένων αναφοράς, λαμβάνοντας ταυτοχρόνως υπόψη το μέγεθος των σφαλμάτων (Bandi, Pydi, & Yudu, 2023).

Η μετρική MRE έχει χρησιμοποιηθεί στο μοντέλο table-GAN, προκειμένου να αξιολογήσει την ακρίβεια (accuracy) και την ομοιότητα των παραχθέντων συνθετικών δεδομένων ως προς τα δεδομένα αναφοράς.

4 Μεθοδολογίες αξιολόγησης AI συστημάτων (Testing methods)

Στο παρόν κεφάλαιο παρουσιάζονται οι μέθοδοι που έχουν αναπτυχθεί προκειμένου να γίνει δομική αξιολόγηση των συστημάτων ML. Στην Εικόνα 13 αναπαρίστανται σχηματικά μερικές από τις ευρέως χρησιμοποιούμενες τεχνικές αξιολόγησης των συστημάτων AI. Οι τεχνικές αυτές αναλύονται στα επόμενα κεφάλαια. Ακόμα, στην Εικόνα 14 αντιστοιχίζονται οι μεθοδολογίες με το είδος του προβλήματος το οποίο μπορούν να αντιμετωπίσουν.



Εικόνα 13: Σχηματική αναπαράσταση των μεθοδολογιών αξιολόγησης των AI συστημάτων (Ahuja, Gotlieb, & Spieker, 2022)

	DT	MT	MuT	APT	CT
Quality of the Model		✓	✓	✓	✓
Quality of Training Data			✓	✓	
Oracle Problem	✓	✓			
Test Input Selection		✓	✓		✓
Adversarial Detection				✓	

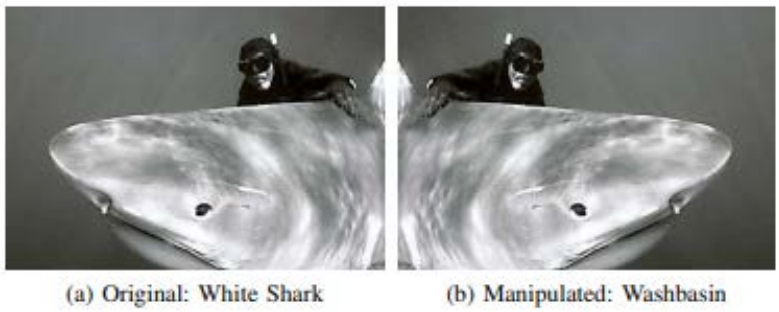
Εικόνα 14: Προσδιορισμός προβλημάτων που αντιμετωπίζονται ανά Μεθοδολογίας Αξιολόγησης (Ahuja, Gotlieb, & Spieker, 2022)

4.1 Metamorphic testing (MT)

Η μεθοδολογία «Metamorphic testing» χρησιμοποιείται για την αξιολόγηση συστημάτων ML, κυρίως, όταν είναι δύσκολο ή αδύνατον να προσδιοριστεί η αναμενόμενη έξοδος για ένα συγκεκριμένων σύνολο εισόδων (Ahuja, Gotlieb, & Spieker, 2022).

Για την εύρεση των πιθανών προβλημάτων (defects/bugs) του συστήματος, η μεθοδολογία ML, βασίζεται στον εντοπισμό των μεταμορφικών σχέσεων (MR – metamorphic relations). Ως MR

καλείται κάθε γνωστή σχέση μεταξύ των δεδομένων εισόδου και εξόδου ενός προγράμματος, και μέσω του συνόλου των MR μπορούν να εντοπιστούν σφάλματα στο πρόγραμμα.



Εικόνα 15: Η οριζόντια αναστροφή (horizontal flipping) της αρχικής εικόνας είχε το ως αποτέλεσμα μοντέλο Zoo (το οποίο διανέμεται μέσω του PyTorch framework) να την ταξινομήσει λανθασμένα στην κλάση «Washbasin», αντί στην «White Shark»

Για παράδειγμα, έστω ότι ο σκοπός ενός συστήματος ML είναι να ταξινομήσει τις εικόνες σε 2 ή περισσότερες κλάσεις (classification). Μια τυπική μεταμορφική σχέση μπορεί να είναι η ακόλουθη - εάν δημιουργήσουμε μια νέα εικόνα (b) αναστρέφοντάς την αρχική εικόνα (a) οριζόντια, τότε το ML σύστημα θα πρέπει να ταξινομήσει την ανεστραμμένη εικόνα στην ίδια κλάση με την αρχική εικόνα παρόλο που η ετικέτα ταξινόμησης της είναι άγνωστη.

Ας θεωρήσουμε επιπλέον το ακόλουθο παράδειγμα: έστω ότι ο σκοπός του συστήματος ML είναι η ανίχνευση των αντικειμένων (objects) που υπάρχουν σε μια εικόνα (το μοντέλο ML θα πρέπει να υπολογίσει το περίγραμμα του κάθε αντικειμένου), και στη συνέχεια η ταξινόμηση τους. Σε αυτό το σύστημα, είναι δυνατός ο ορισμός πολλών μεταμορφικών σχέσεων (MR), μερικοί εκ των οποίων είναι οι ακόλουθοι:

1. **MR 1: Μετασχηματισμός μεγέθους εικόνας.**
    - Συρρίκνωση της εικόνας κατά X%. (π.χ. 50%)
    - Μεγέθυνση της εικόνας κατά Y%. (π.χ. 200%)
2. **MR 2: Μετασχηματισμός ανάλυσης εικόνας.**
    - Αλλαγή της ανάλυση της εικόνας (π.χ., από 1080p σε 720p)
3. **MR 3: Μετασχηματισμός περιστροφής.**
    - Περιστροφή της εικόνας κατά 90 μοίρες.
    - Περιστροφή της εικόνας κατά 180 μοίρες.

- Περιστροφή της εικόνα κατά X μοίρες (τυχαία γωνία).

4. **MR 4: Μετασχηματισμός φωτεινότητας.**
   - Αύξηση της φωτεινότητας κατά X% (π.χ. 20%).
   - Μείωσης της φωτεινότητας κατά Y% (π.χ. 50%).
   - Προσθήκη θορύβου στην εικόνα.
5. **MR 5: Μετασχηματισμός μετατόπισης.**
   - Μετακίνηση όλων των αντικειμένων X pixel (π.χ. 20 pixel) προς τα δεξιά.
   - Μετακίνηση όλων των αντικειμένων Y pixel (π.χ 10 pixel) προς τα κάτω.
   - Μετακίνηση των αντικειμένων σε μια νέα τυχαία θέση.
6. **MR 6: Μετασχηματισμός κλίμακας χρώματος.**
   - Αλλαγή της ισορροπία λευκού της εικόνας.
   - Μετατροπή της εικόνα σε κλίμακα του γκρι.
   - Εφαρμογή φίλτρου χρώματος στην εικόνα (π.χ., sepia).
7. **MR 7: Μετασχηματισμός προσθήκης αντικειμένων.**
   - Προσθήκη ενός νέου αντικειμένου γνωστής κατηγορίας σε μια άκρη της εικόνας.
   - Προσθήκη ενός νέου αντικειμένου άγνωστης κατηγορίας σε μια κεντρική θέση της εικόνας.
   - Προσθήκη πολλαπλών αντικειμένων σε διάφορες θέσεις στην εικόνα.

Όταν πραγματοποιηθούν οι προαναφερθείσες αλλαγές στις εικόνες, το σύστημα θα πρέπει να ανιχνεύει τον ίδιο αριθμό αντικειμένων που έχει η αρχική εικόνα, καθώς επίσης να ταξινομεί σωστά στις αντίστοιχες κλάσεις τα αντικείμενα.

Όπως γίνεται αντιληπτό, ξεκινώντας από μια αρχική περίπτωση δοκιμής (Test case) είναι δυνατή η δημιουργία επιπρόσθετων δοκιμαστικών περιπτώσεων, των οποίων το τελικό αποτέλεσμα θα πρέπει να ικανοποιεί τις ιδιότητες που ορίζονται από τη μεταμορφική σχέση. Όταν η μεταμορφική σχέση δεν ικανοποιείται, τότε η δοκιμαστική περίπτωση αποτυγχάνει, υποδεικνύοντας ότι υπάρχει πρόβλημα στο σύστημα. Παρόλα αυτά, το σημείο του συστήματος που υπάρχει το σφάλμα καθώς και η αναμενόμενη έξοδος του προγράμματος, δεν προσδιορίζονται.

Η μεθοδολογία Metamorphic testing έχει εφαρμοστεί για τη δομική αξιολόγηση βιβλιοθηκών ML, μοντέλων και συστημάτων ML, καθώς επίσης για την αξιολόγηση αλγορίθμων ML, όπως είναι ο k-nearest neighbor classifiers, ο Support Vector Machines (SVM), ο MartiRank, και ο Naïve

Bayes. Ακόμα, η μεθοδολογία MT έχει εφαρμοστεί για τη δομική αξιολόγηση των DNN, τα οποία χρησιμοποιούνται στα αυτόνομα οχήματα (self-driving cars), και στους ταξινομητές εικόνων (Chandrasekaran, Cody, Mccarthy, Lanus, & Freeman, 2023) .

Συνοψίζοντας, για την εφαρμογή της μεθοδολογίας MT, αρχικά θα πρέπει να προσδιοριστούν οι μεταμορφικές σχέσεις, βάσει των οποίων δημιουργούνται οι δοκιμαστικές περιπτώσεις. Αφού δημιουργηθούν οι δοκιμαστικές περιπτώσεις, στη συνέχεια αξιολογούνται και εκτελούνται. Σε αντίθεση με τις τεχνικές δοκιμών που εφαρμόζονται στα συμβατικά συστήματα, όπου τα δεδομένα εισόδου συγκρίνονται με την αναμενόμενη τιμή εξόδου, η μεθοδολογία MT αξιολογεί εάν δοκιμαστικές περιπτώσεις ικανοποιούν ή όχι τις μεταμορφικές σχέσεις. Αυτή η προσέγγιση εξαλείφει την ανάγκη δημιουργίας ενός Test Oracle, που θα ήταν ούτως ή άλλως εξαιρετικά δυσχερής έως αδύνατη, λόγω της στοχαστικής φύσης των προβλέψεων των μοντέλων DL.

Στις συμβατικές δοκιμές αξιολόγησης, η απουσία περιπτώσεων δοκιμών που αποτυγχάνουν στην σουίτα, δεν εξασφαλίζει ότι στο υπό δοκιμή σύστημα δεν υπάρχουν προβλήματα (bugs). Το ίδιο ισχύει και για τη μεθοδολογία MT: ακόμα και εάν η σουίτα δοκιμών ικανοποιεί όλες τις μεταμορφικές σχέσεις, δεν είναι βέβαιο, ότι δεν υπάρχουν σφάλματα στο σύστημα ML καθώς είναι δυνατόν το μοντέλο να έπρεπε να ικανοποιεί κάποιες πρόσθετες μεταμορφικές σχέσεις οι οποίες ωστόσο δεν έχουν προστεθεί στη σουίτα και κατά συνέπεια δεν είναι δυνατόν να διαπιστωθεί ότι το μοντέλο αποτυγχάνει στο να τις ικανοποιήσει.

Θα πρέπει να τονιστεί ότι, η αποτελεσματικότητα των μεταμορφικών δοκιμών καθορίζεται από την ποιότητα των μεταμορφικών σχέσεων. Για να προσδιοριστούν MRs υψηλής ακρίβειας που είναι σχετικές και κατάλληλες για το υπό δοκιμή σύστημα, είναι αναγκαίο ο αξιολογητής (Tester) να έχει πολύ καλή γνώση του τομέα εφαρμογής (business domain) του συστήματος. Αξίζει να σημειωθεί ότι οι έρευνες δείχνουν ότι κάνοντας χρήση 3-6 διαφορετικών MRs, μπορούν να εντοπιστούν πάνω από το 90% των ελαττωμάτων που θα μπορούσαν να βρεθούν εάν χρησιμοποιούνταν τεχνικές που βασίζονται στη δημιουργία ενός κλασσικού Test Oracle.

4.2 Differential testing (DT)

Η μεθοδολογία «Differential testing» ή Back-to-Back Testing χρησιμοποιείται για την εύρεση πιθανών προβλημάτων (bugs) στα συστήματα ML. Για τη χρήση αυτής της μεθοδολογίας, θα πρέπει είτε να υπάρχει τουλάχιστον ένα ακόμα μοντέλο/εφαρμογή που να εκτελεί την ίδια

λειτουργία (function) με την υπό αξιολόγηση εφαρμογή, είτε το μοντέλο που υπόκειται προς αξιολόγηση να έχει μία διαφορετική έκδοση (version).

Πιο συγκεκριμένα, για ένα ορισμένο σύνολο δεδομένων εισόδου, οι δοκιμαστικές περιπτώσεις (Test cases) εκτελούνται σε δυο ή περισσότερα διαφορετικά μοντέλα που χρησιμοποιούνται για την ίδια λειτουργία. Η μεθοδολογία DT συγκρίνει τα αποτελέσματα της σουίτας δοκιμών προκειμένου να αξιολογήσει την ορθότητα του προγράμματος (Chandrasekaran, Cody, Mccarthy, Lanus, & Freeman, 2023). Εάν εντοπιστούν διαφορετικές έξοδοι σε μια από τις υλοποιήσεις, τότε το πρόγραμμα θεωρείται ότι υπάρχει ροπή σε σφάλματα και χρειάζεται περαιτέρω διερεύνηση για τον προσδιορισμό της αιτίας του προβλήματος.

Όπως γίνεται αντιληπτό, η DF εξαλείφει την ανάγκη δημιουργίας ενός Test Oracle, διότι συγκρίνει τις εξόδους παρόμοιων υλοποιήσεων και τις μη ομοιόμορφες εξόδους τις θεωρεί ενδείξεις μη σωστής συμπεριφοράς. Από την άλλη πλευρά, οι κύριοι περιορισμοί της DT είναι οι ακόλουθοι:

1. δεν καθίσταται δυνατή η εύρεση του ελλαττωματικού συστήματος, καθώς δεν είναι εφικτό να εντοπιστεί πιο σύστημα δίδει σωστές απαντήσεις
2. Τα σφάλματα ανιχνεύονται μόνο εάν μία από τις υλοποιήσεις παράξει μια διαφορετική έξοδο σε σύγκριση με τις άλλες.

Το DeepXplore είναι το πρώτο framework που χρησιμοποίησε τη μεθοδολογία DT για την αξιολόγηση των μοντέλων ML. Χρησιμοποιώντας 5 σύνολα δεδομένων (MNIST, ImageNet, VirusTotal, Udacity video, και Drebin) τα οποία είναι διαθέσιμα στο ευρύ κοινό ως είσοδο, το DeepXplore αξιολόγησε μια σειρά από μοντέλα ML, δείχνοντας ότι συνδυάζοντας την κάλυψη ενεργοποίησης των νευρώνων (neural activation coverage) με το DT είναι δυνατόν να παραχθούν δεδομένα εισόδου που δημιουργούν λανθασμένες συμπεριφορά (Pei, Cao, Yang, & Jana, 2017). Στη συνέχεια, χρησιμοποιώντας τα δεδομένα που προκαλούν πρόβλημα στα περισσότερα μοντέλα, είναι δυνατή η επαν-εκπαίδευση του εκάστοτε μοντέλου, προκειμένου να αυξηθεί η ορθότητα (accuracy) του.

Επίσης, η μεθοδολογία DT έχει εφαρμοστεί για την αξιολόγηση μοντέλων ML, τα οποία χρησιμοποιούνται στο σύστημα διεύθυνσης των αυτόνομων οχημάτων, καθώς και σε συστήματα αναγνώρισης φωνής (automatic speech recognition systems). Επιπλέον, το CRADLE (Pham, Lutellier, Qi, & Tan, 2019) έχει αξιοποιήσει τη μεθοδολογία DT προκειμένου να αξιολογήσει τα

ακόλουθα ML Frameworks: TensorFlow, CNTK, και Theano. Για τον σκοπό αυτό χρησιμοποιήθηκαν 11 σύνολα δεδομένων και 30 μοντέλα ML. Μέσω αυτής της μεθοδολογίας, οι ερευνητές κατάφεραν να εντοπίσουν περισσότερα από 100 προβλήματα στα 3 frameworks. Επιπλέον, το Differential testing έχει εφαρμοστεί για την αξιολόγηση non-DL αλγορίθμων ταξινόμησης οι οποίοι αναπτύχθηκαν με την χρήση των ακόλουθων βιβλιοθηκών Caret, SparkMLLib, scikit-learn και WEKA.

Τέλος, η βιβλιογραφία έχει δείξει ότι το DT έχει χρησιμοποιηθεί επιτυχώς για την αξιολόγηση της ορθότητας (correctness) των ML μοντέλων, frameworks και των βιβλιοθηκών που χρησιμοποιούνται για την εκπαίδευση τους.

4.3 Fuzzing testing (FT)

Η μεθοδολογία «Fuzz testing» χρησιμοποιείται τόσο για τον εντοπισμό των πιθανών σφαλμάτων/ευπαθειών τους συστήματος (π.χ.: καταρρεύσεων του προγράμματος (program crashes), και των καταστροφών δεδομένων στη μνήμη (memory corruption)), καθώς και για την αξιολόγηση της ανθεκτικότητας (robustness) και της ασφάλειας (safety) του.

Κατά τη διάρκεια του Fuzz Testing, διοχετεύονται στο σύστημα τυχαία/ημι-τυχαία ή μη-αναμενόμενα δεδομένα, ενώ ταυτόχρονα, παρακολουθείται ο τρόπος λειτουργίας του συστήματος προκειμένου να ανιχνευθούν πιθανές ευπάθειές της εφαρμογής (Chandrasekaran, Cody, Mccarthy, Lanus, & Freeman, 2023). Σημειώνεται δε, ότι αυτά του είδους τα σφάλματα δεν θα ήταν δυνατόν να εντοπιστούν από τις συμβατικές τεχνικές δοκιμών.

Επιπλέων, για την αξιολόγηση των συστημάτων DL, έχουν προταθεί δυο framework που βοηθούν στη δημιουργία των δοκιμαστικών περιπτώσεων, και βασίζονται στη μεθοδολογία «Fuzz testing». Αναλυτικότερα:

- Το framework δημιουργίας δοκιμαστικών περιπτώσεων που πρότειναν οι (Guo, Jiang, Zhao, Chen, & Sun, 2018), αξιοποιεί την ανατροφοδότηση κάλυψης (coverage feedback) των βαθιών νευρωνικών δικτύων προκειμένου να δημιουργήσει μη αναμενόμενες τιμές εισόδου.

- Αντίστοιχα, το framework δημιουργίας δοκιμαστικών περιπτώσεων που πρότειναν οι (XIE, et al., 2019), αξιοποιεί την τεχνική coverage-guided fuzzing (ή οποία είναι γνωστή και ως greybox fuzzing) για τη δημιουργία των δεδομένων εισόδου.

Επιπρόσθετα, βιβλιοθήκες DL έχουν αξιολογηθεί εφαρμόζοντας τη μεθοδολογία «Fuzz testing».

Συνοψίζοντας, η μεθοδολογία FT μπορεί να χρησιμοποιηθεί για να αξιολογήσει ένα σύστημα ML ως προς την συμπεριφορά του στις ακραίες τιμές (outlier scenarios), την ανθεκτικότητα του (robustness) και την ασφάλεια του (security).

4.4 Concolic testing (Dynamic Symbolic Execution -DSE)

Η μεθοδολογία «Concolic testing» η οποία είναι γνωστή και ως Dynamic Symbolic Execution (DSE) και συνδυάζει τη συμβολική εκτέλεση (symbolic execution) με την κανονική εκτέλεση (concrete execution) για να εντοπίσει πιθανά προβλήματα (bugs) στο σύστημα καθώς και για να προσδιορίσει την κάλυψη του των τεστ αναφορικά με τον κώδικα (code coverage).

Η βασική ιδέα της μεθοδολογίας DSE είναι να αξιοποιηθεί το συντακτικό δέντρο του υπό αξιολόγηση κώδικα προκειμένου να εντοπιστούν είσοδοι που παράγουν σφάλματα κατά τη διάρκεια εκτέλεσης του, και παράλληλα να επιτευχθεί υψηλό επίπεδο κάλυψης του κώδικα.

Η συμβολική εκτέλεση επιλύει τους περιορισμούς (constrains) που προκύπτουν από τις εντολές διακλάδωσης (branch instructions) προκειμένου να δημιουργήσουν μια νέα τιμή εισόδου, ενώ η κανονική εκτέλεση χρησιμοποιείται για να καθοδηγήσει τη συμβολική εκτέλεση στο μονοπάτι ενδιαφέροντος.

Για την εκτέλεση της μεθοδολογίας DSE απαιτούνται δύο περιβάλλοντα, το συμβολικό και το κανονικό. Στο συμβολικό περιβάλλον γίνεται αντιστοίχιση των μεταβλητών στις συμβολικές εκφράσεις τους, ενώ στο κανονικό περιβάλλον γίνεται αντιστοίχιση των μεταβλητών στις πραγματικές (concrete) τιμές τους.

Η μεθοδολογία DSE αρχικά εκτελεί το πρόγραμμα χρησιμοποιώντας ως είσοδο μια πραγματική τιμή (concrete value). Η πραγματική τιμή μπορεί να δοθεί είτε από τον χρήστη είτε να δημιουργηθεί τυχαία (ή καθοδηγούμενα). Αφού ολοκληρωθεί η κανονική εκτέλεση, επιλέγεται ευρεστικά μια διαφορετική διαδρομή (path) εκτέλεσης. Στη συνέχεια, η νέα διαδρομή

κωδικοποιείται συμβολικά για να μπορέσει το πρόγραμμα να παράξει μια νέα πραγματική τιμή, η οποία διατηρεί όλους τους περιορισμούς της διαδρομής εκτέλεσης. Η συμβολική και κανονική εκτέλεση εναλλάσσονται έως ότου επιτευχθεί το επιθυμητό επίπεδο κάλυψης του κώδικα (Zhang, Harman, Ma, & Yang, 2020). Η DSE, αποτελεί κύρια μεθοδολογία πολλών εργαλείων εντοπισμού σφαλμάτων καθώς έχει διαπιστωθεί ότι είναι ακριβής (accurate) και αποτελεσματική (effective) μεθοδολογία.

Όπως έχει ήδη αναφερθεί, η απόδοση ενός ML καθορίζεται τόσο από τον κώδικά όσο και από τα δεδομένα που χρησιμοποιούνται για την εκπαίδευση του μοντέλου, ως εκ τούτου η μεθοδολογία αυτή έχει εφαρμογή τόσο στα δεδομένα όσο και στον κώδικα.

Η εφαρμογή της DSE στην μηχανική μάθηση είναι μια απαιτητική διαδικασία καθώς:

- Οι διακλαδώσεις των νευρωνικών δικτύων δεν είναι σαφώς ορισμένες
- Τα νευρωνικά δίκτυα μπορεί να παρουσιάζουν υψηλή μη-γραμμικότητα, με αποτέλεσμα να μην μπορούν να προσδιοριστούν εύκολα οι περιορισμοί που προκύπτουν από αυτά τα δίκτυα, και επομένως, η επόμενη διαδρομή εκτέλεσης
- Δημιουργούνται προβλήματα επεκτασιμότητας (scalability) διότι τα ML μοντέλα είναι πολύ πολύπλοκα και προς το παρών δεν υπάρχουν εργαλεία που μπορούν να πραγματοποιήσουν τόσο απαιτητική συμβολική ανάλυση

Για να αντιμετωπιστούν αυτές οι δυσκολίες, δημιουργήθηκε το DeepCheck, το οποίο μετατρέπει ένα DNN σε πρόγραμμα, ώστε να γίνει δυνατή η εφαρμογή της DSE μεθοδολογίας. Μέσω της DSE γίνεται δυνατός ο εντοπισμός των εικονοστοιχείων (pixels) που μπορούν να χρησιμοποιηθούν για να πραγματοποιηθούν επιθέσεις στο σύστημα και έχουν το ίδιο μοτίβο ενεργοποίησης με την αρχική εικόνα. Ειδικότερα, το DeepCheck, δημιουργεί επιθέσεις ενός ή και δυο εικονοστοιχείων εντοπίζοντας τα εικονοστοιχεία ή τα ζεύγη των εικονοστοιχείων των τροποποιημένων εικόνων που δεν ταξινομούνται σωστά από το νευρωνικό δίκτυο.

Ένα ακόμα εργαλείο που αντιμετωπίζει τις δυσκολίες της απ'ευθείας εφαρμογής της DSE στα ML μοντέλα και συστήματα ονομάζεται LIME (Local Interpretable Model-agnostic Explanations). Η LIME αποτελεί ένα τοπικό εργαλείο εξήγησης. Αυτό σημαίνει ότι για κάθε παράδειγμα δεδομένων που εξετάζεται, η LIME προσπαθεί να κατανοήσει τον τρόπο με τον οποίο το μοντέλο μηχανικής μάθησης καταλήγει στην απόφασή του, δημιουργώντας ένα τοπικό ερμηνεύσιμο μοντέλο (π.χ.:

γραμμικό μοντέλο, δέντρο αποφάσεων, ή μια λίστα κανόνων που αποτυγχάνουν) γύρω από το συγκεκριμένο παράδειγμα δεδομένων προκειμένου να εντοπίσει τη διαδρομή που μπορεί να χρησιμοποιηθεί στην συμβολική εκτέλεση. Επιπλέων, έχει αποδειχθεί ότι η LIME δημιουργεί 3.72 φορές πιο αποτελεσματικές περιπτώσεις δοκιμών ως προς την TMEMIS (η οποία δημιουργεί τυχαίες περιπτώσεις δοκιμών).

Επιπρόσθετα, για την αξιολόγηση των DNN έχει δημιουργηθεί και το εργαλείο DeepConcolic. Το εργαλείο αυτό, χρησιμοποιεί την κανονική εκτέλεση για να κατευθύνει τη συμβολική ανάλυση στα διάφορα MC/DC κριτήρια αναλύοντας τις ιδιότητες του ML μοντέλου. Θα πρέπει να σημειωθεί ότι το DeepConcolic καταφέρνει 10% υψηλότερη κάλυψη νευρώνων ως προς το DeepXplore.

4.5 Adversarial Perturbation Testing (APT)

Όλα τα ML συστήματα, συμπεριλαμβανομένων των state-of-the-art μοντέλων, είναι επιρρεπή σε adversarial attacks. Adversarial attack πραγματοποιείται όταν ένας κακόβουλος δράστης δίδει στο μοντέλο μια είσοδο η οποία είναι πανομοιότυπη με την αναμενόμενη, προκειμένου να αναγκάσει το ML μοντέλο να κάνει λανθασμένες προβλέψεις και να επιδείξει ανεπιθύμητη συμπεριφορά. Οι ανεπιθύμητες είσοδοι είναι δύσκολο να εντοπιστούν με τα κλασικά μέσα ανίχνευσης ή και με το ανθρώπινα μάτια διότι δεν διαφέρουν σημαντικά από τις αναμενόμενες τιμές εισόδου (Zhang, Harman, Ma, & Yang, 2020).

Θα πρέπει να σημειωθεί ότι, τα adversarial δεδομένα είναι μεταβιβάσιμα (transferable). Αυτό σημαίνει ότι εάν τα adversarial δεδομένα εισόδου προκαλούν ανεπιθύμητη συμπεριφορά σε ένα ML σύστημα A που έχει εκπαιδευτεί να εκτελεί την X εργασία, τότε τα ίδια adversarial δεδομένα θα κάνουν και το ML σύστημα B που εκτελεί την ίδια (X) εργασία να αποτύχει - ακόμη και εάν τα ML συστήματα βασίζονται σε διαφορετικές αρχιτεκτονικές και έχουν εκπαιδευτεί διαφορετικά δεδομένα.

Adversarial attacks μπορούν να προκληθούν εφαρμόζοντας είτε white-box είτε black-box μεθοδολογίες. Για την εφαρμογή της white-box μεθοδολογίας, ο Αξιολογιτής (Tester), θα πρέπει να ξέρει ποιος αλγόριθμος χρησιμοποιήθηκε για την εκπαίδευση του μοντέλου, καθώς και τις ρυθμίσεις του μοντέλου (model settings) και των παραμέτρων του (parameters). Έχοντας αυτές τις πληροφορίες, ο Αξιολογήτης ή ο κακόβουλος χρήστης, μπορεί να δημιουργήσει adversarial

δεδομένα εισόδου (Zhang, Harman, Ma, & Yang, 2020). Για παράδειγμα, μπορεί να εισάγει μικρές διαταραχές στα δεδομένα που διοχετεύονται στο σύστημα και να παρακολουθεί ταυτόχρονα ποιες από αυτά προξενούν μεγάλες διαταραχές στο μοντέλο.

Για την εφαρμογή της black-box μεθοδολογίας, ο κακόβουλος χρήστης θα πρέπει να κατανοήσει την τρόπο λειτουργείας του ML συστήματος που επιθυμεί να πραγματοποιήσει την επίθεση, και στην συνέχεια θα πρέπει να κατασκευάσει ένα παρόμοιο ML μοντέλο (αντίγραφο) που να έχει παραπλήσια λειτουργικότητα. Έπειτα, θα πρέπει να εφαρμόσει τις white-box τεχνική στο μοντέλο που κατασκεύασε προκειμένου να εντοπίσει adversarial δεδομένα εισόδου (Dussa-Zieger, et al., 2021). Όπως αναφέρθηκε προηγμένος, δεδομένου ότι τα adversarial δεδομένα είναι μεταβιβάσιμα, ο ίδια adversarial δεδομένα μπορούν να χρησιμοποιηθούν και στο αρχικό μοντέλο.

Συνοψίζοντας, είναι σημαντικό τα ML συστήματα και συγκεκριμένα τα ML components να αξιολογούνται ως προς την ανθεκτικότητα (robustness) τους σε adversarial attacks. Για το σκοπό αυτό, ο Αξιολογιτής, θα πρέπει να διοχετεύσει στο σύστημα adversarial δεδομένα εισόδου. Τα adversarial δεδομένα μπορούν να δημιουργηθούν εισάγοντας μη ανιχνεύσιμες διαταραχές (perturbations) στις αναμενόμενες εισόδους (natural inputs). Για τη δημιουργία τους έχουν χρησιμοποιούνται διάφορες τεχνικές ή συνδυασμός τεχνικών, όπως είναι:

- Fuzz testing
- Differential testing
- Το Generative Adversarial Network (GAN) σε συνδιασμό με την Metamorphic testing μεθοδολογία

Επίσης για τη δημιουργία adversarial δεδομένων εισόδου καθώς και για την αξιολόγηση της ανθεκτικότητας των ML συστημάτων έχουν δημιουργηθεί πολλές βιβλιοθήκες όπως: AdverTorch, FoolBox SecML, Cleverhans, ART, και το DeepRobust.

4.6 Combinatorial testing (CT)

Ιδανικά, η δομική αξιολόγηση ενός πληροφοριακού συστήματος θα πρέπει να λαμβάνει υπόψιν όλες τις πιθανές καταστάσεις που μπορεί να βρεθεί – αυτό είναι γνωστό ως εξαντλητική δοκιμή (exhaustive testing). Παρόλα ταύτα, στις περισσότερες περιπτώσεις, ιδίως όταν το πρόγραμμα είναι πολύπλοκο, η εξαντλητική δοκιμή είναι ανέφικτή, κυρίως λόγω των χρονικών περιορισμών

και του υψηλού κόστους. Απώτερος σκοπός της «Combinatorial testing» μεθοδολογίας είναι να καλύψει όλες τις πιθανές συνθήκες που μπορεί να βρεθεί το υπό αξιολόγηση σύστημα για τις διάφορες τιμές εισόδου, επιτρέποντας με αυτόν τον τρόπο στους Αξιολογητές να εκτελέσουν ψευδο-εξαντλιτική δοκιμή.

Η "Combinatorial testing" μεθοδολογία βασίζονται στην ακόλουθη παραδοχή - οι παράμετροι εισόδου (οι οποίες αποτελούνται κατά βάση από 2 ή 3 τιμές) συνήθως αλληλοεπηρεάζονται, και επομένως ένα ζεύγος εισόδων μπορεί να αναδείξει πιθανά προβλήματα του συστήματος. Άρα, αντί να εξετάζονται όλοι οι πιθανοί συνδυασμοί των παραμέτρων, η CT εστιάζει στις αλληλεπιδράσεις των αντιπροσωπευτικών τους τιμών, για να ασκήσει πιο αποτελεσματικά και δομημένα τη δοκιμή αξιολόγησης (Zhang, Harman, Ma, & Yang, 2020). Πρόσφατες μελέτες έχουν δείξει ότι η χρήση 6 ή λιγότερων χαρακτηριστικών (features) αρκούν για να αναδείξουν το μεγαλύτερο μέρος σφαλμάτων ενός συστήματος.

Αυτό το συγκριτικό της πλεονέκτημα, οδήγησε την εφαρμογή της "Combinatorial testing" μεθοδολογίας στην αξιόλογη DL μοντέλων. Στην προκειμένη περίπτωση, η CT χρησιμοποιείται ως black-box μεθοδολογία, και εισχωρεί στο νευρωνικό δίκτυο προκειμένου να προσδιορίσει τις αλληλεπιδράσεις μεταξύ των «αρχιτεκτονικών νευρώνων» (architectural neurons). Υπάρχουν δύο είδη CT:

- Σταθερής ισχύος (fixed strength)
- Μεταβλητής ισχύος (variable strength)

Παράδειγμα χρήσης της CT σταθερής ισχύος σε DL μοντέλο: Η ερευνητές χρησιμοποίησαν αυτή την τεχνική (που στην προκειμένη περίπτωση την αποκαλούν ως DeepCT) για να αξιολογήσουν την ανθεκτικότητα (robusteness) του συστήματος μειώνοντας τον αριθμό των πιθανών καταστάσεων που μπορεί να βρεθεί κατά τη διάρκεια της εκτέλεσης. Αναλυτικότερα, η DeepCT εξετάζει πώς αλληλοεπιδρούν μεταξύ τους οι νευρώνες στα διάφορα επίπεδα, για να εντοπίσει πιθανά προβλήματα στην ροή των πληροφοριών του δίκτυο. Οι ερευνητές προτείνουν την χρήση της CT, όταν το DL που υπόκειται σε δοκιμές αξιολόγησης παράγει μεγάλο αριθμό (διαφορετικών) εξόδων, και άρα έχει πολλές καταστάσεις εκτέλεσης (large runtime states). Οι ερευνητές, πρότειναν επίσης τη «CT Coverage Guided Test Generation» για τη δημιουργία

δοκιμαστικών περιπτώσεων οι οποίες μπορούν εντοπίσουν σφάλματα στο νευρωνικό δίκτυο πιο γρήγορα και πιο αποτελεσματικά κατά τα πρώιμα στάδια ανάπτυξης του μοντέλου.

Παράδειγμα χρήσης της CT μεταβλητής ισχύος σε DL μοντέλο: Η CT μεταβλητής ισχύος, αξιολογεί πώς η συνδυαστική ενεργοποίηση πολλαπλών νευρώνων στο pre-layer (Το επίπεδο που τροφοδοτεί πληροφορίες στο επόμενο επίπεδο) επηρεάζει την ενεργοποίηση ενός νευρώνα του post-layer (Το επίπεδο που δέχεται τις πληροφορίες από το προηγούμενο επίπεδο). Κατανοώντας αυτές τις αλληλεπιδράσεις, ο Αξιολογητής μπορεί να αποκτήσει πολύτιμες γνώσεις για το πώς λειτουργούν τα DNNs, δίνοντάς του τη δυνατότητα βελτιώσει τις δοκιμές αξιολόγησης του εκάστοτε νευρωνικού δικτύου.

Η μεθοδολογία CT έχει χρησιμοποιηθεί για να αντιμετωπίσει αρκετά προβλήματα που δημιουργούνται κατά την αξιολόγηση των ML συστημάτων. Η CT έχει χρησιμοποιηθεί για τον σχεδιασμό δοκιμαστικών περιπτώσεων, για τη δημιουργία συνθετικών δεδομένων, για την σύγκριση και αξιολόγηση της ποιότητας συνόλων δεδομένων, για την τμηματοποίηση των συνόλων (slicing datasets), για την αξιολόγηση ML αλγορίθμων και DNN μοντέλων, για να εξηγήσει πως ένα ML μοντέλο λαμβάνει αποφάσεις, καθώς και για αξιολογήσει εάν το ML μοντέλο κάνει δίκαιες (fair) προβλέψεις.

Συνοψίζοντας, η δυνατότητα της CT να αξιολογεί πολύπλοκά υπολογιστικά συστήματα, χρησιμοποιώντας σχετικά μικρό αριθμό περιπτώσεων δοκιμών, την κάνει ιδανική για αντιμετώπιση των προκλήσεων που δημιουργούνται στα ML συστήματα, λόγο του μεγάλο αριθμό εισόδου (large input space) που μπορούν να δεχθούν. Παρόλα ταύτα, η αποτελεσματικότητα της CT μεθδολογίας βασίζεται στην σωστή στη μοντελοποίηση των παραμέτρων καθώς και των σχετικών τιμών τους κατά τη διάρκεια δημιουργίας των δοκιμαστικών περιπτώσεων.

5 Επάρκεια Δοκιμών

Η Επάρκεια Δοκιμών (Test Adequacy) μετρά πόσο καλά ένα σύνολο δοκιμαστικών περιπτώσεων καλύπτει τις απαιτήσεις ενός συστήματος. Με άλλα λόγια, στόχος της είναι να εξασφαλίσει ότι οι δοκιμαστικές περιπτώσεις που χρησιμοποιούνται είναι επαρκείς για να ανιχνεύσουν τυχόν προβλήματα στο λογισμικό. Επίσης, τα κριτήρια της επάρκεια δοκιμών έχουν χρησιμοποιηθεί για να καθοδηγήσουν τη δημιουργία των δεδομένων δοκιμής.

Επιπρόσθετα, η Επάρκεια Δοκιμών είναι σημαντική διότι α) βελτιώνει την ποιότητα του λογισμικού (Ένα σύνολο δοκιμών με υψηλή Επάρκεια Δοκιμών έχει μεγαλύτερη πιθανότητα να ανιχνεύσει σφάλματα, οδηγώντας σε ένα πιο σταθερό και αξιόπιστο σύστημα.), β) μειώνει το κόστος συντήρησης (η έγκαιρη ανίχνευση σφαλμάτων κατά τη διάρκεια της δοκιμής μπορεί να εξοικονομήσει χρόνο και χρήματα που θα δαπανώνταν για την επιδιόρθωση σφαλμάτων μετά την εγκατάσταση του στο παραγωγικό περιβάλλον) και γ) αυξάνει την εμπιστοσύνη στο σύστημα: Ένα σύνολο δοκιμών με υψηλή επάρκεια Δοκιμών μπορεί να ενισχύσει την εμπιστοσύνη των χρηστών και των stakeholders (π.χ.: πελάτες, ιδιοκτήτες) του σύστημα, μειώνοντας τον κίνδυνο αποτυχιών.

5.1 Test Coverage

Κατά τη δομική αξιολόγησης των συμβατικών συστημάτων, η κάλυψη του κώδικα (code coverage) χρησιμοποιείται για να μετρήσει τον βαθμό εκτέλεσης του πηγαίου κώδικα από τη δοκιμαστική σουίτα. Όσο μεγαλύτερη είναι η κάλυψη που επιτυγχάνει η σουίτα δοκιμών, τόσο μεγαλύτερη είναι η πιθανότητα εντοπισμού των σφαλμάτων που υπάρχουν στον κώδικα.

Αντίθετα, η κάλυψη του κώδικα των ML συστημάτων δεν θεωρείται σημαντική διότι το ML μοντέλο βασίζεται στα δεδομένα για να λάβει τις αποφάσεις και όχι σε λογικές εκφράσεις και δομές ελέγχου (Zhang, Harman, Ma, & Yang, 2020). Για τον λόγο αυτό η ερευνητές πρότειναν νέα είδη αξιολόγησης της κάλυψης των δοκιμαστικών περιπτώσεων, τα σημαντικότερα εκ των οποίων αναλύονται στις επόμενες ενότητες, και είναι το Neural Coverage, το MC/DC coverage variants, το Layer-level coverage, και το State-Level coverage.

5.1.1 Neuron coverage

Το Neural coverage αξιολογεί την επάρκεια του συνόλου δοκιμών για ένα DNN. Ελέγχει εάν οι είσοδοι δοκιμών μπορούν να ενεργοποιήσουν τους νευρώνες (Zhang, Harman, Ma, & Yang,

2020). Σημειώνεται ότι, ο νευρώνας ενεργοποιείται όταν η τιμή εξόδου του προηγούμενου επιπέδου είναι μεγαλύτερη από το καθορισμένο κατώφλι ενεργοποίησης (το οποίο έχει οριστεί από τον χρήστη). Το Neural coverage υπολογίζεται από τον συνολικό αριθμό των νευρώνων που ενεργοποιούνται όταν εισάγονται δεδομένα δοκιμών στο DNN, ως προς το συνολικό αριθμό των νευρώνων.

Οι ερευνητές βασιζόμενοι στο Neural Coverage, κατάφεραν να αναπτύξουν περεταίρω την μέθοδο. Αναλυτικότερα, οι (Ma, et al., 2018) αρχικά χαρτογράφησαν ένα DNN (Usman, et al., 2022), προκειμένου να κατανοήσουν πως ενεργοποιούνται οι νευρώνες για τα διάφορα δεδομένα εκπαίδευσης και πρότειναν τα ακόλουθα κριτήρια κάλυψης των νευρώνων, τα οποία λαμβάνουν υπόψη τους όλες τις πιθανές συμπεριφορές ενός βαθέως νευρωνικού δικτύου:

- **k-multisection Neuron coverage:** Το «k-multisection neuron coverage» χωρίζει το εύρος των τιμών ενεργοποίησης των νευρώνων σε τμήματα (k-sections) και ελέγχει εάν τα δεδομένα δοκιμών ενεργοποιούν τα διάφορα τμήματα.
- **Neuron boundary coverage:** Το «Neuron boundary coverage» ελέγχει τις ενεργοποιήσεις των νευρώνων στις μέγιστες και ελάχιστες οριακές περιπτώσεις
- **Strong neuron activation coverage (SNAC):** Το SNAC επικεντρώνεται κυρίως στον έλεγχο των ακραίων περιπτώσεων (corner cases) λαμβάνοντας υπόψιν την ανώτερη οριακή τιμή.

5.1.2 MC/DC coverage variants

Η μέθοδος MC/DC αναπτύχθηκε αρχικά από τη NASA προκειμένου να αξιολογήσει τα υπολογιστικά της συστήματα. Στη συνέχεια, η μέθοδος προσαρμόστηκε για να χρησιμοποιηθεί στην αξιολόγηση των συστημάτων ML. Η μέθοδος «MC/DC coverage variants» θεωρεί ότι το δίκτυο DNN είναι πάντα πλήρως συνδεδεμένο (fully connected) και δεν λαμβάνει υπόψιν της ούτε τον τρόπο λειτουργίας του νευρώνα εντός του στρώματος του (layer), ούτε τις πιθανές εξαρτήσεις μεταξύ των νευρώνων του ίδιου στρώματος.

Αναλυτικότερα, η μέθοδος MC/DC χρησιμοποιείται στα συμβατικά συστήματα για να εντοπίσει αλλαγές που προκαλούνται όταν τροποποιείται μια Boolean μεταβλητή, ενώ η μέθοδος «MC/DC coverage variants» στοχεύει να εντοπίσει όλες οι πιθανές αλλαγές στη συμπεριφορά ενός νευρώνα που προκαλούνται από τις αλλαγές στις εισόδους δοκιμών (αλλαγή του πρόσημου (sign), της τιμής

(value) ή της απόστασης του νευρώνα από μια συγκεκριμένη τιμή) (Zhang, Harman, Ma, & Yang, 2020). Πιο συγκεκριμένα, στην εργασία αυτή προτάθηκαν οι ακόλουθες μετρικές:

- Sign-Sign coverage (SS)
- Sign-Value coverage (SV)
- Value-Sign coverage (VS)
- Value-Value coverage (VV)

5.1.3 Layer-level coverage

Η μεθοδολογία Layer-level coverage αξιολογεί την ποιότητα της σουίτας δοκιμών βασιζόμενη στους νευρώνες που ενεργοποιούνται περισσότερο (top hyperactive) σε κάθε επίπεδο και τις συνδυαστικές ενεργοποιήσεις τους (ή τις αλληλουχίες τους). Με τον τρόπο αυτό, η κάλυψη δοκιμών δεν εξετάζει μόνο τους μεμονωμένους νευρώνες, αλλά και τη συνολική συμπεριφορά του κάθε επιπέδου (Zhang, Harman, Ma, & Yang, 2020).

Αναλυτικότερα, οι (Ma, et al., 2018) αξιολόγησαν τα κριτήρια κάλυψης επιπέδων σε συνδυασμό με την κάλυψη μεμονωμένων νευρώνων (neuron coverage) χρησιμοποιώντας τα σύνολα δεδομένων MNIST[13] και ImageNet. Διαπίστωσαν ότι ο συνδυασμός αυτών των κριτηρίων οδήγησε σε καλύτερη απόδοση της κάλυψης δοκιμών. Επιπλέων, πρότειναν τη μεθοδολογία Combinatorial testing coverage, η οποία ελέγχει τη συνδυαστική κατάσταση ενεργοποίησης των νευρώνων του ίδιου στρώματος. Πρακτικά, αυτή η μεθοδολογία ελέγχει πόσοι νευρώνες του ίδιου στρώματος ενεργοποιούνται ταυτόχρονα για τις διάφορες τιμές εισόδου.

Επίσης, ο Sekhon και ο Fleming όρισαν κριτήρια κάλυψης τα οποία βασίζονται στα ζεύγη των νευρώνων (Sekhon & Fleming, 2019). Η προσέγγισή αυτή αναζητά:

- Για όλα τα ζεύγη των νευρώνων του ίδιου στρώματος, τους πιθανούς συνδυασμούς τιμών ενεργοποίησης.
- Για όλα τα ζεύγη των διαδοχικών στρωμάτων των νευρώνων, τους πιθανούς συνδυασμούς τιμών ενεργοποίησης.

[13] http://yann.lecun.com/exdb/mnist/

5.1.4 State-level coverage

Το State-level coverage χρησιμοποιείται για την αξιολόγηση των δοκιμαστικών περιπτώσεων που εφαρμόζονται σε stateful συστήματα ML. Αντίθετα με τα feed-forward ML μοντέλα, τα οποία δεν διατηρούν την κατάσταση τους μεταξύ των διαδοχικών εισόδων, τα stateful ML μοντέλα, όπως τα Recurrent Neural Networks (RNN), διατηρούν (αποθηκεύουν) πληροφορίες σχετικά με την ιστορία των εισόδων τους για να κάνουν προβλέψεις. Αυτό τα καθιστά κατάλληλα για εφαρμογές που βασίζονται σε ακολουθίες δεδομένων, όπως η μεταφράσεις, η αναγνώριση φωνής, η επεξεργασία της φυσικής γλώσσας, η ανάλυση χρονοσειρών (Zhang, Harman, Ma, & Yang, 2020) κ.ά.

Πιο συγκεκριμένα, οι ερευνητές πρότειναν ένα σύνολο κριτηρίων για τα stateful συστήματα ML. Αρχικά, θεώρησαν ότι αυτού του είδους τα συστήματα μπορούν να μεταβαίνουν πιθανολογικά από τη μία κατάσταση στην άλλη, ανταποκρινόμενα στις εισόδους του. Με βάση αυτή την μοντελοποίηση, πρότειναν κριτήρια που βασίζονται στην κατάσταση (state) και τα ίχνη (traces) των μεταβάσεων του συστήματος. Αυτά τα κριτήρια έχουν σκοπό να μοντελοποιοήσουν τη δυναμική συμπεριφορά των μεταβάσεων κατάστασης, δηλαδή το πώς αλλάζει η κατάσταση του συστήματος με την πάροδο του χρόνου ή μετά από συγκεκριμένες εισόδους.

5.2 Mutation testing (MuT)

Στα συμβατικά συστήματα, η μεθοδολογία Mutation Testing (MuT), χρησιμοποιείται για να αξιολογήσει την ποιότητας της σουίτας δοκιμών κάνοντας μικρές τροποποιήσεις (mutation) στον κώδικα (π.χ. αλλαγές τελεστών, ονόματα μεταβλητών). Στην παρούσα πτυχιακή, υιοθετούμε την προσέγγιση των (Zhang, Harman, Ma, & Yang, 2020), σύμφωνα με την οποία η σουίτα δοκιμών θεωρείται ποιοτική όταν οι δοκιμαστικές περιπτώσεις είναι αποτελεσματικές στην ανίχνευση σφαλμάτων. Μαθηματικά, η ποιότητα της υπολογίζεται από τον αριθμό των σφαλμάτων που εντοπίστηκαν από την σουίτα δοκιμών ως προς τα σφάλματα που εισήχθησαν στο υπό αξιολόγηση σύστημα, και καλείται «Mutation score».

Όπως έχει ήδη αναφερθεί, η συμπεριφορά των συστημάτων ML εξαρτάται από το πρόγραμμα εκπαίδευσης, τα δεδομένα, και από τη δομή του μοντέλου. Οι ερευνητές, βασιζόμενοι σε αυτά τα στοιχεία, καθώς και στο MuT που εφαρμόζεται στα συμβατικά συστήματα, δημιούργησαν τη μεθοδολογία DeepMutation για την αξιολόγηση των συστημάτων DL. Η DeepMutation προκαλεί

μεταλλάξεις (mutations) είτε στο επίπεδο του μοντέλου (π.χ: τροποποιήσεις στις συνδέσεις μεταξύ των νευρώνων, στις τιμές των βαρών των συνάψεων, ή ακόμα και στην αρχιτεκτονική του δικτύου), είτε στο επίπεδο του πηγαίου κώδικα (source code) προκειμένου να προκαλέσει ελαφρές τροποποιήσεις στο όριο αποφάσεων (decision boundary) του DNN. Σε αυτή την περίπτωση το «Mutation score» ορίζεται ως ο συνολικός αριθμός των δοκιμαστικών περιπτώσεων των οποίων το αποτέλεσμα έχει αλλάξει (εξ αιτίας των τροποποιήσεων στο κώδικα ή στο μοντέλο), ως προς τον συνολικό αριθμό των περιπτώσεων δοκιμής. Υψηλή τιμή για το «Mutation score» υποδηλώνει ότι οι περιπτώσεις δοκιμών είναι ευαίσθητες στις τροποποιήσεις και ενδεχόμενα προβλήματα στο σύστημα θα μπορέσουν να εντοπιστούν από την σουίτα δοκιμών (Ahuja, Gotlieb, & Spieker, 2022).

Όταν οι δοκιμαστικές περιπτώσεις επιτυγχάνουν να εντοπίσουν τις τροποποιήσεις στο DNN, τότε λέμε ότι η τροποποίηση «σκοτώθηκε» («was killed»). Αυτό σημαίνει ότι οι δοκιμαστικές περιπτώσεις εντόπισαν επιτυχώς την αλλαγή στο δίκτυο, υποδεικνύοντας πιθανό σφάλμα. Αντίθετα, όταν τα τεστ αποτύχουν να εντοπίσουν τις τροποποιήσεις στο DNN, τότε λέμε ότι η τροποποίηση «επιβίωσε» («survived»). Αυτό μπορεί να σημαίνει ότι οι δοκιμαστικές περιπτώσεις δεν είναι αρκετά ευαίσθητες για να εντοπίσουν την αλλαγή, θέτοντας υπό αμφισβήτηση την αποτελεσματικότητά τους.

Αξίζει να αναφέρουμε, ότι μεθοδολογία Mutation Testing έχει χρησιμοποιηθεί και ως μηχανισμός εντοπισμού κακόβουλων (adversarial) δεδομένων εισόδου. Αναλυτικότερα, οι ερευνητές παρατήρησαν ότι τα κακόβουλα δεδομένα εισόδου είναι πιο ευαίσθητα στα σημεία που έχει τροποποιηθεί (mutation) το μοντέλο DL, απ’ ότι τα αρχικά δεδομένα εισόδου (Zhang, Harman, Ma, & Yang, 2020). Αξιοποιώντας αυτή την παρατήρηση, δημιούργησαν μια μέθοδο η οποία προκαλεί μεταλλάξεις στο μοντέλο DL, και παρακολουθώντας τις αλλαγές στα δεδομένα εξόδου, μπορούν να καθορίσουν εάν η είσοδος ήταν κακόβουλη ή όχι. Στη συνέχεια, κάνοντας περεταίρω ανάλυση, προσδιορίζεται η ποιότητα δεδομένων δοκιμής και το επίπεδο ανίχνευσης σφαλμάτων.

5.3 Surprise Adequacy (SA)

Η μεθοδολογία Surprise Adequacy χρησιμοποιείται για την αξιολόγηση της ποιότητας της σουίτας δοκιμών μετρώντας το βαθμό ανομοιογένειας των δεδομένων δοκιμών (test set). Η SA θεωρεί ότι τα δεδομένα δοκιμών θα πρέπει να είναι διαφορετικά από τα δεδομένα εκπαίδευσης (training set)

του μοντέλου, αλλά ταυτόχρονα δεν θα πρέπει να αποκλίνουν κατά πολύ από κατανομή των δεδομένων εκπαίδευσης.

Άρα, το σύνολο των δεδομένα δοκιμών θεωρείται ποιοτικό όταν υπάρχουν:

- δεδομένα που είναι παρόμοια με τα δεδομένα εκπαίδευσης, ώστε να μπορεί να διασφαλίζεται ότι το μοντέλο λειτουργεί καλά σε γνωστά παραδείγματα
- *αναπάντεχα δεδομένα* (surprise data), δηλ. δεδομένα που δεν έχουν ξαναχρησιμοποιηθεί στο υπό αξιολόγηση μοντέλο, αλλά εξακολουθούν να είναι σχετικά με το πρόβλημα που καλείται να επιλύσει το μοντέλο.

Επομένως, η αξιολόγηση του μοντέλου ML με αυτό το είδους των δεδομένων δοκίμων, επιτρέπει στον Αξιολογητή να κατανοήσει τον τρόπο λειτουργίας και υπό την παρουσία δεδομένων που βρίσκονται εκτός της κανονικής κατανομής (out-of-distribution data) (Zhang, Harman, Ma, & Yang, 2020).

Η μετρική Surprise Adequacy μπορεί να υπολογιστεί με έναν από τους ακόλουθους τύπους:

- **Likelihood-based Surprise Adequacy (LSA):** Μετρά την πιθανότητα (likelihood) το σύστημα να έχει δει παρόμοια είσοδο κατά τη διάρκεια της εκπαίδευσης,
- **Distance-based Surprise Adequacy (DSA):** Μετρά την απόσταση μεταξύ των διανυσμάτων που αντιστοιχούν στα ίχνη (traces) ενεργοποίησης των νευρώνων για τα δεδομένα εισόδου, ως προς τα αντίστοιχά διανύσματα των δεδομένων εκπαίδευσης.

Σημειώνεται ότι και οι δύο μετρικές υπολογίζουν τη διαφορετικότητα των εισόδων δοκιμής ως προς τα δεδομένα εκπαίδευσης.

6 Ιεράρχηση και Μείωση των προς Εκτέλεση Δοκιμών

Τα συστήματα ML μπορούν να δεχθούν πολλά και διαφορετικά είδη δεδομένων (μεγάλος χώρος εισόδου). Επομένως, για την για την αξιολόγηση της συμπεριφοράς των συστημάτων ML για όλες τις πιθανές τιμές εισόδου, απαιτείται η δημιουργία ενός πολύ μεγάλου αριθμού δοκιμαστικών περιπτώσεων. Επίσης, σε κάθε δοκιμαστική περίπτωση απαιτείται να ανατεθούν ετικέτες (labels), δηλαδή θα πρέπει να εντοπιστεί η αναμενόμενη έξοδος για κάθε είσοδο δοκιμής, ώστε να μπορέσει να αξιολογηθεί η ακρίβεια πρόβλεψης του συστήματος ML (Zhang, Harman, Ma, & Yang, 2020). Αυτές οι δύο πτυχές, ο μεγάλος χώρος εισόδου και η απαίτηση ανάθεσης ετικετών, έχει ως αποτελέσματα το υψηλό κόστος δημιουργίας δοκιμαστικών περιπτώσεων για τα συστήματα ML.

Ο ερευνητές (Byun, Sharma, Vijayakumar, Rayadurgam, & Cofer, 2019) για να αντιμετωπίσουν αυτή την πρόκληση, και στο πλαίσιο της μείωσης του αριθμού των προς εκτέλεση δοκιμών πρότειναν τις ακόλουθες μετρικές οι οποίες βασίζονται στα DNN:

- **Cross entropy:** Εκφράζει την απόσταση μεταξύ της κατανομής πιθανότητας που προβλέπει το μοντέλο και της πραγματικής κατανομής.
- **Surprisal**: Εκφράζει το πόσο επηρεάζεται το μοντέλο για μια συγκεκριμένη είσοδο.
- **Bayesian uncertainty:** Εκφράζει την αβεβαιότητα πρόβλεψης του μοντέλου για μια συγκεκριμένη είσοδο.

Οι (Byun, Sharma, Vijayakumar, Rayadurgam, & Cofer, 2019) απέδειξαν πειραματικά ότι αυτές οι μετρικές βοηθούν στον εντοπισμό των πιθανών σφαλμάτων του συστήματος.

7 Συμπεράσματα

Η παρούσα διπλωματική εργασία αναδεικνύει τη σημασία των δοκιμών αξιολόγησης στα συστήματα μηχανικής μάθησης (ML) και προσφέρει μια ολοκληρωμένη επισκόπηση των διαδικασιών, μετρικών και μεθόδων που χρησιμοποιούνται για την αποτίμηση της απόδοσής τους. Η ανάλυση κατέδειξε τα ακόλουθα:

- Την πολυπλοκότητα και την κρισιμότητα της αξιολόγησης συστημάτων ML, αναδεικνύοντας παράλληλα την ανάγκη για ολοκληρωμένες και προσαρμοσμένες στρατηγικές, λαμβάνοντας υπόψη το εκάστοτε πρόβλημα και το επιθυμητό αποτέλεσμα.
- Την αναγκαιότητα ύπαρξης προκαθορισμένων κριτηρίων αποδοχής, η οποία καθιστά ευκολότερη την λήψη τεκμηριωμένων αποφάσεων για τη βελτίωση του συστήματος.
- Η επιλογή των κατάλληλων μετρικών αξιολόγησης είναι κρίσιμη και οφείλει να εναρμονίζεται με τους στόχους του συστήματος ML.
- Η διεξαγωγή πολλαπλών δοκιμών αξιολόγησης με διαφορετικές μεθόδους και δεδομένα ενισχύει την αξιοπιστία των αποτελεσμάτων.

Βιβλιογραφία

Ahuja, M. K., Gotlieb, A., & Spieker, H. (2022). Testing Deep Learning Models: A First Comparative Study of Multiple Testing Techniques. *IEEE*, 130-137. doi:10.1109/icstw55395.2022.00035

Alex Krizhevsky, I. S. (2012). ImageNet Classification with Deep Convolutional. Στο *Advances in Neural Information Processing Systems 25 (NIPS 2012)* (σ. 9).

Bandi, A., Pydi, A. V., & Yudu, E. V. (2023). The Power of Generative AI: A Review of Requirements,. *Future Internet, 15*. doi:10.3390/fi15080260

Breck , E., Cai, S., & Nielsen, E. (2017). The ML test score: A rubric for ML production readiness and technical debt reduction. *2017 IEEE International Conference on Big Data (Big Data)*, 1123-1132. Ανάκτηση από https://api.semanticscholar.org/CorpusID:6244440

Byun, T., Sharma, V., Vijayakumar, A., Rayadurgam, S., & Cofer, D. (2019). Input Prioritization for Testing Neural Networks. *2019 IEEE International Conference On Artificial Intelligence Testing (AITest)* (σσ. 63-70). Newark, CA, USA: IEEE. doi:10.1109/AITest.2019

Chandrasekaran, J., Cody, T., Mccarthy, N., Lanus, E., & Freeman, L. (2023). Test & Evaluation Best Practices for Machine Learning-Enabled Systems. *arXiv*. doi:10.48550/arXiv.2310.06800

Chen, M., Tworek, J., Jun, H., Yuan, Q., Kaplan, J., Edwards, H., . . . Krueger, G. (2021). Evaluating Large Language Models Trained on Code. *arXiv*.

Donges, N. (2019, 6 16). *A GUIDE TO RNN: UNDERSTANDING RECURRENT NEURAL NETWORKS AND LSTM*. Ανάκτηση από https://builtin.com/data-science/recurrent-neural-networks-and-lstm

Dussa-Zieger, K., Henschelchen, W., Kocher, V., Liu, Q., Reid, S., Siemens, K., & Smith, A. (2021). *Certified Tester AI Testing (CT-AI) Syllabus Version 1.0*. ISTQB.

Google. (2024). *Evaluating models BLUE*. Ανάκτηση από Google: https://cloud.google.com/translate/automl/docs/evaluate

Guo, J., Jiang, Y., Zhao, Y., Chen, Q., & Sun, J. (2018). DLFuzz: differential fuzzing testing of deep learning systems. In. *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2018)* (σσ. 739–743). New York, NY, USA: Association for Computing Machinery. doi:https://doi.org/10.1145/3236024.3264835

Kaiming He, X. Z. (2016, 12 12). Deep Residual Learning for Image Recognition. *IEEE*, σ. 9.

Karen Simonyan, A. Z. (2015). VERY DEEP CONVOLUTIONAL FOR. *ICLR 2015*, σ. 14.

Lee, S., Lee, J., Moon, H., Park, C., Seo, J., Eo, S., . . . Lim, H. (2023). A Survey on Evaluation Metrics for Machine Translation. *Mathematics*. doi:10.3390/math11041006

Ma, L., Juefei-Xu, F., Zhang, F., Sun, J., Xue, M., Li, B., . . . Wang, Y. (2018). DeepGauge: Multi-Granularity Testing Criteria for Deep Learning Systems. *The 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE 2018)*. Montpellier, France. doi:https://doi.org/10.1145/3238147.3238202

Naidu, G., Zuva, T., & Sibanda, E. M. (2023). A Review of Evaluation Metrics in Machine Learning Algorithms. *Springer, Cham*. doi:10.1007/978-3-031-35314-7_2

Nilsson, J., & Akenine-Möller, T. (2020). Understanding SSIM. *arXiv*.

Pei, K., Cao, Y., Yang, J., & Jana, S. (2017). DeepXplore: Automated Whitebox Testing of Deep Learning Systems. doi:10.1145/3132747.3132785

Pham, H. V., Lutellier, T., Qi, W., & Tan, L. (2019). CRADLE: Cross-Backend Validation to Detect and Localize Bugs in Deep Learning Libraries. *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, (σσ. 1027-1038). Montreal, QC, Canada.

Phi, M. (2018, 9 24). Illustrated Guide to LSTM's and GRU's: A step by step explanation. *https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21*.

Rainio, O., Teuho, J., & Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports, 14*. doi:10.1038/s41598-024-56706-x

Ren, S., Guo, D., Lu, S., Zhou, L., Liu, S., Tang, D., . . . Ma, S. (2020). CodeBLEU: a Method for Automatic Evaluation of Code Synthesis. *arXiv*.

Riccio, V., Jahangirova, G., Stocco, A., Humbatova, N., Weiss, M., & Tonella, P. (2020). Testing machine learning based systems: a systematic mapping. *Empirical Software Engineering*, 5193-5254. doi:10.1007/s10664-020-09881-0

Sarker, & Iqbal, H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*. doi:10.1007/s42979-021-00592-x

Sekhon, J., & Fleming, C. (2019). Towards Improved Testing For Deep Learning. *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)* (σσ. 85-88). Montreal, QC, Canada: IEEE. doi:10.1109/ICSE-NIER.2019.00030

Terven, J., Córdova-Esparza, D. M., Ramírez-Pedraza, A., & Chávez-Urbiola, E. A. (2023). LOSS FUNCTIONS AND METRICS IN DEEP LEARNING. *arXiv*. doi:10.48550/arXiv.2307.02694

Usman, M., Sun, Y., Gopinath, D., Dange, R., Manolache, L., & P˘as˘areanu, C. (2022). An Overview of Structural Coverage Metrics for Testing Neural Networks. *arXiv*. Ανάκτηση από https://arxiv.org/pdf/2208.03407

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., . . . Polosukhin, I. (2017). Attention Is All You Need. *CoRR*. doi:https://dblp.org/rec/journals/corr/VaswaniSPUJGKP17.bib

Vedantam, R., Zitnick, L. C., & Parikh, D. (2015). CIDEr: Consensus-based Image Description Evaluation. *arXiv*. doi:1411.5726v2

XIE, X., CHEN, H., LI, Y., MA, L., LIU, Y., & and ZHAO, J. (2019). Coverage-guided fuzzing for feedforward neural networks. *Proceedings of the 34th IEEE/ACM International Conference on Automated Software Engineering*, (pp. 1162-1165). San Diego.

Zhang, J. M., Harman, M., Ma, L., & Yang, L. (2020). Machine Learning Testing: Survey, Landscapes and Horizons. *Transactions on Software Engineering*, 1-36. doi:10.1109/TSE.2019.2962027