ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

# «Αυτόματη Εξαγωγή Θεματικών Λέξεων-Κλειδιών στον Τομέα της Κυβερνοασφάλειας με Χρήση SKOS Οντολογιών και Αλγορίθμων Μηχανικής Μάθησης»


**Του φοιτητή**

**ΒΟΛΟΝΑΚΗ ΠΑΝΤΕΛΕΗΜΩΝ**

**Αρ. Μητρώου: 1/2023**

**Επιβλέπων**

**ΧΑΡΑΛΑΜΠΟΣ ΜΠΡΑΤΣΑΣ**

**Βαθμίδα: Επίκουρος Καθηγητής**


**Ημερομηνία 05/09/2025**

Τίτλος Δ.Ε.: «Αυτόματη Εξαγωγή Θεματικών Λέξεων-Κλειδιών στον Τομέα της Κυβερνοασφάλειας με Χρήση SKOS Οντολογιών και Αλγορίθμων Μηχανικής Μάθησης».
Κωδικός Δ.Ε. 25203
Ονοματεπώνυμο φοιτητή: Παντελεήμων Βολονάκης
Ονοματεπώνυμο εισηγητή: Χαράλαμπος Μπράτσας
Ημερομηνία ανάληψης Δ.Ε: 24/03/2025
Ημερομηνία περάτωσης Δ.Ε.: 05/09/2025

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



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

# Πρόλογος

Η επιλογή του θέματος της παρούσας διπλωματικής εργασίας υπήρξε στοχευμένη και άμεσα συνδεδεμένη με το προσωπικό μου ενδιαφέρον στον τομέα της κυβερνοασφάλειας. Πρόκειται για έναν χώρο που εξελίσσεται διαρκώς, με συνεχώς αυξανόμενες απαιτήσεις για αξιόπιστη, έγκαιρη και καλά οργανωμένη πληροφόρηση. Στην πράξη, οι περιγραφές και τα δεδομένα που αφορούν κυβερνο-απειλές, όπως ευπάθειες, επιθέσεις και αδυναμίες συστημάτων, είναι συχνά κατακερματισμένα, μη τυποποιημένα και δύσχρηστα για αυτόματη επεξεργασία.

Η επιλογή των SKOS οντολογιών ως βάσης για την αναπαράσταση των θεματικών εννοιών του πεδίου, προέκυψε από την ανάγκη για ένα πρότυπο που να προσφέρει ευελιξία, επεκτασιμότητα και δυνατότητα διασύνδεσης δεδομένων. Με αυτόν τον τρόπο, οι έννοιες που σχετίζονται με την κυβερνοασφάλεια όπως οι κατηγορίες του CWE (Common Weakness Enumeration) μπορούν να οργανώνονται σημασιολογικά, να συνδέονται ιεραρχικά και να γίνονται ευκολότερα αναζητήσιμες και αξιοποιήσιμες.

Η συνδυαστική αξιοποίηση αλγορίθμων μηχανικής μάθησης για την αυτόματη εξαγωγή θεματικών λέξεων-κλειδιών από περιγραφές κυβερνο-ευπαθειών, με την ταυτόχρονη χρήση της οντολογικής γνώσης που προσφέρουν τα SKOS λεξιλόγια, επέτρεψε την ανάπτυξη ενός συστήματος ικανού να προτείνει συναφείς έννοιες με μεγαλύτερη ακρίβεια και συνοχή. Μέσα από την ανάπτυξη και αξιολόγηση αυτού του συστήματος, απέκτησα βαθύτερη κατανόηση τόσο των τεχνικών επεξεργασίας φυσικής γλώσσας και εκπαίδευσης μοντέλων, όσο και των πρακτικών οργάνωσης θεματικής πληροφορίας με σημασιολογικά πρότυπα.

# Περίληψη

Στην παρούσα εργασία υλοποιήθηκε και αξιολογήθηκε ένα σύστημα αυτόματης θεματικής ανάθεσης (automatic subject indexing) για την κατηγοριοποίηση και εξαγωγή λέξεων-κλειδιών από περιγραφές ευπαθειών στον τομέα της κυβερνοασφάλειας. Η υλοποίηση βασίστηκε στο Annif, ένα εργαλείο ανοιχτού κώδικα για αυτόματη θεματική ευρετηρίαση, το οποίο αξιοποιήθηκε σε συνδυασμό με την οντολογία Common Weakness Enumeration (CWE) σε μορφή SKOS. Η μεθοδολογία περιλάμβανε την συλλογή και προεπεξεργασία επιλεγμένων δεδομένων που περιείχαν ζεύγη περιγραφών ευπαθειών και αντίστοιχων URIs της CWE, την εκπαίδευση μοντέλων μηχανικής μάθησης, με κύριο backend το Omikuji, που είναι κατάλληλο για ταξινόμηση πολλαπλών ετικετών (extreme multi-label classification) αλλά και άλλους αλγορίθμους. Δημιουργήθηκε φιλικό προς τον χρήστη user interface (UI), μέσω του οποίου το εκπαιδευμένο μοντέλο μπορεί να προβλέψει τις πλέον συναφείς θεματικές κατηγορίες CWE για άγνωστες περιγραφές ευπαθειών. Το σύστημα υποστηρίζει τη διαδικασία ημιαυτόματης κατηγοριοποίησης σε εφαρμογές κυβερνοασφάλειας, ανάλυσης απειλών και διαχείρισης ευπαθειών, μειώνοντας τον χρόνο και την ανθρώπινη προσπάθεια που απαιτείται για την ταξινόμηση μεγάλου όγκου δεδομένων. Η απόδοση του τελικού μοντέλου αξιολογήθηκε με τη χρήση καθιερωμένων μετρικών, όπως Precision, Recall, F1-score, MAP και nDCG, αποδεικνύοντας την αποτελεσματικότητά του στην αυτόματη θεματική ανάθεση.

«Automated Thematic Keyword Extraction in Cybersecurity through SKOS Ontologies and Machine Learning Approaches»

«Pantelis Volonakis»

## Abstract

In this thesis, an automatic subject indexing system was developed and evaluated for the classification and keyword extraction from vulnerability descriptions in the field of cybersecurity. The implementation was based on Annif, an open-source tool for automated subject indexing, combined with the Common Weakness Enumeration (CWE) ontology in SKOS format. The methodology included collecting and preprocessing selected data consisting of pairs of vulnerability descriptions and corresponding CWE URIs, as well as training machine learning models, primarily using the Omikuji backend, which is suitable for extreme multi-label classification, along with other algorithms. A user-friendly UI was developed, enabling the trained model to predict the most relevant CWE thematic categories for unknown vulnerability descriptions. The system supports semi-automatic categorization in cybersecurity, threat analysis, and vulnerability management applications, reducing the time and human effort required to classify large volumes of data. The performance of the final model was evaluated using established metrics such as Precision, Recall, F1-score, MAP, and nDCG, demonstrating its effectiveness in automated subject indexing.

# Ευχαριστίες

Ευχαριστώ θερμά τον επιβλέποντα καθηγητή και την ερευνητική του ομάδα για την καθοδήγηση, την επιστημονική υποστήριξη και την ενθάρρυνση που μου παρείχαν. Η πολύτιμη βοήθεια και οι εποικοδομητικές παρατηρήσεις τους συνέβαλαν καθοριστικά στη βελτίωση της ποιότητας της εργασίας αυτής.

Θα ήθελα επίσης να εκφράσω τις ευχαριστίες μου, στη σύζυγό μου και στα παιδιά μου για την κατανόηση και την υπομονή τους καθ' όλη τη διάρκεια της εκπόνησης της παρούσας διπλωματικής εργασίας. Ο χρόνος που αφιέρωσα στην έρευνα και τη συγγραφή της σήμαινε λιγότερες στιγμές μαζί τους, και εκτιμώ βαθιά την αγάπη και την αφοσίωσή τους, που μου έδωσαν τη δύναμη να συνεχίσω.

# Περιεχόμενα

# Κατάλογος Σχημάτων



# Κατάλογος Πινάκων

# Συντομογραφίες

| | |
|---|---|
| APTs | Advanced Persistent Threats |
| ASI | Automatic Subject Indexing |
| ATT&CK | Adversarial Tactics Techniques, and Common Knowledge |
| BM25 | Best Matching 25 |
| CIA | Confidentiality, Integrity, Availability |
| CNNs | Convolutional Neural Networks |
| CSF | Cybersecurity Framework |
| CTI | Cyber Threat Intelligence |
| CVE | Common Vulnerabilities and Exposures |
| CVSS | Common Vulnerability Scoring System |
| CWE | Common Weakness Enumeration |
| DDoS | Distributed Denial of Service |
| ENISA | European Union Agency for Cybersecurity |
| IR | Information Retrieval |
| LCSH | The Library of Congress Subject Headings |
| LDA | Latent Dirichlet Allocation |
| LLMs | Large Language Models |
| LMIR | Language Models for IR |
| LSA | Latent Semantic Analysis |
| MAP | Mean Average Precision |
| ML | Machine Learning |
| nDCG | normalized Discounted Cumulative Gain |
| NIST | National Institute of Standards and Technology |
| NLP | Natural Language Processing |
| NVD | National Vulnerability Database |
| OWL | Web Ontology Language |
| RaaS | Ransomware-as-a-Service |
| RDF | Resource Description Framework |
| RNNs | Recurrent Neural Networks |
| SKOS | Simple Knowledge Organization System |
| SVMs | Support Vector Machines |

| | |
|---|---|
| TF-IDF | Term Frequency – Inverse Document Frequency |
| UI | User Interface |
| URI | Uniform Resource Identifier |
| VSM | Vector Space Model |
| W3C | World Wide Web Consortium |

# Κεφάλαιο 1ο: Εισαγωγή

Η τεχνολογική ανάπτυξη προχωρά με αλματώδεις ρυθμούς, οδηγώντας σε αυξημένες ανάγκες για νέα προϊόντα και υπηρεσίες, ενώ ταυτόχρονα ενισχύεται η εξάρτησή μας από την τεχνολογία. Αυτή η εξάρτηση συνοδεύεται και από την αυξανόμενη σημασία της κυβερνοασφάλειας, καθώς η καθημερινή μας δραστηριότητα στο διαδίκτυο παράγει ολοένα και περισσότερα δεδομένα προσωπικού χαρακτήρα [1]. Όσο περισσότερο παραμένουμε συνδεδεμένοι, τόσο αυξάνεται η πιθανότητα να αποτελέσουμε στόχο κυβερνοεπιθέσεων ή κυβερνοεγκλημάτων.

Στο σύγχρονο ψηφιακό περιβάλλον, η ασφάλεια της πληροφορίας αποτελεί κρίσιμο παράγοντα για οργανισμούς, επιχειρήσεις, κυβερνήσεις αλλά και ιδιώτες. Η κυβερνοασφάλεια περιλαμβάνει την πρόληψη, την ανίχνευση, την απόκριση και την ανάκαμψη από περιστατικά που λαμβάνουν χώρα στον κυβερνοχώρο [2]. Τα περιστατικά αυτά μπορεί να κυμαίνονται από την τυχαία ή σκόπιμη διαρροή πληροφοριών, έως στοχευμένες επιθέσεις σε οργανισμούς και κρίσιμες υποδομές, την κλοπή ευαίσθητων δεδομένων, ακόμη και την παρέμβαση σε δημοκρατικές διαδικασίες. Οι επιπτώσεις τέτοιων επιθέσεων μπορεί να είναι σημαντικές, επηρεάζοντας άτομα, οργανισμούς, κοινότητες και κράτη.

Η επίθεση WannaCry τον Μάιο του 2017 αποτέλεσε ένα από τα πιο καταστροφικά ransomware περιστατικά επηρεάζοντας πάνω από 200.000 συστήματα σε 150 χώρες, αξιοποιώντας το exploit 'EternalBlue' και προκαλώντας εκτεταμένες διακοπές υπηρεσιών, όπως στο βρετανικό NHS, με οικονομικό και κοινωνικό αντίκτυπο [3]. Λίγα χρόνια αργότερα, η επίθεση SolarWinds το 2020 αποκάλυψε την ευαλωτότητα των συστημάτων λογισμικού αλυσίδας εφοδιασμού. Ένας κακόβουλος κώδικας είχε εισαχθεί στην πλατφόρμα Orion, επηρεάζοντας τουλάχιστον 18.000 οργανισμούς, συμπεριλαμβανομένων ομοσπονδιακών υπηρεσιών των ΗΠΑ, αποδεικνύοντας τη σοβαρότητα των επιθέσεων που στρέφονται στον πυρήνα της ψηφιακής υποδομής [4].

Η ραγδαία αύξηση και η πολυπλοκότητα των κυβερνοεπιθέσεων καθιστούν επιτακτική την ανάγκη για αποτελεσματική διαχείριση της πληροφορίας που σχετίζεται με κυβερνοαπειλές (Cyber Threat Intelligence – CTI). Για να επιτευχθεί αυτό, οι οργανισμοί χρειάζονται καλά δομημένα και σαφώς ορισμένα πλαίσια εννοιών και πληροφορίας, ικανά να υποστηρίξουν την αναγνώριση, την ταξινόμηση και την ανταλλαγή δεδομένων για τις απειλές αυτές [1].

Στο πλαίσιο αυτό, τα πρότυπα σημασιολογικού ιστού, οι οντολογίες και οι θησαυροί όρων αποτελούν ισχυρά εργαλεία για την αναπαράσταση και οργάνωση της γνώσης. Σε αντίθεση με απλές λίστες ή ιεραρχίες, η χρήση του SKOS (Simple Knowledge Organization System) προσφέρει τη δυνατότητα ομαδοποίησης και σύνδεσης εννοιών [5], διευκολύνοντας την ενοποίηση, την ανάλυση και την ανταλλαγή πληροφοριών στον τομέα της κυβερνοασφάλειας.

## 1.1 Σκοπός και Στόχοι της Εργασίας

Η παρούσα εργασία διερευνά πώς οι SKOS οντολογίες μπορούν να συνδυαστούν με αλγορίθμους μηχανικής μάθησης για την αυτόματη εξαγωγή θεματικών λέξεων-κλειδιών από περιγραφές κυβερνοευπαθειών και συναφών περιστατικών. Στόχος είναι η ανάπτυξη μιας διαδικασίας που ταξινομεί και αποδίδει σχετικές έννοιες του CTI (όπως απειλές, τύποι επιθέσεων, κακόβουλο λογισμικό και τεχνικές) σε αδόμητα κείμενα, αξιοποιώντας το λεξιλόγιο του CWE [6] ή άλλα συναφή πρότυπα.

Η εργασία εστιάζει επίσης στη διασύνδεση της προτεινόμενης μεθοδολογίας με διεθνώς αναγνωρισμένα πρότυπα, όπως το MITRE ATT&CK [7] με σκοπό τη βελτίωση της διαλειτουργικότητας μεταξύ διαφορετικών συστημάτων και εργαλείων κυβερνοασφάλειας.

Τελικός στόχος είναι να γεφυρωθεί το χάσμα ανάμεσα στη συλλογή δεδομένων και την οργανωμένη παρουσίασή τους σε μορφή μεταδεδομένων, ώστε να διευκολύνεται η ανάλυση και η ανταλλαγή πληροφοριών. Με αυτό τον τρόπο, η εργασία συμβάλλει στη βελτίωση των πρακτικών διαχείρισης της πληροφορίας και στην ενίσχυση της αποτελεσματικότητας των διαδικασιών CTI.

## 1.2 Επιστημονικό και Ερευνητικό Πλαίσιο

Η κυβερνοασφάλεια έχει εξελιχθεί σε έναν ιδιαίτερα διεπιστημονικό τομέα, συνδυάζοντας τεχνικές, οργανωσιακές και νομικές διαστάσεις. Η έννοια του Cyber Threat Intelligence (CTI) έχει αναδειχθεί ως βασικό στοιχείο για την κατανόηση και αντιμετώπιση των απειλών, καθώς εστιάζει στη συστηματική συλλογή και ανάλυση δεδομένων που αφορούν κυβερνοεπιθέσεις και ευπάθειες [1]. Πηγές όπως το CVE, το CWE και το NVD αποτελούν θεμέλια για την τυποποιημένη καταγραφή και κατηγοριοποίηση ευπαθειών, ενώ πλαίσια όπως το MITRE ATT&CK παρέχουν μια ολοκληρωμένη εικόνα των τακτικών και τεχνικών που χρησιμοποιούν οι επιτιθέμενοι.

Σε αυτό το περιβάλλον, η μηχανική μάθηση προσφέρει νέα εργαλεία για την αυτοματοποίηση διαδικασιών, όπως η ταξινόμηση κειμένων και η θεματική ευρετηρίαση. Η συνδυαστική χρήση SKOS οντολογιών με αλγορίθμους μηχανικής μάθησης μπορεί να βελτιώσει την ακρίβεια και την αποτελεσματικότητα της θεματικής κατηγοριοποίησης, παρέχοντας μια ισχυρή βάση για την ανάπτυξη καινοτόμων εργαλείων στον χώρο της κυβερνοασφάλειας.

## 1.3 Δομή της εργασίας

Η εργασία οργανώνεται σε επτά διακριτά κεφάλαια, τα οποία αναπτύσσονται με λογική ακολουθία, ξεκινώντας από το θεωρητικό υπόβαθρο και καταλήγοντας στην παρουσίαση των αποτελεσμάτων και των συμπερασμάτων.

Στο Κεφάλαιο 1 (Εισαγωγή) παρουσιάζεται το γενικό πλαίσιο της έρευνας, η σημασία της κυβερνοασφάλειας στο σύγχρονο ψηφιακό περιβάλλον, καθώς και τα κίνητρα που οδήγησαν στην επιλογή του θέματος. Αναλύονται ο σκοπός και οι στόχοι της εργασίας και περιγράφεται το επιστημονικό και ερευνητικό πλαίσιο μέσα στο οποίο εντάσσεται.

Το Κεφάλαιο 2 (Ανάκτηση και Οργάνωση Πληροφορίας) εστιάζει στο θεωρητικό υπόβαθρο της Ανάκτησης Πληροφορίας (IR), παρουσιάζοντας κλασικά και σύγχρονα μοντέλα αναζήτησης και κατηγοριοποίησης κειμένων. Στη συνέχεια, εισάγεται η έννοια των οντολογιών και αναλύεται ο ρόλος τους στην οργάνωση της γνώσης. Ιδιαίτερη έμφαση δίνεται στο πρότυπο SKOS (Simple Knowledge Organization System), περιγράφοντας τις βασικές κλάσεις και ιδιότητες, καθώς και τις εφαρμογές του στην αναπαράσταση θεματικών λεξιλογίων. Τέλος, εξετάζεται η χρήση SKOS οντολογιών σε σύγχρονες εφαρμογές πληροφορίας και η συμβολή τους στη βελτίωση της αναζήτησης και ευρετηρίασης.

Στο Κεφάλαιο 3 (Κυβερνοασφάλεια και Ευπάθειες) παρουσιάζονται οι θεμελιώδεις αρχές της κυβερνοασφάλειας μέσα από το τρίπτυχο CIA Triad (Confidentiality, Integrity, Availability) και τις επεκτάσεις του. Ακολουθεί η ανάλυση των κατηγοριών απειλών και επιθέσεων, η επισκόπηση διεθνών προτύπων και οργανισμών που δραστηριοποιούνται στον χώρο (NIST, ISO, ENISA) και τέλος, γίνεται εκτενής αναφορά στις ευπάθειες ασφάλειας και στις βάσεις δεδομένων CVE, CWE και NVD, που αποτελούν θεμέλια εργαλεία για την τυποποιημένη καταγραφή και κατηγοριοποίησή τους.

Το Κεφάλαιο 4 (Μηχανική Μάθηση και Ανάλυση Κειμένων) συνδέει το προηγούμενο θεωρητικό πλαίσιο με τις τεχνικές προσεγγίσεις που εφαρμόστηκαν. Εισάγεται η έννοια της αυτόματης θεματικής ευρετηρίασης και αναλύονται οι κατηγορίες της μηχανικής μάθησης. Στη συνέχεια, παρουσιάζονται οι

βασικοί αλγόριθμοι που υποστηρίζει το εργαλείο Annif (TF-IDF, fastText, Omikuji, nn_ensemble), οι δυνατότητες παραμετροποίησής τους, η ροή εκπαίδευσης–πρόβλεψης–αξιολόγησης, καθώς και οι μετρικές που χρησιμοποιούνται για την εκτίμηση της απόδοσης.

Στο Κεφάλαιο 5 (Μεθοδολογία) περιγράφεται η πρακτική υλοποίηση της έρευνας. Αναλύεται η διαδικασία εγκατάστασης και ρύθμισης του Annif, η συλλογή και προετοιμασία του συνόλου δεδομένων, η δημιουργία της δομής του project, καθώς και η εκπαίδευση και αξιολόγηση των μοντέλων με βάση τις επιλεγμένες τεχνικές.

Το Κεφάλαιο 6 (Αποτελέσματα) συγκεντρώνει και παρουσιάζει τα αποτελέσματα των πειραμάτων, τόσο σε πίνακες όσο και σε γραφήματα, επιτρέποντας τη συγκριτική αποτίμηση των διαφορετικών αλγορίθμων και παραμετροποιήσεων.

Στο Κεφάλαιο 7 (Συζήτηση Αποτελεσμάτων και Συμπεράσματα) γίνεται η συνολική αποτίμηση της έρευνας, συζητούνται τα ευρήματα σε σχέση με τους στόχους και την υπάρχουσα βιβλιογραφία και προτείνονται μελλοντικές κατευθύνσεις για τη βελτίωση της μεθοδολογίας και την αξιοποίηση επιπλέον πηγών δεδομένων.

Τέλος, η εργασία ολοκληρώνεται με τη Βιβλιογραφία και δύο Παραρτήματα, τα οποία περιλαμβάνουν τη δομή του GitLab repository που χρησιμοποιήθηκε, καθώς και αποσπάσματα κώδικα Python που υποστηρίζουν την αναπαραγωγιμότητα των πειραμάτων.

## 1.4 Επίλογος

Το παρόν κεφάλαιο ανέδειξε τη σημασία της κυβερνοασφάλειας στο σύγχρονο ψηφιακό περιβάλλον, παρουσιάζοντας ταυτόχρονα το θεωρητικό πλαίσιο και το ερευνητικό υπόβαθρο της εργασίας. Η ανάλυση του πεδίου, η παρουσίαση των βασικών προτύπων και η σύνδεση με τις οντολογίες SKOS θέτουν τις βάσεις για την κατανόηση της μεθοδολογίας που ακολουθεί. Συνολικά, η εργασία φιλοδοξεί να συμβάλει στην πρόοδο της αυτόματης θεματικής ευρετηρίασης στον τομέα της κυβερνοασφάλειας, συνδυάζοντας εργαλεία σημασιολογικού ιστού και μηχανικής μάθησης για την αντιμετώπιση σύγχρονων προκλήσεων.

# Κεφάλαιο 2ο: Ανάκτηση και Οργάνωση Πληροφορίας

## 2.1 Εισαγωγή

Η Ανάκτηση Πληροφορίας (Information Retrieval – IR) είναι ο κλάδος της Πληροφορικής που ασχολείται με τον εντοπισμό και την ανάκτηση σχετικών πληροφοριών από μεγάλες συλλογές δεδομένων, συνήθως σε μορφή κειμένου [8]. Σε αντίθεση με τα κλασικά συστήματα βάσεων δεδομένων, τα οποία βασίζονται σε ακριβή ερωτήματα και δομημένα δεδομένα, τα συστήματα IR στοχεύουν στη διαχείριση αδόμητης πληροφορίας, όπως φυσική γλώσσα, έγγραφα, ιστοσελίδες και πολυμέσα.

## 2.2 Ιστορική Εξέλιξη

Η IR έχει τις ρίζες της ήδη από τη δεκαετία του 1950, όταν οι πρώτες προσπάθειες επικεντρώνονταν σε μηχανογραφημένα συστήματα βιβλιοθηκών. Τα συστήματα αυτά χρησιμοποιούσαν απλές μεθόδους αντιστοίχισης λέξεων-κλειδιών με καταλόγους εγγράφων, προσπαθώντας να διευκολύνουν την αναζήτηση σε επιστημονικά άρθρα και βιβλία [9]. Κατά τη δεκαετία του 1960, αναπτύχθηκαν οι πρώτες πειραματικές πλατφόρμες IR, όπως το SMART του Gerard Salton, το οποίο αποτέλεσε θεμέλιο λίθο για τις επόμενες δεκαετίες [10].

Στη δεκαετία του 1970 και 1980, καθιερώθηκαν βασικές έννοιες που εξακολουθούν να χρησιμοποιούνται έως σήμερα. Το Boolean Retrieval, με χρήση λογικών τελεστών (AND, OR, NOT), επέτρεψε πιο σύνθετες αναζητήσεις, ενώ οι inverted indexes (ανεστραμμένοι δείκτες) βελτίωσαν την αποδοτικότητα των συστημάτων, καθιστώντας δυνατή την ταχύτατη ανάκτηση σχετικών εγγράφων [11].

Κατά τη δεκαετία του 1990, η έρευνα στράφηκε σε πιο εκλεπτυσμένα μαθηματικά μοντέλα. Το Vector Space Model εισήγαγε την αναπαράσταση εγγράφων και ερωτημάτων ως διανυσμάτων σε πολυδιάστατο χώρο, επιτρέποντας τον υπολογισμό ομοιότητας με μετρικές όπως η cosine similarity. Την ίδια περίοδο, η μέθοδος TF-IDF (Term Frequency – Inverse Document Frequency) καθιερώθηκε ως βασική τεχνική στάθμισης όρων, βελτιώνοντας την ακρίβεια και την ικανότητα διάκρισης μεταξύ εγγράφων [12].

Στις αρχές του 2000, το BM25 (Best Matching 25) αναδείχθηκε ως εξέλιξη του Probabilistic Relevance Framework, ενσωματώνοντας παράγοντες όπως το μήκος του εγγράφου και τη συχνότητα εμφάνισης όρων. Το BM25 αποτέλεσε ένα από τα πιο επιτυχημένα και διαδεδομένα μοντέλα για web search engines και παραμένει σημείο αναφοράς στην αξιολόγηση συστημάτων IR [13].

Από τη δεκαετία του 2010 έως σήμερα, η IR έχει επηρεαστεί καταλυτικά από τις εξελίξεις στη Μηχανική Μάθηση και την Επεξεργασία Φυσικής Γλώσσας (NLP). Τα deep learning μοντέλα, όπως τα convolutional και recurrent neural networks, άρχισαν να εφαρμόζονται σε προβλήματα IR, ενώ η εισαγωγή των pretrained language models (π.χ. BERT, GPT) άλλαξε ριζικά το πεδίο. Συστήματα όπως τα BERT-based retrievers και τα Dense Passage Retrievers (DPR) εκμεταλλεύονται contextual embeddings, επιτυγχάνοντας σαφώς καλύτερες επιδόσεις σε σχέση με τα κλασικά στατιστικά μοντέλα. Σήμερα, η έρευνα επικεντρώνεται σε cross-lingual IR, σε πολυτροπικά δεδομένα (εικόνα–κείμενο) και στην ενοποίηση IR με μεγάλα γλωσσικά μοντέλα (LLMs), ανοίγοντας νέους δρόμους για εφαρμογές σε αναζήτηση, ερωταπαντήσεις και knowledge graphs.

## 2.3 Θεωρητικά Μοντέλα Ανάκτησης Πληροφορίας

Η Ανάκτηση Πληροφορίας έχει αναπτυχθεί μέσα από διαφορετικά θεωρητικά μοντέλα, καθένα από τα οποία προσφέρει μια διαφορετική οπτική στην έννοια της συνάφειας. Κλασικά μοντέλα που χρησιμοποιούνται είναι:

**Boolean Model:** Πρόκειται για το απλούστερο μοντέλο IR, όπου τα έγγραφα είτε θεωρούνται σχετικά είτε όχι, με βάση λογικούς τελεστές (AND, OR, NOT). Αν και προσφέρει ακρίβεια στις αναζητήσεις, η δυαδική του φύση δεν λαμβάνει υπόψη τη βαθμίδα συνάφειας ενός εγγράφου, κάτι που περιορίζει την πρακτική του αξία σε μεγάλα σύνολα δεδομένων [8].

**Vector Space Model (VSM):** Αναπτύχθηκε τη δεκαετία του 1970 και θεωρείται θεμέλιος λίθος της IR. Στο VSM, έγγραφα και ερωτήματα αναπαρίστανται ως διανύσματα σε πολυδιάστατο χώρο, όπου κάθε διάσταση αντιστοιχεί σε έναν όρο. Η συνάφεια υπολογίζεται μέσω μέτρων ομοιότητας, όπως το cosine similarity. Σε αντίθεση με το Boolean Model, το VSM επιτρέπει την ιεράρχηση των αποτελεσμάτων, προσφέροντας πιο πρακτική προσέγγιση [11].

**Probabilistic Model:** Βασίζεται στην εκτίμηση της πιθανότητας ότι ένα έγγραφο είναι σχετικό με το ερώτημα του χρήστη. Ένα από τα πιο γνωστά μοντέλα είναι το Probabilistic Relevance Framework (PRF), το οποίο αποτέλεσε τη βάση για μελλοντικές εξελίξεις, όπως το BM25 το οποίο είναι ένα εξελιγμένο μοντέλο πιθανοκρατικής φύσης που βελτιώνει την ακρίβεια λαμβάνοντας υπόψη το μήκος των εγγράφων και τη συχνότητα όρων [13].

Επιπλέον, η μέθοδος TF-IDF (Term Frequency – Inverse Document Frequency) αποτελεί κεντρικό εργαλείο στην IR, καθώς σταθμίζει τους όρους με βάση τη συχνότητα εμφάνισής τους σε σχέση με όλη τη συλλογή, δίνοντας μεγαλύτερη σημασία σε διακριτικούς όρους [14].

Στη σύγχρονη εποχή, η IR έχει επηρεαστεί έντονα από τα νευρωνικά γλωσσικά μοντέλα. Τα Language Models for IR (LMIR) καθώς και τα BERT-based retrieval συστήματα [15] έχουν βελτιώσει δραματικά την απόδοση, αξιοποιώντας contextual embeddings. Εργαλεία όπως το Pyserini και το ElasticSearch αποτελούν δημοφιλείς πλατφόρμες εφαρμογής IR σε πραγματικά δεδομένα.

## 2.4 Εφαρμογές IR σε Πραγματικά Σενάρια

Η τεχνολογία IR εφαρμόζεται σε πληθώρα τομέων της καθημερινότητας:

- Μηχανές αναζήτησης (Google, Bing), όπου τα συστήματα πρέπει να επεξεργάζονται δισεκατομμύρια έγγραφα σε πραγματικό χρόνο.
- Ψηφιακές βιβλιοθήκες (Europeana, Digital Public Library of America), όπου η θεματική ευρετηρίαση επιτρέπει την καλύτερη οργάνωση επιστημονικού και πολιτιστικού περιεχομένου.
- Ηλεκτρονικό εμπόριο (Amazon, eBay), όπου η IR χρησιμοποιείται για την αντιστοίχιση προϊόντων με τα ερωτήματα των χρηστών.
- Κυβερνοασφάλεια, με εφαρμογές στην αναζήτηση και κατηγοριοποίηση ευπαθειών (π.χ. CVE → CWE mapping), όπως γίνεται και στην παρούσα εργασία.

Οι εφαρμογές αυτές δείχνουν ότι η IR δεν είναι απλώς ένα θεωρητικό πεδίο, αλλά μια κρίσιμη τεχνολογία που επηρεάζει άμεσα την καθημερινότητα, από την αναζήτηση πληροφοριών στο διαδίκτυο έως την ανάλυση δεδομένων κυβερνοασφάλειας.

## 2.5 Ευρετηρίαση (Indexing)

Η θεματική ευρετηρίαση είναι η διαδικασία απόδοσης όρων-κλειδιών ή θεματικών επικεφαλίδων σε ένα τεκμήριο, ώστε να αποτυπώνεται το περιεχόμενό του με σαφή και συστηματικό τρόπο. Οι όροι προέρχονται συχνά από ελεγχόμενα λεξιλόγια, διευκολύνοντας τη συνέπεια και την ακρίβεια στην αναπαράσταση της πληροφορίας [16].

Η κύρια σημασία της ευρετηρίασης έγκειται στη βελτίωση της αναζήτησης και ανάκτησης πληροφοριών, αφού καθιστά δυνατή τη σημασιολογική σύνδεση σχετικών τεκμηρίων ανεξαρτήτως γλωσσικών διαφορών [17]. Ιδιαίτερα σε ψηφιακά περιβάλλοντα, χρησιμοποιούνται αυτοματοποιημένα εργαλεία ευρετηρίασης βασισμένα σε τεχνικές μηχανικής μάθησης όπως το Annif, τα οποία αντιμετωπίζουν την πρόκληση της μεγάλης κλίμακας [18].

Η ανάθεση λέξεων-κλειδιών μπορεί να πραγματοποιηθεί είτε χειροκίνητα είτε αυτόματα μέσω υπολογιστικών συστημάτων. Η χειροκίνητη μέθοδος προσφέρει ακρίβεια και σημασιολογική συνέπεια, καθώς βασίζεται στην ανθρώπινη κατανόηση του περιεχομένου, αλλά είναι χρονοβόρα και δύσκολα επεκτάσιμη σε μεγάλες συλλογές [16].

Αντίθετα, η αυτόματη ανάθεση αξιοποιεί τεχνικές μηχανικής μάθησης και φυσικής γλώσσας, όπως συμβαίνει στο εργαλείο Annif, επιτυγχάνοντας ταχύτητα και οικονομία σε μεγάλης κλίμακας δεδομένα [19]. Ωστόσο, οι αυτόματες μέθοδοι ενδέχεται να παρουσιάσουν σφάλματα και χαμηλότερη ακρίβεια σε σύνθετα ή ασαφή κείμενα [18]. Για το λόγο αυτό, συχνά εφαρμόζεται συνδυαστική προσέγγιση, όπου η μηχανική ευρετηρίαση συνοδεύεται από ανθρώπινη εποπτεία για την επίτευξη ισορροπίας μεταξύ ταχύτητας και ποιότητας [17].

## 2.6 Οντολογίες

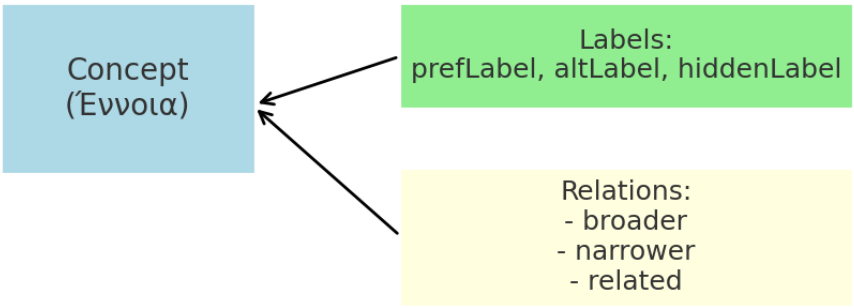
Στην Πληροφορική, οι οντολογίες αποτελούν δομημένες αναπαραστάσεις γνώσης, περιγράφοντας έννοιες (κλάσεις) και τις μεταξύ τους σχέσεις σε ένα συγκεκριμένο πεδίο. Συνιστούν θεμέλιο για την ανταλλαγή πληροφορίας μεταξύ ανθρώπων και μηχανών. Μια οντολογία περιλαμβάνει κλάσεις, ιδιότητες, σχέσεις ιεραρχίας και λεξιλογικά στοιχεία (όπως ετικέτες και ορισμούς). Υλοποιείται συνήθως με σημασιολογικά πρότυπα όπως RDF, OWL ή SKOS, τα οποία επιτρέπουν την επεξεργάσιμη και διαλειτουργική γνώση [20].

Διαφέρουν από τις ταξινομίες (taxonomies) και τους θησαυρούς (thesauri), καθώς προσφέρουν πιο πλούσια σημασιολογική πληροφόρηση. Το RDF (Resource Description Framework) και το OWL (Web Ontology Language) είναι τα βασικά πρότυπα του Σημασιολογικού Ιστού [21]. Το OWL διακρίνεται σε OWL-Lite, OWL-DL και OWL-Full, με διαφορετικά επίπεδα εκφραστικότητας. Στην ιατρική, η οντολογία SNOMED CT χρησιμοποιείται για την τυποποίηση ιατρικών εννοιών. Στην κυβερνοασφάλεια το MITRE ATT\&CK λειτουργεί ως γνώση-βάση τακτικών και τεχνικών επιθέσεων [22].

Καθώς η πληροφορία γίνεται ολοένα πιο περίπλοκη και διασυνδεδεμένη, οι οντολογίες προσφέρουν ένα ισχυρό εργαλείο σημασιολογικής κατανόησης και αυτοματισμού, ενισχύοντας την αποτελεσματικότητα των πληροφοριακών συστημάτων [23].

## 2.7 Το πρότυπο SKOS (Simple Knowledge Organization System)

Το SKOS αποτελεί ένα πρότυπο του W3C σχεδιασμένο για την αναπαράσταση λεξικογραφικών και θεματικών συστημάτων οργάνωσης της γνώσης όπως θησαυροί, ταξινομικά σχήματα, θεματικές επικεφαλίδες και ορολογίες. Εστιάζει στην απλουστευμένη αναπαράσταση εννοιών, χωρίς τις πολυπλοκότητες πιο φορμαλιστικών μοντέλων, όπως τα πλήρη οντολογικά σχήματα OWL [24]. Σε σύγκριση με το OWL το SKOS είναι πιο απλό και προσανατολισμένο στην οργάνωση εννοιών, ενώ το OWL προσφέρει αυστηρή σημασιολογική εκφραστικότητα.




*Σχήμα 1 Δομή SKOS με Έννοιες, Ετικέτες και Σχέσεις.*

Το SKOS βασίζεται στις εξής θεμελιώδεις δομές:

- Concepts (skos\:Concept): Αποτελούν τις βασικές θεματικές μονάδες. Κάθε έννοια είναι μοναδική και φέρει εννοιολογικό περιεχόμενο.
- Labels (Ετικέτες):
  skos\:prefLabel – Η προτιμώμενη λεκτική απόδοση μιας έννοιας (π.χ., "Υπολογιστές").
  skos\:altLabel – Εναλλακτικές λεκτικές αποδόσεις (π.χ., "Η/Υ").
  skos\:hiddenLabel – Απόκρυφες μορφές που χρησιμοποιούνται για αναζήτηση αλλά δεν εμφανίζονται.
- Relations (Σχέσεις):
  skos\:broader / narrower – Ιεραρχικές σχέσεις μεταξύ εννοιών.
  skos\:related – Συσχετισμένες έννοιες χωρίς ιεραρχική σχέση.
  skos\:definition, skos\:scopeNote – Περιγραφικά στοιχεία για την ερμηνεία και χρήση της έννοιας.

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:skos="http://www.w3.org/2004/02/skos/core#"
>
  <rdf:Description rdf:about="http://cwe.mitre.org/data/cwe#CWE-1037">
    <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
    <skos:prefLabel xml:lang="en">Processor Optimization Removal or Modification of Security-critical Code</skos:prefLabel>
    <skos:definition xml:lang="en">The developer builds a security-critical protection mechanism into the software, but the processor
    <skos:broader rdf:resource="http://cwe.mitre.org/data/cwe#CWE-1038"/>
```

*Σχήμα 2 Στιγμιότυπο από οντολογία SKOS*

Οι δομές του SKOS βασίζονται στο πρότυπο RDF, επιτρέποντας την αναπαράσταση, διασύνδεση και κοινή χρήση εννοιολογικών μοντέλων στο διαδίκτυο. Η χρήση του SKOS στην ευρετηρίαση και κατηγοριοποίηση πληροφοριακού περιεχομένου προσφέρει σημαντικά πλεονεκτήματα:

- Σημασιολογική συνοχή: Οι έννοιες εκπροσωπούνται με μοναδικά URI, αποφεύγοντας την ασάφεια της φυσικής γλώσσας [5].
- Πολλαπλή γλωσσική υποστήριξη: Μέσω `xml:lang` χαρακτηριστικών, το SKOS επιτρέπει την αναπαράσταση εννοιών σε πολλές γλώσσες, που είναι πολύ σημαντικό για πολυγλωσσικά συστήματα ευρετηρίασης.
- Διαλειτουργικότητα: Επιτρέπει την εύκολη ενσωμάτωση και συσχέτιση διαφορετικών θησαυρών και λεξιλογίων [25].
- Ευκολία ενσωμάτωσης με εργαλεία ευρετηρίασης: Πλατφόρμες όπως το Annif που υποστηρίζουν SKOS για αυτόματη ευρετηρίαση θεμάτων και λέξεων-κλειδιών.

- Ευκολία συντήρησης και επέκτασης: Η απλή δομή του SKOS επιτρέπει τη σταδιακή ανάπτυξη και τροποποίηση ενός λεξιλογίου χωρίς πλήρη αναδόμηση.

Το SKOS έχει χρησιμοποιηθεί σε πλήθος έργων. Το EuroVoc, το πολύγλωσσο λεξιλόγιο της ΕΕ, βασίζεται στο SKOS. Η DBpedia χρησιμοποιεί SKOS για την κατηγοριοποίηση δεδομένων της Wikipedia. Το AGROVOC (FAO) και το GEMET (EEA) είναι επίσης παραδείγματα εφαρμογής SKOS σε θεματικούς τομείς. Στην κυβερνοασφάλεια, SKOS οντολογίες μπορούν να χρησιμοποιηθούν για την τυποποίηση όρων σε CTI πλαίσια.

Καθώς η ανάγκη για σημασιολογικά οργανωμένη πληροφορία αυξάνεται, το SKOS προσφέρει μια ισορροπημένη λύση μεταξύ απλότητας και σημασιολογικής δύναμης για εφαρμογές θεματικής ευρετηρίασης, μεταδεδομένων και γνώσης.

## 2.8 Χρήση SKOS οντολογιών σε εφαρμογές πληροφορίας

Οι SKOS οντολογίες έχουν καθιερωθεί ως ευέλικτο και τυποποιημένο μέσο αναπαράστασης συστημάτων οργάνωσης γνώσης σε πληθώρα εφαρμογών πληροφορίας. Βασισμένες στο πρότυπο RDF και σχεδιασμένες για τη σημασιολογική μοντελοποίηση θεματικών λεξιλογίων, βρίσκουν εφαρμογή τόσο σε παραδοσιακά πληροφοριακά συστήματα (όπως βιβλιοθήκες), όσο και σε πιο σύγχρονες, σημασιολογικά ενισχυμένες εφαρμογές ιστού και συστήματα τεχνητής νοημοσύνης.

Στις ψηφιακές βιβλιοθήκες, οι SKOS οντολογίες χρησιμοποιούνται για την κατηγοριοποίηση και θεματική αναπαράσταση τεκμηρίων, αντικαθιστώντας ή εμπλουτίζοντας παραδοσιακά συστήματα επικεφαλίδων (όπως το LCSH). Εργαλεία όπως το Annif χρησιμοποιούν SKOS λεξιλόγια ως βάση για την αυτόματη ανάθεση θεμάτων σε βιβλιογραφικά αρχεία [19].

Το SKOS αποτελεί δομικό στοιχείο του Σημασιολογικού Ιστού (Semantic Web), καθώς επιτρέπει τη δημοσίευση και διασύνδεση σημασιολογικά περιγραφόμενων εννοιών. Έννοιες που αναπαρίστανται με URI μπορούν να συσχετιστούν με άλλες γλωσσικές ή θεματικές πηγές (π.χ. DBpedia, EuroVoc), ενισχύοντας τη διαλειτουργικότητα μεταξύ ανομοιογενών συστημάτων [26].

Με τη χρήση SKOS οντολογιών, η αναζήτηση πληροφοριών μπορεί να υποστηριχθεί από συστήματα επέκτασης ερωτημάτων (query expansion) ή σημασιολογική ευαισθησία, επιτρέποντας στον χρήστη να βρίσκει σχετικά τεκμήρια ακόμη και όταν χρησιμοποιεί διαφορετική φρασεολογία. Εφαρμογές IR ενσωματώνουν SKOS για να βελτιώσουν την ακρίβεια και πληρότητα στα αποτελέσματα [27].

Στην εκπαίδευση και στα ψηφιακά πολιτιστικά αποθετήρια, οι SKOS οντολογίες επιτρέπουν την πολυγλωσσική αναπαράσταση, την εννοιολογική πλοήγηση (concept-based navigation), αλλά και τη σύνδεση μαθησιακού περιεχομένου με συγκεκριμένα θεματικά πλαίσια [28].

Στον επιστημονικό και κυβερνητικό τομέα, SKOS χρησιμοποιείται για την ενοποίηση διαφορετικών ταξινομήσεων και τη διασύνδεση θεματικών πεδίων σε έργα ανοιχτών δεδομένων, εθνικών αρχείων, στατιστικών υπηρεσιών και θεματικών αποθετηρίων.

## 2.9 Επίλογος

Η Ανάκτηση και Οργάνωση Πληροφορίας αποτελεί θεμέλιο λίθο της σύγχρονης επιστήμης της Πληροφορικής, καθώς επιτρέπει την αποδοτική αξιοποίηση του συνεχώς αυξανόμενου όγκου δεδομένων. Από τα κλασικά μοντέλα IR, όπως το Boolean και το Vector Space Model, μέχρι τα πιο εξελιγμένα όπως το BM25 και τα νευρωνικά γλωσσικά μοντέλα, παρατηρείται μια σταδιακή μετάβαση προς μεθόδους που ενσωματώνουν περισσότερη σημασιολογική κατανόηση και προσαρμοστικότητα στις ανάγκες του χρήστη.

Παράλληλα, η θεματική ευρετηρίαση και η χρήση ελεγχόμενων λεξιλογίων αναδεικνύουν τη σημασία της συστηματικής οργάνωσης της πληροφορίας. Σε αυτό το πλαίσιο, οι οντολογίες λειτουργούν ως εννοιολογικά μοντέλα που γεφυρώνουν τη γλώσσα των ανθρώπων με τις απαιτήσεις των υπολογιστικών συστημάτων, επιτρέποντας την αποδοτική αναπαράσταση και διασύνδεση γνώσης.

Το SKOS προσφέρει μια ισορροπημένη προσέγγιση ανάμεσα στην απλότητα και τη σημασιολογική δύναμη, καθιστώντας το ιδιαίτερα χρήσιμο για εφαρμογές θεματικής ευρετηρίασης και σημασιολογικού ιστού. Οι εφαρμογές του σε βιβλιοθήκες, επιστημονικά αποθετήρια, αλλά και σε συστήματα τεχνητής νοημοσύνης αποδεικνύουν την πρακτική του αξία.

Συνολικά, το κεφάλαιο αυτό αναδεικνύει πως η συνδυαστική χρήση μοντέλων IR, οντολογιών και προτύπων όπως το SKOS ενισχύει τη δυνατότητα εντοπισμού και κατηγοριοποίησης πληροφοριών σε μεγάλη κλίμακα. Η κατανόηση αυτών των εννοιών αποτελεί κρίσιμο υπόβαθρο για την παρούσα εργασία, η οποία αξιοποιεί το εργαλείο Annif και SKOS οντολογίες για την αυτόματη θεματική κατηγοριοποίηση ευπαθειών στον τομέα της Κυβερνοασφάλειας.

# Κεφάλαιο 3ο: Κυβερνοασφάλεια και Ευπάθειες

## 3.1 Εισαγωγή

Η κυβερνοασφάλεια (cybersecurity) αναφέρεται στη συλλογή τεχνολογιών, διαδικασιών και πρακτικών που έχουν ως στόχο την προστασία συστημάτων πληροφορικής, δικτύων και δεδομένων από μη εξουσιοδοτημένη πρόσβαση, κακόβουλες επιθέσεις και καταστροφές. Καθώς η ψηφιοποίηση επεκτείνεται σε κάθε πτυχή της κοινωνικής και οικονομικής ζωής, η κυβερνοασφάλεια αποτελεί κρίσιμο παράγοντα για τη διασφάλιση της εμπιστευτικότητας, της ακεραιότητας και της διαθεσιμότητας των πληροφοριακών πόρων [29].

Η ραγδαία αύξηση των σύνθετων απειλών – όπως ransomware, phishing, επιθέσεις τύπου zero-day και Advanced Persistent Threats (APT) – έχει ενισχύσει την ανάγκη για προληπτικά και ανιχνευτικά μέτρα ασφάλειας. Παράλληλα, η κυβερνοασφάλεια επεκτείνεται σε τομείς όπως η ασφάλεια κρίσιμων υποδομών, η προστασία προσωπικών δεδομένων και η ψηφιακή εμπιστοσύνη, καθιστώντας την διεπιστημονική και διαρκώς εξελισσόμενη.

## 3.2 Θεμελιώδεις Αρχές της Κυβερνοασφάλειας

Η θεωρητική βάση της Κυβερνοασφάλειας στηρίζεται στο γνωστό τρίπτυχο CIA Triad (Confidentiality, Integrity, Availability), το οποίο καθορίζει τις βασικές αρχές προστασίας των πληροφοριακών συστημάτων [30].


*Σχήμα 3 Το μοντέλο CIA Triad*

Η **Εμπιστευτικότητα** (Confidentiality) αναφέρεται στην προστασία των δεδομένων από μη εξουσιοδοτημένη πρόσβαση. Ο στόχος είναι να διασφαλίζεται ότι μόνο άτομα ή συστήματα με τα κατάλληλα δικαιώματα μπορούν να έχουν πρόσβαση σε ευαίσθητες πληροφορίες. Χαρακτηριστικά παραδείγματα παραβίασης της εμπιστευτικότητας είναι οι διαρροές προσωπικών δεδομένων ή η υποκλοπή διαπιστευτηρίων πρόσβασης.

Η **Ακεραιότητα** (Integrity) αφορά τη διασφάλιση ότι τα δεδομένα παραμένουν ακριβή και αμετάβλητα, εκτός εάν τροποποιούνται με εξουσιοδοτημένο τρόπο. Η παραβίαση της ακεραιότητας μπορεί να έχει σοβαρές συνέπειες, όπως στην περίπτωση αλλοίωσης ιατρικών αρχείων ή παραποίησης οικονομικών δεδομένων.

Η **Διαθεσιμότητα** (Availability) εστιάζει στο να είναι οι πληροφορίες και οι υπηρεσίες συνεχώς προσβάσιμες στους νόμιμους χρήστες τους. Επιθέσεις όπως τα Distributed Denial of Service (DDoS) απειλούν άμεσα αυτή την αρχή, εμποδίζοντας την ομαλή λειτουργία κρίσιμων συστημάτων.

## 3.3 Επέκταση στο CIA Triad και Νέες Διαστάσεις Ασφάλειας

Το τρίπτυχο Confidentiality, Integrity, Availability αποτελεί τη θεμελιώδη αρχή της κυβερνοασφάλειας. Ωστόσο, η εξέλιξη των απειλών οδήγησε στην ανάδειξη νέων διαστάσεων που συμπληρώνουν το παραδοσιακό μοντέλο:

Η **Γνησιότητα** (Authenticity): Διασφαλίζει ότι τα δεδομένα ή τα συστήματα προέρχονται από έγκυρη πηγή και δεν έχουν αλλοιωθεί. Είναι ιδιαίτερα κρίσιμη σε περιβάλλοντα ηλεκτρονικών συναλλαγών και λογισμικού αλυσίδας εφοδιασμού.

Η **Μη-αποποίηση** (Non-repudiation): Εξασφαλίζει ότι ο αποστολέας μιας ενέργειας (π.χ. email, συναλλαγή) δεν μπορεί να αρνηθεί τη συμμετοχή του. Αυτό υλοποιείται συνήθως μέσω κρυπτογραφικών μηχανισμών, όπως οι ψηφιακές υπογραφές και η αποστολή ενός ηλεκτρονικού μηνύματος [31].

Η **Ανθεκτικότητα** (Resilience): Αναφέρεται στη δυνατότητα ενός οργανισμού να συνεχίζει τη λειτουργία του ακόμη και μετά από κυβερνοεπίθεση, μέσω μηχανισμών ανάκαμψης και εφεδρικών συστημάτων.

Οι επεκτάσεις αυτές καθιστούν σαφές ότι η κυβερνοασφάλεια δεν είναι μόνο τεχνικό ζήτημα, αλλά και θέμα εμπιστοσύνης, νομιμότητας και επιχειρησιακής συνέχειας [32].

## 3.4 Ανάλυση Σύγχρονων Απειλών

Οι σύγχρονες κυβερνοεπιθέσεις χαρακτηρίζονται από ολοένα και μεγαλύτερη πολυπλοκότητα, συνδυάζοντας τεχνικές αδυναμίες συστημάτων με την εκμετάλλευση του ανθρώπινου παράγοντα. Δεν περιορίζονται πλέον σε μεμονωμένα περιστατικά, αλλά εξελίσσονται σε στρατηγικά οργανωμένες δράσεις που αποσκοπούν σε οικονομικό, πολιτικό ή και γεωπολιτικό όφελος.

Μια ιδιαίτερα σοβαρή κατηγορία αποτελούν οι Advanced Persistent Threats (APT). Πρόκειται για μακροχρόνιες και στοχευμένες επιθέσεις που συνήθως υποστηρίζονται από κρατικούς ή οργανωμένους φορείς. Οι APT δεν επιδιώκουν μια γρήγορη και θορυβώδη παραβίαση, αλλά παραμένουν κρυφές στα συστήματα για μεγάλα χρονικά διαστήματα, συλλέγοντας πληροφορίες και παρακολουθώντας κρίσιμες υποδομές. Ενδεικτικά, στόχοι τέτοιων επιθέσεων μπορεί να είναι κυβερνητικά δίκτυα, ενεργειακές υποδομές ή μεγάλοι οργανισμοί στρατηγικής σημασίας [33].

Ιδιαίτερα επικίνδυνες είναι επίσης οι Zero-day Exploits, δηλαδή επιθέσεις που εκμεταλλεύονται αδυναμίες λογισμικού για τις οποίες δεν έχει κυκλοφορήσει ακόμη ενημέρωση (patch). Το γεγονός ότι αυτές οι ευπάθειες δεν είναι γνωστές στους προγραμματιστές και στους διαχειριστές συστημάτων καθιστά τους οργανισμούς ευάλωτους, καθώς δεν υπάρχουν διαθέσιμα μέτρα προστασίας. Συχνά, οι επιτιθέμενοι διακινούν τέτοια exploits στη μαύρη αγορά με υψηλό κόστος, γεγονός που αναδεικνύει τη σοβαρότητά τους.

Τα τελευταία χρόνια έχει αναπτυχθεί ένα νέο επιχειρηματικό μοντέλο στο χώρο του κυβερνοεγκλήματος, γνωστό ως Ransomware-as-a-Service (RaaS). Σε αυτό, οργανωμένες ομάδες κυβερνοεγκληματιών αναπτύσσουν πλατφόρμες ransomware και τις παρέχουν σε τρίτους έναντι αμοιβής ή μερίσματος από τα κέρδη. Με αυτόν τον τρόπο, ακόμη και άτομα με περιορισμένες τεχνικές γνώσεις μπορούν να εξαπολύσουν επιθέσεις ransomware, αυξάνοντας θεαματικά τη συχνότητα τέτοιων περιστατικών. Οι συνέπειες είναι ιδιαίτερα σοβαρές, καθώς οι οργανισμοί έρχονται αντιμέτωποι με κρυπτογράφηση κρίσιμων δεδομένων και αιτήματα για καταβολή λύτρων.

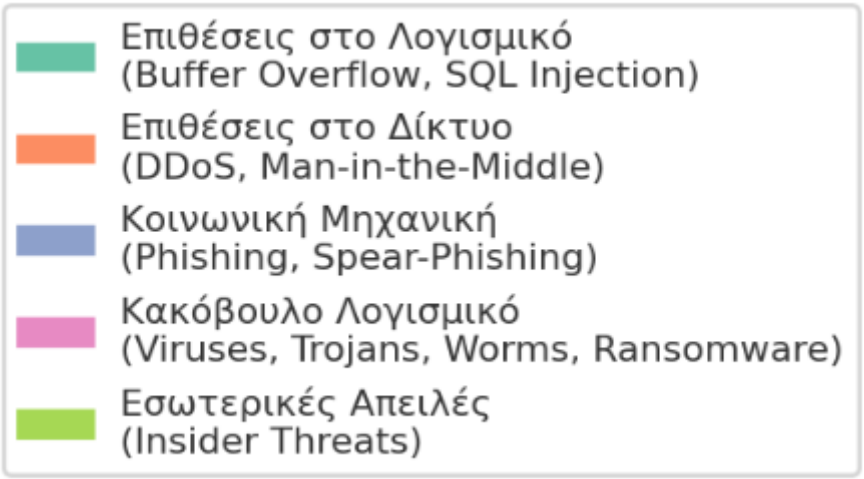
Τέλος, σημαντική απειλή αποτελούν οι Supply Chain Attacks, δηλαδή οι επιθέσεις που στοχεύουν όχι τον τελικό οργανισμό, αλλά τους προμηθευτές ή συνεργάτες του. Μέσω αυτής της στρατηγικής, οι

επιτιθέμενοι εισάγουν κακόβουλο κώδικα σε λογισμικό ή υπηρεσίες που χρησιμοποιεί ο οργανισμός-στόχος, αποκτώντας πρόσβαση με έμμεσο τρόπο. Το πιο γνωστό παράδειγμα είναι η επίθεση SolarWinds το 2020, κατά την οποία μολύνθηκε το λογισμικό Orion, οδηγώντας σε παραβίαση χιλιάδων οργανισμών, συμπεριλαμβανομένων και κρατικών φορέων των ΗΠΑ.

Οι παραπάνω μορφές απειλών αναδεικνύουν τη δυναμική και πολυδιάστατη φύση της κυβερνοασφάλειας. Οι οργανισμοί δεν μπορούν να αρκεστούν σε παραδοσιακά μέτρα άμυνας, αλλά απαιτείται συνεχής προσαρμογή στρατηγικών, συνδυασμός τεχνολογικών εργαλείων, εκπαίδευση προσωπικού και διεθνής συνεργασία, ώστε να αντιμετωπίζονται αποτελεσματικά οι εξελισσόμενες επιθέσεις.

### 3.5 Κατηγορίες Απειλών και Επιθέσεων

Οι απειλές στον κυβερνοχώρο καλύπτουν ένα ευρύ φάσμα επιθέσεων, οι οποίες μπορούν να στοχεύσουν διαφορετικά επίπεδα ενός πληροφοριακού συστήματος.




*Σχήμα 4 Κατηγορίες Κυβερνοαπειλών*

Οι επιθέσεις στο λογισμικό αφορούν την εκμετάλλευση σφαλμάτων και ευπαθειών σε εφαρμογές, όπως οι επιθέσεις Buffer Overflow και SQL Injection, που επιτρέπουν σε επιτιθέμενους να εκτελέσουν κακόβουλο κώδικα ή να αποκτήσουν πρόσβαση σε βάσεις δεδομένων [34].

Οι επιθέσεις στο δίκτυο περιλαμβάνουν πρακτικές όπως τα DDoS, τα οποία υπερφορτώνουν τους διακομιστές ώστε να καταστούν μη διαθέσιμοι, ή τις επιθέσεις man-in-the-middle, όπου ο εισβολέας παρεμβάλλεται μεταξύ δύο επικοινωνούντων μερών για να υποκλέψει ή να τροποποιήσει τα δεδομένα.

Η κοινωνική μηχανική (Social Engineering) εκμεταλλεύεται τον ανθρώπινο παράγοντα, χρησιμοποιώντας μεθόδους εξαπάτησης όπως το Phishing ή το Spear-Phishing. Σε αυτές τις περιπτώσεις, οι χρήστες παραπλανούνται ώστε να αποκαλύψουν προσωπικά δεδομένα ή να εγκαταστήσουν κακόβουλο λογισμικό.

Οι επιθέσεις με κακόβουλο λογισμικό (malware) περιλαμβάνουν ιούς, trojans, worms και ransomware, που έχουν ως στόχο την παραβίαση συστημάτων, την κλοπή δεδομένων ή την απαίτηση λύτρων για την αποκατάσταση της πρόσβασης.

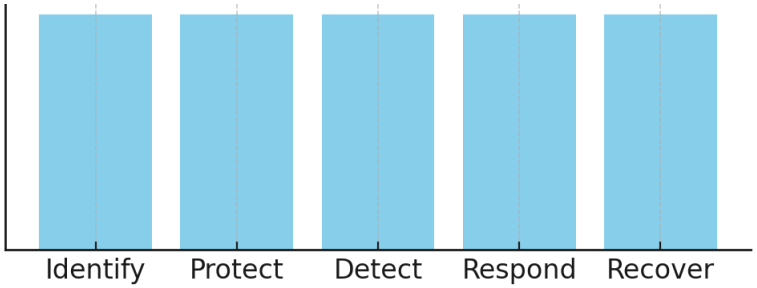
Τέλος, οι εσωτερικές απειλές (insider threats) προέρχονται από άτομα που έχουν νόμιμη πρόσβαση σε συστήματα και δεδομένα, αλλά εκμεταλλεύονται αυτή την πρόσβαση για κακόβουλους ή ακούσιους

σκοπούς [1]. Η κατηγοριοποίηση των απειλών είναι κρίσιμη, καθώς επιτρέπει στους οργανισμούς να αναπτύξουν στοχευμένες στρατηγικές πρόληψης και αντιμετώπισης ανάλογα με τον τύπο της επίθεσης.

### 3.6 Πρότυπα, Πλαίσια και Οργανισμοί

Η Κυβερνοασφάλεια δεν περιορίζεται μόνο σε τεχνικές λύσεις, αλλά στηρίζεται και σε διεθνώς αναγνωρισμένα πρότυπα και πλαίσια που καθοδηγούν τις πρακτικές προστασίας.

Το Cybersecurity Framework (CSF) είναι ένα από τα πιο διαδεδομένα πλαίσια, που έχει αναπτυχθεί από το National Institute of Standards and Technology (NIST). Περιλαμβάνει πέντε βασικές λειτουργίες: Identify, Protect, Detect, Respond, Recover, οι οποίες παρέχουν έναν ολιστικό τρόπο προσέγγισης της ασφάλειας πληροφοριών και χρησιμοποιούνται εκτεταμένα από οργανισμούς σε όλο τον κόσμο [35].




*Σχήμα 5 NIST Cybersecurity Framework (CSF)*

Το ISO/IEC 27001 είναι το διεθνές πρότυπο που καθορίζει τις απαιτήσεις για τη δημιουργία, εφαρμογή και διατήρηση ενός Συστήματος Διαχείρισης Ασφάλειας Πληροφοριών (ISMS). Το πρότυπο αυτό δίνει έμφαση στη διαχείριση κινδύνων και στην εφαρμογή πολιτικών που διασφαλίζουν την ασφάλεια των δεδομένων [30].

Η ENISA (European Union Agency for Cybersecurity) αποτελεί τον κεντρικό οργανισμό της Ευρωπαϊκής Ένωσης για θέματα ασφάλειας στον κυβερνοχώρο. Ο ρόλος της περιλαμβάνει την παροχή κατευθυντήριων γραμμών, την εκπόνηση μελετών και την ενίσχυση της συνεργασίας μεταξύ των κρατών-μελών για την ενίσχυση της ανθεκτικότητας έναντι κυβερνοαπειλών [1].

Η υιοθέτηση τέτοιων προτύπων και πλαισίων ενισχύει τη δυνατότητα των οργανισμών να αντιμετωπίζουν αποτελεσματικά τις απειλές και να συμμορφώνονται με νομικές και κανονιστικές απαιτήσεις.

### 3.7 Ευπάθειες Ασφαλείας

Οι ευπάθειες (vulnerabilities) είναι αδυναμίες σε συστήματα ή λογισμικό που μπορούν να αξιοποιηθούν από επιτιθέμενους για να υπονομεύσουν την ασφάλεια [34]. Στην πράξη, οι ευπάθειες καταγράφονται και τυποποιούνται μέσω διεθνών βάσεων δεδομένων, οι σημαντικότερες από τις οποίες είναι:

- CVE (Common Vulnerabilities and Exposures) τυποποιημένη λίστα γνωστών ευπαθειών.
- CWE (Common Weakness Enumeration) ταξινόμηση τύπων ευπαθειών λογισμικού.
- NVD (National Vulnerability Database) η επίσημη βάση δεδομένων του NIST.

Οι ευπάθειες μπορεί να είναι τεχνικές (π.χ., buffer overflows, improper input validation), σχεδιαστικές (π.χ., incomplete access control), ή οργανωσιακές (π.χ., inadequate staff training) [36]. Η συστηματική κατηγοριοποίηση των ευπαθειών είναι απαραίτητη για την κατανόηση των κινδύνων, την ιεράρχηση προτεραιοτήτων αποκατάστασης και την ενίσχυση της πρόληψης. Ταξινομητικά σχήματα όπως το CWE και πρότυπα όπως το CVSS (Common Vulnerability Scoring System) συμβάλλουν στην αξιολόγηση και περιγραφή των ευπαθειών με τρόπο επαναχρησιμοποιήσιμο και αναλυτικό [37].

Η Βάση Δεδομένων CVE είναι ένα διεθνώς αναγνωρισμένο πρότυπο που προσφέρει μοναδικούς αναγνωριστικούς κωδικούς για δημόσια γνωστές ευπάθειες σε πληροφοριακά συστήματα. Δημιουργήθηκε από το MITRE το 1999 και λειτουργεί υπό την αιγίδα του U.S. Department of Homeland Security μέσω του προγράμματος CVE Numbering Authorities (CNAs).

Κάθε CVE εγγραφή περιλαμβάνει:

- CVE ID (π.χ., CVE-2023-12345)
- περιγραφή της ευπάθειας
- ημερομηνία καταχώρησης
- αναφορές σε πηγές ή διορθώσεις

Η βάση δεδομένων CVE αποτελεί τη θεμέλια πηγή για πλήθος άλλων εργαλείων και πλαισίων, όπως το NVD, το CVSS και οι βάσεις ευπαθειών εταιρειών ασφαλείας. Παίζει καίριο ρόλο στην τυποποίηση και διαλειτουργικότητα των συστημάτων διαχείρισης ευπαθειών και απειλών.

Η κατηγοριοποίηση των ευπαθειών έχει προσεγγιστεί μέσα από δομημένα λεξιλόγια, συστήματα αναγνώρισης μοτίβων, αλλά και τεχνικές μηχανικής μάθησης. Το CWE είναι ένα από τα πιο διαδεδομένα ταξινομητικά σχήματα, που παρέχει μια ιεραρχική δομή αδυναμιών λογισμικού σε μορφή οντολογίας, διευκολύνοντας την ενιαία περιγραφή ευπαθειών [37].

Στο πεδίο της αυτόματης κατηγοριοποίησης, έχουν εφαρμοστεί τεχνικές εξόρυξης κειμένου (text mining) και επεξεργασίας φυσικής γλώσσας (NLP) για την αντιστοίχιση CVE περιγραφών με CWE έννοιες. Εργαλεία μηχανικής μάθησης, όπως ταξινομητές SVM, Random Forests ή μοντέλα νευρωνικών δικτύων, αξιοποιούνται για την ημιαυτόματη αντιστοίχιση ευπαθειών με κατηγορίες αδυναμιών, επιτυγχάνοντας ακρίβεια και επεκτασιμότητα [38].

## 3.8 Επίλογος

Η ανάλυση των βασικών αρχών, απειλών, προτύπων και ευπαθειών της Κυβερνοασφάλειας καταδεικνύει τον πολυδιάστατο χαρακτήρα του πεδίου. Το τρίπτυχο CIA Triad παραμένει θεμελιώδες για την κατανόηση της προστασίας πληροφοριακών συστημάτων, ενώ η επέκτασή του με έννοιες όπως η Αυθεντικότητα και η μη-Αποποίηση αντανακλά τις σύγχρονες ανάγκες για ενισχυμένη εμπιστοσύνη στις ψηφιακές συναλλαγές.

Οι κατηγορίες επιθέσεων – από τις τεχνικές ευπάθειες λογισμικού έως την κοινωνική μηχανική και τις εσωτερικές απειλές – αποδεικνύουν ότι η κυβερνοασφάλεια δεν είναι μόνο τεχνολογικό, αλλά και κοινωνικό και οργανωτικό ζήτημα. Παράλληλα, η ύπαρξη διεθνών πλαισίων και προτύπων, όπως το NIST CSF, το ISO/IEC 27001 και οι κατευθυντήριες γραμμές της ENISA, παρέχουν κοινό σημείο αναφοράς για την ανάπτυξη στρατηγικών ασφάλειας με διεθνή αναγνώριση και εφαρμογή.

Σε αυτό το πλαίσιο, οι ευπάθειες αποτελούν τον κεντρικό σύνδεσμο ανάμεσα στη θεωρία και την πράξη. Η συστηματική καταγραφή και κατηγοριοποίησή τους μέσω των CVE, CWE και NVD, σε συνδυασμό με πρότυπα όπως το CVSS, δημιουργούν τη βάση για την επιστημονική μελέτη και την επιχειρησιακή διαχείριση των κινδύνων. Η αξιοποίηση τεχνικών μηχανικής μάθησης και επεξεργασίας φυσικής γλώσσας για την αυτόματη κατηγοριοποίηση ευπαθειών αναδεικνύει τον εξελισσόμενο ρόλο της τεχνητής νοημοσύνης στην Κυβερνοασφάλεια.

Συνολικά, το κεφάλαιο αυτό υπογραμμίζει ότι η κατανόηση των θεμελιωδών αρχών και προκλήσεων της Κυβερνοασφάλειας είναι απαραίτητη προϋπόθεση για την ανάπτυξη καινοτόμων λύσεων, όπως αυτή που προτείνεται στην παρούσα εργασία, με στόχο την αυτόματη θεματική κατηγοριοποίηση ευπαθειών στον τομέα της Κυβερνοασφάλειας.

# Κεφάλαιο 4ο: Μηχανική Μάθηση και ανάλυση κειμένων

## 4.1 Εισαγωγή

Η Μηχανική Μάθηση (Machine Learning – ML) και η Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing – NLP) αποτελούν δύο αλληλένδετους τομείς της Τεχνητής Νοημοσύνης που έχουν συμβάλει σημαντικά στην αυτόματη ανάλυση και κατανόηση κειμένων. Η ML προσφέρει μεθόδους εκπαίδευσης αλγορίθμων ώστε να μαθαίνουν από δεδομένα και να πραγματοποιούν προβλέψεις ή ταξινομήσεις, ενώ η NLP επικεντρώνεται στην κατανόηση, ανάλυση και παραγωγή φυσικής γλώσσας [8].

Οι κυριότερες προσεγγίσεις ML για ανάλυση κειμένων περιλαμβάνουν μεθόδους επιβλεπόμενης μάθησης (supervised learning), όπως οι Naïve Bayes, Support Vector Machines (SVMs) και νευρωνικά δίκτυα [39]. Ειδικά στον χώρο της ανάκτησης πληροφορίας και της θεματικής κατηγοριοποίησης, οι SVMs και τα δέντρα απόφασης έχουν αποδειχθεί ιδιαίτερα αποτελεσματικά. Επιπλέον, τα νεότερα μοντέλα βαθιάς μάθησης (deep learning), όπως τα LSTMs και τα Transformers, έχουν βελτιώσει σημαντικά την απόδοση σε εργασίες κατανόησης κειμένου [40].

Η προ επεξεργασία των κειμένων αποτελεί βασικό στάδιο στις NLP εφαρμογές. Η κατάτμηση κειμένου (tokenization) διαχωρίζει τις προτάσεις σε λέξεις ή φράσεις, οι οποίες αποτελούν τις βασικές μονάδες ανάλυσης. Στη συνέχεια, το stemming ή το lemmatization μειώνει τις λέξεις στη ρίζα τους, ώστε να περιοριστεί η ποικιλία των μορφολογικών τύπων. Παράλληλα, η αφαίρεση των stop words (συχνών λέξεων όπως 'και', 'το', 'είναι') μειώνει τον θόρυβο και βελτιώνει την ποιότητα των χαρακτηριστικών που χρησιμοποιούνται στους αλγόριθμους [8].

Η συνδυασμένη χρήση ML και NLP είναι ιδιαίτερα σημαντική για την αυτόματη εξαγωγή θεματικών λέξεων-κλειδιών (automatic keyword extraction). Μέσω τεχνικών όπως TF-IDF, TextRank και αλγορίθμων βαθιάς μάθησης, είναι εφικτή η αναγνώριση των πιο σημαντικών εννοιών ενός κειμένου. Στον τομέα της κυβερνοασφάλειας, η εξαγωγή λέξεων-κλειδιών αποκτά ιδιαίτερη σημασία, καθώς επιτρέπει την αυτόματη κατηγοριοποίηση ευπαθειών, επιθέσεων και απειλών, διευκολύνοντας την ανάλυση και ανταλλαγή γνώσης [41].

## 4.2 Αυτόματη Θεματική Ευρετηρίαση

Η αυτόματη θεματική ευρετηρίαση (Automatic Subject Indexing – ASI) αναφέρεται στη διαδικασία απόδοσης θεματικών όρων ή ετικετών (subject terms, keywords) σε έγγραφα, με στόχο τη διευκόλυνση της αναζήτησης και της ανάκτησης πληροφορίας [42]. Σε αντίθεση με τη χειροκίνητη θεματική ευρετηρίαση, η οποία είναι ιδιαίτερα χρονοβόρα και απαιτεί εξειδικευμένη γνώση, η αυτόματη προσέγγιση μειώνει το κόστος, επιταχύνει τη διαδικασία και ενισχύει τη συνέπεια στην απόδοση θεματικών όρων.

Οι πρώτες μέθοδοι ASI βασίστηκαν σε στατιστικές τεχνικές επεξεργασίας κειμένου. Το TF-IDF αποτέλεσε μια από τις πλέον διαδεδομένες μεθόδους, καθώς προσδιορίζει τη σχετική σημασία μιας λέξης σε ένα κείμενο σε σχέση με ένα σύνολο εγγράφων [8]. Στη συνέχεια, εισήχθησαν πιο σύνθετες τεχνικές όπως η Latent Semantic Analysis (LSA), η οποία βασίζεται στη μείωση διαστάσεων μέσω SVD και αναδεικνύει λανθάνουσες εννοιολογικές σχέσεις. Ακόμη, η Latent Dirichlet Allocation (LDA) επέτρεψε τη μοντελοποίηση θεμάτων με πιθανοτικό τρόπο, επιτρέποντας την ανάθεση θεμάτων σε κείμενα με βάση κατανομές πιθανοτήτων.

Τα τελευταία χρόνια, η ανάπτυξη της βαθιάς μάθησης (Deep Learning) έχει φέρει σημαντικές αλλαγές στον τομέα της αυτόματης ευρετηρίασης. Νευρωνικά δίκτυα, όπως τα Convolutional Neural Networks

(CNNs) και Recurrent Neural Networks (RNNs), έχουν χρησιμοποιηθεί με επιτυχία για την κατηγοριοποίηση κειμένων και την εξαγωγή θεματικών όρων [43]. Ακόμη πιο πρόσφατα, τα μοντέλα μετασχηματιστών (Transformer models), όπως το BERT [44] και το RoBERTa [45], έχουν αποδειχθεί εξαιρετικά αποτελεσματικά, καθώς αξιοποιούν μηχανισμούς αυτο-προσοχής (self-attention) για να κατανοήσουν συμφραζόμενα λέξεων και προτάσεων με υψηλή ακρίβεια. Η χρήση τους στην αυτόματη θεματική ευρετηρίαση προσφέρει πλέον καλύτερα αποτελέσματα σε ποικίλα σύνολα δεδομένων.

Η ανάγκη για αυτόματη θεματική ευρετηρίαση είναι ιδιαίτερα έντονη σε πεδία όπου ο όγκος των δεδομένων είναι τεράστιος, όπως στις βιβλιοθήκες, τα ψηφιακά αποθετήρια, αλλά και στον τομέα της κυβερνοασφάλειας, όπου απαιτείται η γρήγορη και αξιόπιστη κατηγοριοποίηση πληροφοριών για απειλές και ευπάθειες [46]. Η ASI αποτελεί πλέον βασικό συστατικό της σύγχρονης ανάκτησης πληροφορίας και συνεχίζει να εξελίσσεται με την πρόοδο της τεχνητής νοημοσύνης.

## 4.3 Κατηγορίες Μηχανικής Μάθησης

Η Μηχανική Μάθηση (Machine Learning – ML) αποτελεί έναν από τους βασικότερους κλάδους της Τεχνητής Νοημοσύνης και επικεντρώνεται στην ανάπτυξη αλγορίθμων που επιτρέπουν στα υπολογιστικά συστήματα να βελτιώνουν την απόδοσή τους μέσα από την εμπειρία.

Η μηχανική μάθηση διακρίνεται σε τρεις βασικές κατηγορίες: την επιβλεπόμενη, τη μη επιβλεπόμενη και την ενισχυτική μάθηση [47]. Στην επιβλεπόμενη μάθηση, το σύστημα εκπαιδεύεται με δεδομένα που συνοδεύονται από τις σωστές ετικέτες. Στόχος είναι να μάθει μια συνάρτηση που, για κάθε νέα είσοδο x, θα προβλέπει σωστά την έξοδο y. Στη μη επιβλεπόμενη μάθηση, τα δεδομένα δεν διαθέτουν ετικέτες, και το ζητούμενο είναι να ανακαλυφθούν κρυμμένες δομές ή πρότυπα μέσα σε αυτά. Τέλος, στην ενισχυτική μάθηση, το σύστημα δεν μαθαίνει από έτοιμα δεδομένα αλλά από την αλληλεπίδρασή του με το περιβάλλον: ο πράκτορας δοκιμάζει ενέργειες, λαμβάνει ανταμοιβές ή τιμωρίες, και σταδιακά μαθαίνει πολιτικές που μεγιστοποιούν τη συνολική ανταμοιβή του.

Μεταξύ των βασικών μοντέλων επιβλεπόμενης μάθησης περιλαμβάνονται η Λογιστική Παλινδρόμηση, ο Naïve Bayes, οι Μηχανές Υποστήριξης Διανυσμάτων (SVMs) και τα Νευρωνικά Δίκτυα.

Η Λογιστική Παλινδρόμηση χρησιμοποιείται εκτενώς για δυαδική ταξινόμηση και μοντελοποιεί την πιθανότητα να ανήκει ένα δείγμα σε μια κατηγορία. Τα λογιστικά μοντέλα εφαρμόζονται σε περιπτώσεις όπου η εξαρτημένη μεταβλητή είναι δυαδική. Η λογιστική παλινδρόμηση ανήκει στην κατηγορία των μη γραμμικών μεθόδων, καθώς οι προβλεπόμενες τιμές της περιορίζονται στο διάστημα μεταξύ 0 και 1.

Στόχος των μοντέλων αυτών είναι να εκτιμήσουν την πιθανότητα η εξαρτημένη μεταβλητή να λάβει την τιμή 1 (Y=1), δηλαδή την πιθανότητα να συμβεί ένα συγκεκριμένο γεγονός. Η Λογιστική Παλινδρόμηση χρησιμοποιείται εκτενώς για δυαδική ταξινόμηση και μοντελοποιεί την πιθανότητα να ανήκει ένα δείγμα σε μια κατηγορία μέσω της σιγμοειδούς συνάρτησης:

$$P(y = 1|x) = \frac{1}{1 + e^{-(w^T x + b)}}$$

*Σχήμα 6 Σιγμοειδής συνάρτηση στη Λογιστική Παλινδρόμηση*

Η εκπαίδευση βασίζεται στην ελαχιστοποίηση της συνάρτησης κόστους cross-entropy:

$$L = -\sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

*Σχήμα 7 Συνάρτηση κόστους cross-entropy*

Οι SVM επιλέγουν το υπερεπίπεδο που μεγιστοποιεί το περιθώριο μεταξύ κλάσεων. Η εκπαίδευση βασίζεται στη χρήση της hinge loss:

$$L = \sum_{i=1}^{N} \max(0, 1 - y_i(w^T x_i + b))$$

*Σχήμα 8 Hinge loss στη Μηχανή Υποστήριξης Διανυσμάτων*

Ο Naïve Bayes είναι ένας απλός αλλά ισχυρός ταξινομητής που στηρίζεται στο θεώρημα του Bayes με την υπόθεση ανεξαρτησίας των χαρακτηριστικών:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

*Σχήμα 9 Το θεώρημα του Bayes*

Στην πολυετικετική ταξινόμηση κάθε παράδειγμα μπορεί να ανήκει σε πολλές ετικέτες. Τεχνικές μετασχηματισμού περιλαμβάνουν τη Binary Relevance (BR), τις Αλυσίδες Ταξινομητών (Classifier Chains – CC) και την τεχνική Label Powerset (LP). Παράλληλα, προσαρμοσμένοι αλγόριθμοι ενσωματώνουν τη multi-label πρόβλεψη στο ίδιο το μοντέλο, όπως πολυετικετικά δέντρα ή νευρωνικά δίκτυα με πολλαπλές εξόδους. Η αξιολόγηση γίνεται με μετρικές όπως Precision, Recall και F1-score, αλλά και με μετρικές κατάταξης όπως MAP και nDCG, οι οποίες δίνουν έμφαση στις κορυφαίες προβλέψεις.

## 4.4 Αλγόριθμοι Μηχανικής Μάθησης στο Annif

Το Annif είναι ένα open-source εργαλείο για αυτοματοποιημένη θεματική ευρετηρίαση (subject indexing και ταξινόμησης εγγράφων), που αναπτύχθηκε από τη National Library of Finland (Εθνική Βιβλιοθήκη της Φινλανδίας). Μπορεί να χρησιμοποιεί τόσο λεξικο-βασισμένες Natural Language Processing (NLP) όσο και στατιστικές/ML μεθόδους (π.χ. TF-IDF, fastText, Omikuji, Maui), υποστηρίζοντας πολυγλωσσικότητα και πολλαπλά γνωστικά συστήματα (π.χ. SKOS) [19].

Εκπαιδεύεται με έτοιμα εγχειρίδια θεμάτων (vocabularies) και υφιστάμενα δεδομένα. Προτείνει θέματα για νέα έγγραφα, βοηθώντας βιβλιοθηκονόμους και αρχειονόμους. Χρησιμοποιείται, μεταξύ άλλων, από τη Finto AI, Yle, Εθνικές Βιβλιοθήκες, Getty, Erasmus, Οικονομικό Κέντρο Leibniz κ.ά.

Παρακάτω παρουσιάζονται οι βασικοί αλγόριθμοι (backends) που υποστηρίζει το Annif, με βάση σχετικές μελέτες και εμπειρικές αξιολογήσεις [19].

1. **Omikuji**

Ο Omikuji δεν είναι ένας μοναδικός αλγόριθμος αλλά μια βιβλιοθήκη για Extreme Multi-Label Classification (XMLC) που βασίζεται σε tree-based methods (Label Trees). Η βασική ιδέα του είναι ότι αντί να λύσει ένα πρόβλημα με χιλιάδες ετικέτες ως ένα ενιαίο πολυταξινομητικό πρόβλημα, «σπάει» το σύνολο ετικετών σε ιεραρχία (δένδρο) και εκπαιδεύει ταξινομητές σε κάθε κόμβο [48].

Η βασική μαθηματική διατύπωση στηρίζεται στο one-vs-rest linear classifier με hinge loss (όπως στο SVM):

$$\min_{\mathbf{w}} \frac{1}{2}|\mathbf{w}|^2 \; + \; C \sum_{i=1}^{N} \max(0,\; 1 - y_i(\mathbf{w}^\top \mathbf{x}_i))$$

*Σχήμα 10 Μαθηματικός τύπος Hinge Loss που χρησιμοποιείται στον Omikuji*

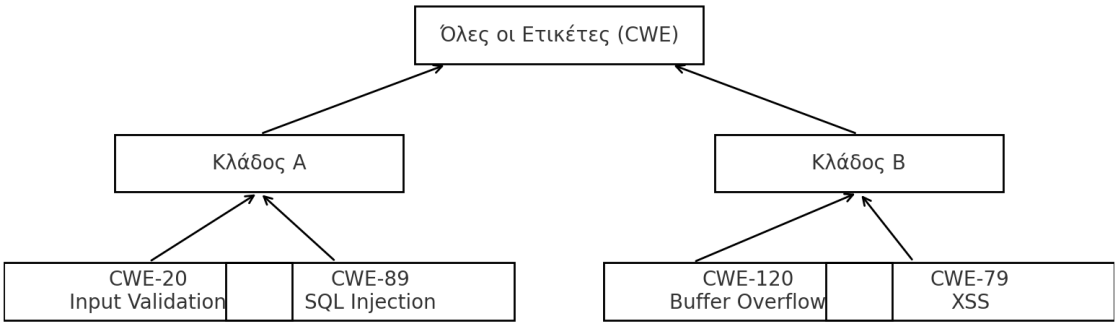
Όπου $y_i \in \{-1, +1\}$ είναι η πραγματική ετικέτα για το δείγμα i, w το διάνυσμα βαρών, $x_i$ το διάνυσμα χαρακτηριστικών και C μια παράμετρος κανονικοποίησης. Ο Omikuji εφαρμόζει τον τύπο αυτό σε κάθε κόμβο του δέντρου, επιτρέποντας την αποδοτική εκπαίδευση σε σενάρια με χιλιάδες ετικέτες.

Η βασική λογική είναι η εξής:

- Κατασκευάζεται ένα δέντρο, όπου κάθε κόμβος αντιπροσωπεύει ένα υποσύνολο ετικετών.
- Σε κάθε κόμβο εκπαιδεύεται τοπικός ταξινομητής (π.χ. γραμμικός SVM).
- Η πρόβλεψη γίνεται με “κατάβαση” στο δέντρο, αποκλείοντας σταδιακά ανενεργές ετικέτες.

Η πολυπλοκότητα μειώνεται δραματικά: από O(L) σε O(log L), όπου L το πλήθος ετικετών.

Η loss function μπορεί να είναι hinge loss, logistic loss ή cross-entropy. Χρησιμοποιείται για να λύνει προβλήματα Extreme Multi-Label με δέντρα ετικετών και τοπικούς ταξινομητές, μειώνοντας την πολυπλοκότητα κατά την πρόβλεψη. Κατάλληλο όταν οι ετικέτες είναι χιλιάδες και η κατανομή τους είναι ανισομερής (long tail).




*Σχήμα 11 Δενδρική δομή χώρου ετικετών (Omikuji – ενδεικτική απεικόνιση)*

Το σχήμα απεικονίζει μια δενδρική δομή ταξινόμησης ετικετών όπως εφαρμόζεται στον αλγόριθμο Omikuji προσαρμοσμένη στα δεδομένα της εργασίας.

Στην κορυφή βρίσκεται το σύνολο όλων των ετικετών (CWE). Αυτό το σύνολο χωρίζεται σε δύο μεγάλους κλάδους (Κλάδος A και Κλάδος B), οι οποίοι αποτελούν ενδιάμεσους κόμβους. Στη συνέχεια, κάθε κλάδος διασπάται σε πιο εξειδικευμένες κατηγορίες:

- Ο Κλάδος A οδηγεί στις κατηγορίες CWE-20 (Improper Input Validation) και CWE-89 (SQL Injection).
- Ο Κλάδος B οδηγεί στις κατηγορίες CWE-120 (Buffer Overflow) και CWE-79 (Cross-Site Scripting – XSS).

Η λογική του Omikuji βασίζεται ακριβώς σε αυτή τη διαίρεση: αντί να εξετάζει όλες τις πιθανές ετικέτες ταυτόχρονα (κάτι πολύ δαπανηρό υπολογιστικά όταν υπάρχουν εκατοντάδες ή χιλιάδες ετικέτες), το μοντέλο "κατεβαίνει" στο δέντρο βήμα-βήμα, μειώνοντας κάθε φορά τον χώρο αναζήτησης.

Έτσι, η δενδρική δομή:

- Απλοποιεί το πρόβλημα ταξινόμησης σε μικρότερες αποφάσεις.
- Μειώνει την πολυπλοκότητα της πρόβλεψης σε περιβάλλοντα Extreme Multi-Label.
- Επιτρέπει στο σύστημα να κλιμακώνεται αποτελεσματικά ακόμη και σε πολύ μεγάλες οντολογίες, όπως η CWE με εκατοντάδες κατηγορίες.

Με αυτόν τον τρόπο, ο Omikuji είναι ιδιαίτερα κατάλληλος για την αυτόματη θεματική ευρετηρίαση σε πεδία με πολλές και ανισομερώς κατανεμημένες ετικέτες, όπως η κυβερνοασφάλεια.

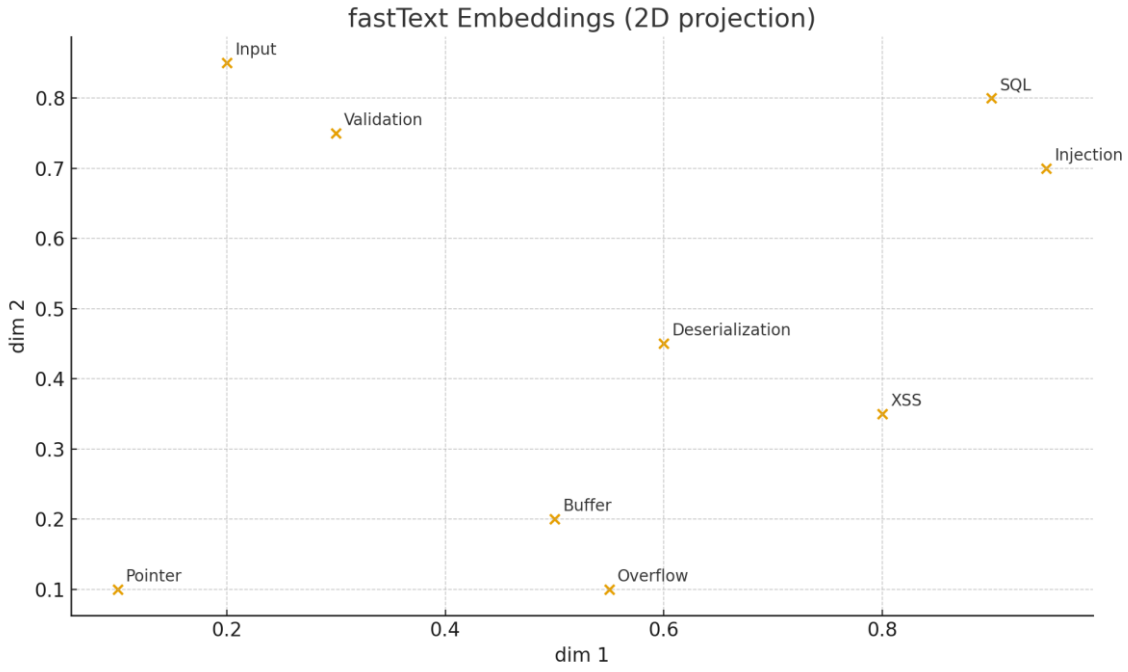
### 2. fastText

Ο αλγόριθμος fastText [49] αναπαριστά κάθε λέξη ως άθροισμα character n-gram embeddings. Έτσι, οι λέξεις διασπώνται σε υπομονάδες (subwords), π.χ. η λέξη “validation” σε “val”, “ali”, “lid”. Η πρόβλεψη βασίζεται σε γραμμικό ταξινομητή softmax:

$$p(y|x) = \frac{\exp(w_y \cdot h(x))}{\sum_{y'} \exp(w_{y'} \cdot h(x))}$$

*Σχήμα 12 Συνάρτηση softmax που χρησιμοποιεί ο αλγόριθμος fastText*

όπου h(x) είναι το embedding του κειμένου και $w_y$ τα βάρη του ταξινομητή. Για multi-label περιπτώσεις, χρησιμοποιείται hierarchical softmax ή negative sampling, μειώνοντας την πολυπλοκότητα από O(V) σε O(log V), όπου V το πλήθος ετικετών.




*Σχήμα 13 Ενδεικτική προβολή 2D embeddings fastText με ετικέτες όρων*

Το σχήμα παρουσιάζει μια δισδιάστατη προβολή (2D projection) των embeddings fastText για όρους σχετικούς με την κυβερνοασφάλεια. Στον οριζόντιο άξονα εμφανίζεται η πρώτη διάσταση (dim 1) και στον κάθετο η δεύτερη διάσταση (dim 2), οι οποίες προκύπτουν από μείωση της διάστασης του αρχικού διανυσματικού χώρου.

Κάθε σημείο στο διάγραμμα αντιστοιχεί σε μία λέξη, με παραδείγματα όπως Input, Validation, SQL, Injection, Buffer, Overflow, XSS, Pointer, Deserialization. Η σχετική θέση των σημείων υποδηλώνει τη σημασιολογική τους εγγύτητα:

- Οι όροι SQL και Injection τοποθετούνται κοντά μεταξύ τους, καθώς συνδέονται άμεσα με επιθέσεις SQL Injection.
- Οι όροι Input και Validation επίσης βρίσκονται σε κοντινή περιοχή, κάτι που αποτυπώνει τη σχέση τους με ζητήματα ελέγχου εισόδου.
- Οι όροι Buffer και Overflow συγκεντρώνονται χαμηλά στον χώρο, αποτυπώνοντας την εννοιολογική τους συσχέτιση σε ευπάθειες buffer overflow.
- Αντίστοιχα, όροι όπως Pointer και Deserialization τοποθετούνται σε διαφορετικές περιοχές, καθώς σχετίζονται με διαφορετικού τύπου προβλήματα.

Η οπτικοποίηση αυτή δείχνει πώς τα embeddings fastText συλλαμβάνουν σημασιολογικές σχέσεις μεταξύ όρων που σχετίζονται με συγκεκριμένες κατηγορίες CWE, προσφέροντας ένα πιο εκλεπτυσμένο πλαίσιο αναπαράστασης σε σχέση με τις παραδοσιακές στατιστικές μεθόδους.

### 3. TF-IDF

Η μέθοδος TF-IDF (Term Frequency–Inverse Document Frequency) αποτελεί κλασικό στατιστικό μοντέλο για την αναπαράσταση κειμένων. Ορίζεται ως:

$$TFIDF(t,d) = TF(t,d) \cdot \log \frac{N}{DF(t)}$$

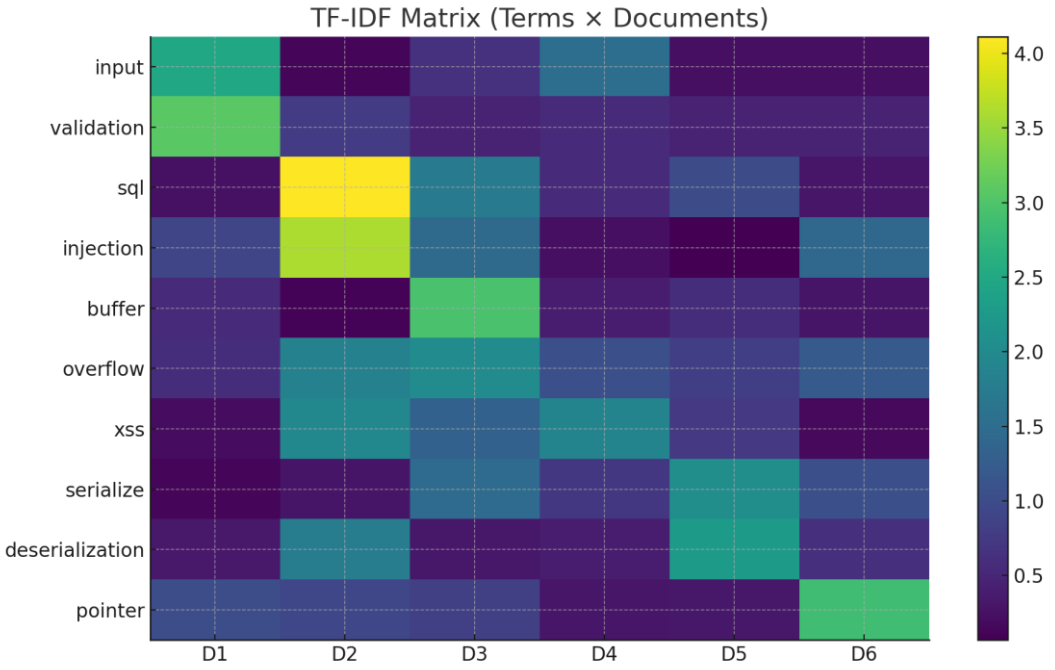
*Σχήμα 14 Υπολογισμός του βάρους TF-IDF*

όπου:

- TF(t,d): συχνότητα εμφάνισης του όρου t στο έγγραφο d.
- DF(t): πλήθος εγγράφων στα οποία εμφανίζεται ο όρος t.
- N: συνολικό πλήθος εγγράφων στη συλλογή.

Με αυτόν τον τρόπο, συχνά εμφανιζόμενοι όροι (π.χ. “password”) αποκτούν μικρότερο βάρος, ενώ όροι που διαφοροποιούν ένα έγγραφο (π.χ. “buffer overflow”) αποκτούν υψηλότερο βάρος. Η κανονικοποίηση (L2) εξασφαλίζει συγκρισιμότητα μεταξύ διαφορετικών κειμένων.

Παρέχει ερμηνευσιμότητα, ιδανικό ως baseline και ως είσοδος σε μοντέλα που απαιτούν αραιές αναπαραστάσεις.




*Σχήμα 15 Θερμικός χάρτης TF-IDF (λέξεις × έγγραφα)*

Το Σχήμα παρουσιάζει έναν θερμικό χάρτη (heat-map) TF-IDF, ο οποίος απεικονίζει τη σημασία συγκεκριμένων όρων σε ένα μικρό σύνολο εγγράφων (D1–D6). Στον κάθετο άξονα βρίσκονται οι όροι (π.χ. input, validation, sql, injection, buffer, overflow, xss, serialize, deserialization, pointer), ενώ στον οριζόντιο άξονα εμφανίζονται τα έγγραφα (D1–D6).

Η χρωματική κλίμακα δεξιά αντιστοιχεί στις τιμές του TF-IDF:

- Σκούρα χρώματα (μπλε/μοβ) υποδηλώνουν χαμηλές τιμές TF-IDF, δηλαδή λέξεις που εμφανίζονται συχνά σε πολλά κείμενα ή έχουν μικρή διακριτική ισχύ.
- Φωτεινά χρώματα (πράσινο, κίτρινο) αντιστοιχούν σε υψηλές τιμές TF-IDF, δηλαδή όρους που είναι ιδιαίτερα χαρακτηριστικοί για συγκεκριμένα έγγραφα.

Για παράδειγμα:

- Οι όροι “sql” και “injection” έχουν υψηλές τιμές στο έγγραφο D2, κάτι που υποδηλώνει ότι το κείμενο αυτό σχετίζεται έντονα με επιθέσεις τύπου SQL Injection.
- Αντίστοιχα, οι όροι “buffer” και “overflow” εμφανίζουν αυξημένες τιμές στο έγγραφο D3, υποδεικνύοντας ότι το περιεχόμενο αφορά προβλήματα buffer overflow.
- Ο όρος “pointer” ξεχωρίζει στο D6, υποδηλώνοντας ότι το έγγραφο σχετίζεται με σφάλματα χειρισμού δεικτών.

Η απεικόνιση αυτή διευκολύνει την κατανόηση της σχέσης όρων–εγγράφων και καταδεικνύει πώς η αναπαράσταση TF-IDF αναδεικνύει κρίσιμες λέξεις που βοηθούν στην κατηγοριοποίηση κειμένων σε κατηγορίες όπως οι CWE.

### 4. nn_ensemble

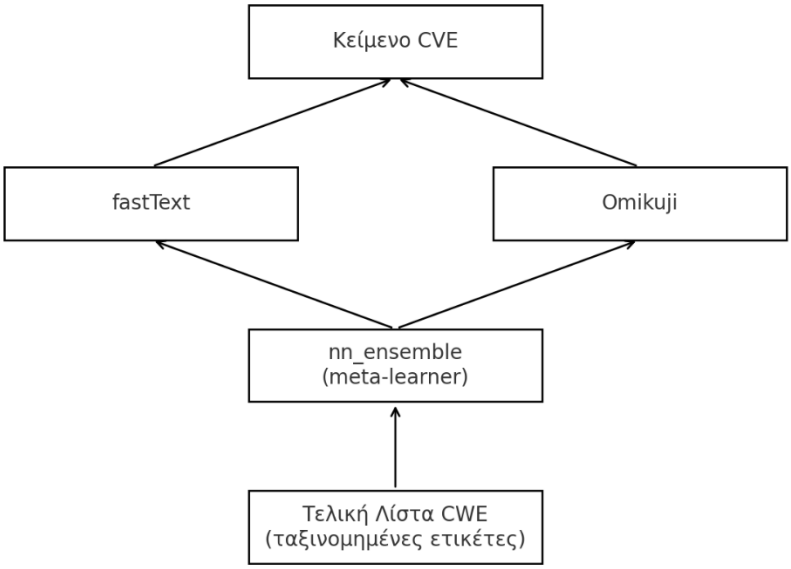
Ο Ensemble συνδυάζει τα αποτελέσματα διαφορετικών μοντέλων (π.χ. fastText και Omikuji). Η τελική πρόβλεψη βασίζεται σε meta-learner που μαθαίνει βάρη $\alpha_i$ για κάθε backend, εκπαιδευόμενο να ζυγίζει τις εισόδους έτσι ώστε να βελτιώνεται η κατάταξη των τελικών ετικετών (π.χ. αύξηση MAP/nDCG).

$$Score(y|x) = \sum_i \alpha_i \cdot Score_i(y|x)$$

*Σχήμα 16 Υπολογισμός του τελικού σκορ σε ensemble μοντέλο*

Έτσι, η ακρίβεια βελτιώνεται αφού αξιοποιούνται διαφορετικά πλεονεκτήματα:

- Ο fastText συλλαμβάνει μορφολογικές/γλωσσικές σχέσεις.
- Ο Omikuji χειρίζεται αποτελεσματικά μεγάλο πλήθος ετικετών.




*Σχήμα 17 Ροή συνδυασμού μοντέλων στο Annif (fastText, Omikuji → nn_ensemble)*

Το σχήμα απεικονίζει τη διαδικασία συνδυασμού δύο backends στο εργαλείο Annif, συγκεκριμένα του Omikuji και του fastText. Η είσοδος του συστήματος είναι το κείμενο που προέρχεται από μία περιγραφή ευπάθειας στο σύνολο CVE.

Το κείμενο αυτό επεξεργάζεται ταυτόχρονα από τα δύο μοντέλα. Από τη μία πλευρά, το fastText αξιοποιεί αναπαραστάσεις λέξεων με χαρακτήρες n-grams, γεγονός που του επιτρέπει να γενικεύει ακόμη και σε σπάνιους ή νέους όρους, προσφέροντας υψηλή απόδοση σε ταξινομήσεις κειμένων. Από την άλλη πλευρά, το Omikuji εφαρμόζει τεχνικές Extreme Multi-Label Classification, χρησιμοποιώντας δενδρικές δομές που μπορούν να χειριστούν αποτελεσματικά πολύ μεγάλο αριθμό κατηγοριών, όπως οι κατηγορίες CWE.

Οι προβλέψεις που παράγονται από τα δύο αυτά μοντέλα δεν παρουσιάζονται μεμονωμένα, αλλά συνδυάζονται από τον nn_ensemble, ο οποίος λειτουργεί ως meta-learner. Ο nn_ensemble μαθαίνει να αποδίδει κατάλληλα βάρη στις εξόδους του Omikuji και του fastText, βελτιστοποιώντας έτσι τη συνολική απόδοση του συστήματος.

Το τελικό αποτέλεσμα του συνδυασμού είναι μία ταξινομημένη λίστα κατηγοριών CWE. Στη λίστα αυτή, οι πιο σχετικές ετικέτες εμφανίζονται στις πρώτες θέσεις, διευκολύνοντας τη διαδικασία αυτόματης θεματικής ευρετηρίασης και επιτυγχάνοντας υψηλότερες τιμές στις μετρικές αξιολόγησης, σε σύγκριση με τη χρήση ενός μόνο αλγορίθμου.

Με αυτό τον τρόπο, το Annif αξιοποιεί τη συμπληρωματικότητα των δύο backends:

- Το fastText συλλαμβάνει μορφολογικές και συντακτικές συσχετίσεις.
- Το Omikuji βελτιώνει την απόδοση σε Extreme Multi-Label σενάρια.

Ο συνδυασμός τους μέσω του nn_ensemble οδηγεί σε καλύτερη ακρίβεια και υψηλότερες τιμές σε μετρικές όπως το MAP και το nDCG, σε σχέση με τη χρήση κάθε μοντέλου ξεχωριστά.

## 4.5 Σύγχρονες Προσεγγίσεις (Transformers, BERT)

Η σύγχρονη έρευνα στην IR και στην ταξινόμηση κειμένων έχει στραφεί σε deep learning μοντέλα, όπως οι Transformers [40] και τα μοντέλα BERT [44]. Τα μοντέλα αυτά χρησιμοποιούν μηχανισμό attention:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

*Σχήμα 18 Υπολογισμός του μηχανισμού προσοχής που χρησιμοποιείται στα Transformer μοντέλα*

Όπου Q, K,V είναι πίνακες queries, keys, values και $d_k$ η διάσταση. Αυτό επιτρέπει στο μοντέλο να συλλαμβάνει συσχετίσεις μακράς απόστασης στα κείμενα και να δημιουργεί context-aware embeddings.

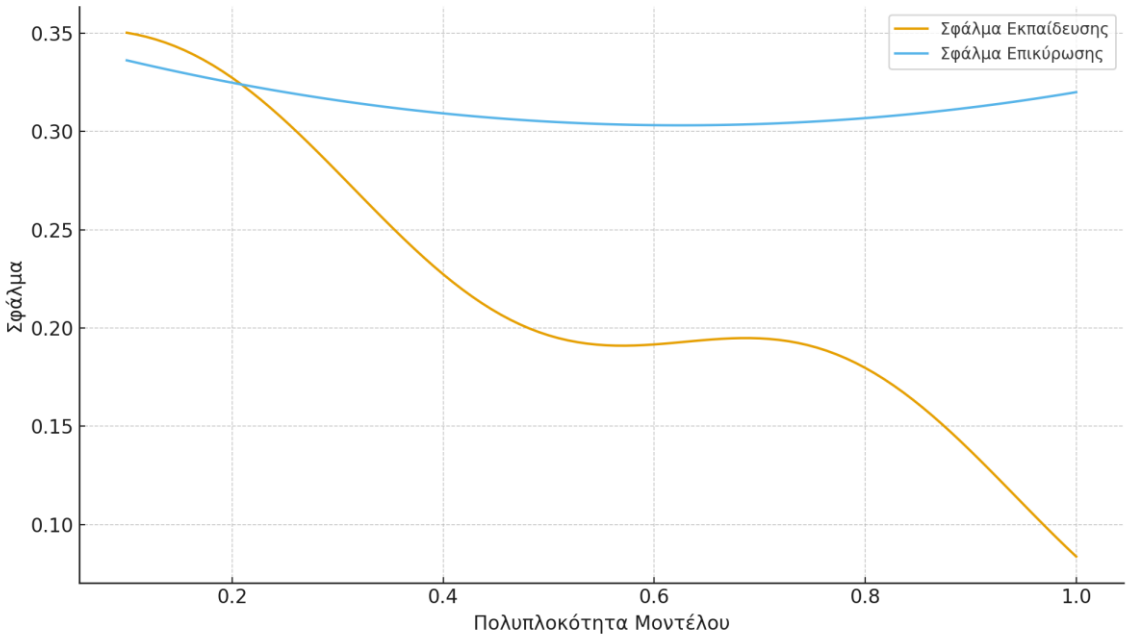
Ωστόσο, η ενσωμάτωσή τους στο Annif δεν είναι εφικτή λόγω:

- Υψηλών υπολογιστικών απαιτήσεων (GPU, μνήμη).
- Μεγάλου κόστους εκπαίδευσης/ fine-tuning σε μεγάλα datasets.
- Έλλειψη άμεσης υποστήριξης από το Annif (που είναι σχεδιασμένο για πιο “lightweight” αλγορίθμους).

## 4.6 Πλεονεκτήματα και Μειονεκτήματα

Κάθε αλγόριθμος έχει διαφορετικά πλεονεκτήματα: Ο fastText ταχύτητα και γενίκευση σε κείμενα, ο Omikuji για Extreme Multi-Label σενάρια και ο nn_ensemble για βελτιωμένη κατάταξη μέσω συνδυασμού. Σε πρακτικές ροές Annif, μια υβριδική στρατηγική με fine-tuning των υπερπαραμέτρων και σωστή προεπεξεργασία παράγει συνήθως την καλύτερη συνολική απόδοση (ιδίως σε MAP/nDCG).

Στο παρακάτω σχήμα παρουσιάζεται με γραφικό τρόπο η μεταβολή για την καλύτερη κατανόηση της εκπαίδευσης αλγορίθμων μηχανικής μάθησης και την επιλογή της κατάλληλης πολυπλοκότητας μοντέλου για εφαρμογές όπως η ταξινόμηση ευπαθειών (CVE) σε κατηγορίες CWE.




*Σχήμα 19 Tradeoff μεροληψίας–διακύμανσης: σφάλμα εκπαίδευσης έναντι σφάλματος επικύρωσης*

Το σχήμα παρουσιάζει τη σχέση μεταξύ της πολυπλοκότητας ενός μοντέλου και του σφάλματος που παρατηρείται κατά την εκπαίδευση και την επικύρωσή του. Στον οριζόντιο άξονα απεικονίζεται η πολυπλοκότητα του μοντέλου, ενώ στον κατακόρυφο άξονα εμφανίζεται η τιμή του σφάλματος.

Η καμπύλη του σφάλματος εκπαίδευσης μειώνεται σταθερά όσο αυξάνεται η πολυπλοκότητα. Αυτό συμβαίνει επειδή τα πιο σύνθετα μοντέλα μπορούν να προσαρμοστούν καλύτερα στα δεδομένα εκπαίδευσης, μειώνοντας το σφάλμα. Ωστόσο, η συνεχής μείωση του σφάλματος εκπαίδευσης δεν αποτελεί εγγύηση για καλύτερη γενίκευση.

Η καμπύλη του σφάλματος επικύρωσης αρχικά μειώνεται καθώς η πολυπλοκότητα αυξάνει, γεγονός που υποδηλώνει βελτίωση της ικανότητας του μοντέλου να συλλαμβάνει τα υποκείμενα πρότυπα των δεδομένων. Ωστόσο, μετά από ένα σημείο, το σφάλμα επικύρωσης σταθεροποιείται ή και αυξάνεται. Αυτό συμβαίνει διότι το μοντέλο αρχίζει να προσαρμόζεται υπερβολικά στις ιδιαιτερότητες των δεδομένων εκπαίδευσης (overfitting), χάνοντας την ικανότητα γενίκευσης σε νέα, άγνωστα δεδομένα.

Η συμπεριφορά αυτή αποτυπώνει το tradeoff μεροληψίας–διακύμανσης (bias–variance tradeoff):

- Στα πολύ απλά μοντέλα (χαμηλή πολυπλοκότητα), το σφάλμα είναι υψηλό και στις δύο καμπύλες λόγω υποπροσαρμογής (underfitting).
- Στα πολύ σύνθετα μοντέλα (υψηλή πολυπλοκότητα), το σφάλμα εκπαίδευσης γίνεται πολύ χαμηλό, αλλά το σφάλμα επικύρωσης αυξάνεται λόγω υπερπροσαρμογής (overfitting).
- Το βέλτιστο σημείο βρίσκεται κάπου ενδιάμεσα, όπου οι δύο καμπύλες είναι αρκετά χαμηλές και το μοντέλο επιτυγχάνει καλή ισορροπία ανάμεσα σε ακρίβεια και γενίκευση.

Συμπερασματικά, καταγράφονται σε πίνακα τα χαρακτηριστικά, καθώς και τα πλεονεκτήματα – μειονεκτήματα των βασικότερων αλγορίθμων που μπορούν να χρησιμοποιηθούν στο Annif.

*Πίνακας 1 Βασικοί Αλγόριθμοι (backends) του Annif*

| Αλγόριθμος (Backend) | Περιγραφή | Πλεονεκτήματα | Μειονεκτήματα |
|---|---|---|---|
| **Omikuji** | Χρησιμοποιεί decision trees και random forests για extreme multi-label ταξινόμηση | Υψηλή ακρίβεια, κατάλληλο για μεγάλα σύνολα και πολλές κατηγορίες | Υψηλή κατανάλωση μνήμης |
| **FastText** | Γραμμικό μοντέλο ταξινόμησης με subword embeddings – κατάλληλο για πολυγλωσσικά σύνολα | Υψηλή ακρίβεια, ταχύτητα, υποστήριξη πολλών γλωσσών | Δεν παρέχει εξηγησιμότητα |
| **TF-IDF** | Βασίζεται στο vector space model, αποδίδοντας βάρη λέξεων ανάλογα με τη συχνότητα εμφάνισής τους | Ταχύτητα, ευκολία χρήσης, καλή baseline απόδοση | Περιορισμένη σημασιολογική κατανόηση |
| **NN Ensemble** | Συνδυάζει πολλούς backends (π.χ. FastText + TF-IDF) σε σύστημα βαρών | Υψηλή γενικευσιμότητα, ευελιξία | Πολυπλοκότητα, αυξημένος υπολογιστικός φόρτος |

Για κάθε περίπτωση εφαρμογής, το καταλληλότερο backend στο Annif εξαρτάται από τη δομή και το μέγεθος του εκπαιδευτικού συνόλου, την πολυπλοκότητα και το πλήθος των κατηγοριών, την ανάγκη για ταχύτητα και ακρίβεια. Στην εργασία αυτή εξετάστηκαν σχεδόν όλοι οι αλγόριθμοι πιλοτικά και επιλέχθηκαν οι αλγόριθμοι Omikuji και FastText που είχαν τις καλύτερες μετρικές αξιολόγησης, ενώ η τελική κατηγοριοποίηση έγινε με τον αλγόριθμο NN Enseble που συνδύασε τα αποτελέσματα των Omikuji και FastText πετυχαίνοντας ακόμη καλύτερα αποτελέσματα.

Στην παρούσα εργασία το Annif φορτώνει το λεξιλόγιο SKOS RDF που παρέχεται, προκειμένου να αναγνωρίζει τα έγκυρα URI concepts (CWE IDs), τις ετικέτες (prefLabel, altLabel) και τη μεταξύ τους ιεραρχία (broader, narrower σχέσεις). Με αυτόν τον τρόπο, το σύστημα γνωρίζει εκ των προτέρων το σύνολο των θεματικών εννοιών (labels) που θα προβλέψει.

Ο εκπαιδευμένος ταξινομητής μαθαίνει να προβλέπει ένα ή περισσότερα URI από το λεξιλόγιο CWE για νέα κείμενα στα Αγγλικά, όπως περιγραφές ευπαθειών από το CVE. Πρόκειται για πρόβλημα πολύ-ετικετικής ταξινόμησης κειμένου (multi-label text classification), όπου κάθε έγγραφο μπορεί να αντιστοιχεί σε πολλαπλές θεματικές έννοιες.

Για την εκπαίδευση και αξιολόγηση εφαρμόστηκε διαχωρισμός δεδομένων σε:

- Training set: 90% (~90.000 εγγραφές)
- Test set: 10% (~10.000 εγγραφές)

Η επιλογή πραγματοποιήθηκε με stratified split ως προς το URI, ώστε κάθε κατηγορία να υπάρχει και στα δύο σύνολα και να αποφεύγονται labels που δεν εμφανίστηκαν στην εκπαίδευση.

Κώδικας Python με stratified split:

```
from sklearn.model_selection import train_test_split

train_df, test_df = train_test_split(
    df_valid, test_size=0.1, random_state=42, stratify=df_valid['text']
)
```

### 4.7 Παράδειγμα παραμετροποίησης αλγόριθμου στο Annif

Το Omikuji είναι υλοποίηση μιας οικογένειας αλγορίθμων tree-based για extreme multi-label classification, με δυνατότητα μίμησης αλγορίθμων όπως Bonsai και AttentionXML. Χρησιμοποιεί ιεραρχική οργάνωση των labels μέσω clustering και επιτυγχάνει υψηλή ποιότητα αποτελεσμάτων ακόμα και χωρίς εκτενή βελτιστοποίηση παραμέτρων.

*Πίνακας 2 Βασικές Παράμετροι (Προεπιλεγμένες στο Annif)*

| **Παράμετρος** | **Τιμή** | **Περιγραφή** |
|---|---|---|
| **n_trees** | 3 | Αριθμός δέντρων στο μοντέλο. |
| **min_branch_size** | 100 | Ελάχιστος αριθμός παραδειγμάτων ανά split. |
| **max_depth** | 20 | Μέγιστο βάθος δέντρου. |
| **centroid_threshold** | 0.0 | Κατώφλι για σύγκλιση κέντρων. |
| **collapse_every_n_layers** | 0 | Συγχώνευση επιπέδων κάθε n επίπεδα. |
| **loss_type** | Hinge | Συνάρτηση κόστους τύπου SVM. |
| **c** | 1.0 | Παράμετρος κανονικοποίησης. |
| **max_iter** | 20 | Μέγιστες επαναλήψεις εκπαίδευσης γραμμικού μοντέλου. |

Το Annif δεν επιτρέπει αλλαγή όλων των παραμέτρων του Omikuji μέσω του project.cfg. Ο χρήστης μπορεί να τροποποιήσει μόνο κάποιες όπως: limit, min_df, ngram, cluster_balanced, cluster_k, max_depth και collapse_every_n_layers.

Bonsai-style

```
[omikuji-bonsai-en]
language=en
backend=omikuji
cluster_balanced=False
cluster_k=100
max_depth=3
```

AttentionXML-style

```
[omikuji-attention-en]
language=en
backend=omikuji
cluster_balanced=False
cluster_k=2
collapse_every_n_layers=5
```

*Σχήμα 20 Παραδείγματα Παραμετροποίησης backend Annif*

Το Annif με backend Omikuji προσφέρει μια αποδοτική λύση για την αυτόματη θεματική ευρετηρίαση σε προβλήματα με μεγάλο αριθμό ετικετών, όπως η ταξινόμηση ευπαθειών με βάση το CWE. Η σωστή

επιλογή και προσαρμογή των παραμέτρων, σε συνδυασμό με ισορροπημένα και αντιπροσωπευτικά δεδομένα εκπαίδευσης, βελτιώνει σημαντικά την ακρίβεια και την αξιοπιστία των προβλέψεων.

### 4.8 Ροή χρήσης: εκπαίδευση, πρόβλεψη, αξιολόγηση

Η τυπική ροή χρήσης του Annif διαμορφώνεται σε τρία στάδια: εκπαίδευση (training), πρόβλεψη (suggestion/prediction), και αξιολόγηση (evaluation).

Το πρώτο στάδιο αφορά την εκμάθηση ενός μοντέλου πάνω σε ένα σύνολο εκπαίδευσης, το οποίο περιλαμβάνει ζεύγη κειμένου–εννοιών (text–subjects), με τις έννοιες ορισμένες σε ένα SKOS λεξιλόγιο. Το λεξιλόγιο (RDF σε SKOS μορφή) και οι παράμετροι του μοντέλου καθορίζονται μέσω του αρχείου project.cfg.

```
[cve-project]
dir = annif-projects/cve-project
name = Cybersecurity Project
language = en
analyzer = snowball(english)
vocab = cve-project
backend = omikuji
```

*Σχήμα 21 Στιγμιότυπο από ρυθμίσεις μοντέλου στο project.cfg*

Μετά την εκπαίδευση, το μοντέλο μπορεί να χρησιμοποιηθεί για πρόβλεψη θεματικών εννοιών σε νέα, μη ετικεταρισμένα κείμενα. Το τρίτο στάδιο αφορά τη μέτρηση της απόδοσης του μοντέλου με βάση γνωστές ετικέτες (ground truth). Κατά την αξιολόγηση του μοντέλου μπορούν να υπολογιστούν μετρικές όπως Precision, Recall, F1-score, MAP και nDCG. Η αξιολόγηση λαμβάνει υπόψη τη σειρά κατάταξης των προβλέψεων και υποστηρίζει multi-label περιπτώσεις.

### 4.9 Μετρικές απόδοσης: Precision, Recall, F1-score

Οι μετρικές Precision, Recall και F1-score χρησιμοποιούνται ευρέως στην αξιολόγηση συστημάτων ταξινόμησης κειμένου, όπως το Annif. Οι μετρικές αυτές παρέχουν ένα μέτρο της ακρίβειας και της πληρότητας των προβλέψεων σε σχέση με τις αναμενόμενες θεματικές ετικέτες.

Η ακρίβεια (Precision) εκφράζει το ποσοστό των σωστών προβλέψεων μεταξύ όλων των προβλέψεων που έκανε το σύστημα.

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

*Σχήμα 22 Ο μαθηματικός ορισμός της μετρικής Precision*

Όπου:

TP: True Positives (σωστά θετικές προβλέψεις)

FP: False Positives (λανθασμένα θετικές προβλέψεις)

Υψηλή Precision σημαίνει ότι οι προτεινόμενες ετικέτες είναι κατά κύριο λόγο σωστές.

Η recall μετρά το ποσοστό των σωστά προβλεπόμενων ετικετών ως προς το σύνολο των ετικετών που θα έπρεπε να έχουν προβλεφθεί.

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

*Σχήμα 23 Ο μαθηματικός τύπος υπολογισμού της Recall*

Όπου:

FN: False Negatives (παραληφθείσες σωστές προβλέψεις)

Υψηλή recall σημαίνει ότι το σύστημα αναγνωρίζει τη συντριπτική πλειοψηφία των σχετικών θεμάτων.

Το F1-score (Αρμονικός Μέσος) συνδυάζει την ακρίβεια και την ανακλητικότητα σε μία ενιαία μετρική, υπολογίζοντας τον αρμονικό μέσο όρο τους.

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

*Σχήμα 24 Ο μαθηματικός τύπος του F1-score*

Το F1-score είναι χρήσιμο όταν χρειάζεται ισορροπία μεταξύ Precision και recall, ιδιαίτερα σε περιπτώσεις ανισόρροπων δεδομένων.

## 4.10 Μετρικές κατάταξης: MAP, nDCG

Οι μετρικές κατάταξης, Mean Average Precision (MAP) και normalized Discounted Cumulative Gain (nDCG), χρησιμοποιούνται για την αξιολόγηση της ποιότητας της κατάταξης των προβλέψεων ενός συστήματος, όπως το Annif. Οι μετρικές αυτές λαμβάνουν υπόψη όχι μόνο την ακρίβεια των προβλέψεων, αλλά και τη σειρά με την οποία επιστρέφονται οι θεματικές ετικέτες.

Η MAP υπολογίζει τη μέση ακρίβεια των σωστών προβλέψεων ανά τεκμήριο, λαμβάνοντας υπόψη τη θέση στην οποία εμφανίζεται κάθε σωστή ετικέτα.

$$AP = \frac{\sum Precision@k\ (\sigma\omega\sigma\tau\acute{\epsilon}\varsigma\ \pi\rho o\beta\lambda\acute{\epsilon}\psi\epsilon\iota\varsigma)}{A\rho\iota\theta\mu\acute{o}\varsigma\ \sigma\omega\sigma\tau\acute{\omega}\nu\ \epsilon\tau\iota\kappa\epsilon\tau\acute{\omega}\nu}$$

*Σχήμα 25 Ο μαθηματικός τύπος για τον υπολογισμό του Average Precision (AP)*

$$MAP = \frac{1}{N} \sum_{i=1}^{N} AP_i$$

*Σχήμα 26 MAP = Μέσος όρος όλων των Average Precision (AP)*

Η MAP είναι κατάλληλη για multi-label προβλήματα και μετρά πόσο νωρίς στη λίστα κατάταξης βρίσκονται οι σωστές ετικέτες.

Η nDCG (normalized Discounted Cumulative Gain) είναι μετρική που βασίζεται στην ιδέα ότι οι σωστές ετικέτες έχουν μεγαλύτερη αξία όταν εμφανίζονται νωρίς στη λίστα προβλέψεων.

$$DCG = \sum \frac{relevance_i}{\log_2(i+1)}$$

$$nDCG = \frac{DCG}{IDCG}$$

*Σχήμα 27 Ο μαθηματικός τύπος για τον υπολογισμό του nDGG*

Όπου:

i: θέση στην κατάταξη,

$relevance_i$: 1 αν η πρόβλεψη είναι σωστή, 0 αλλιώς,

IDCG: η μέγιστη δυνατή τιμή DCG για τις σωστές ετικέτες.

Η nDCG είναι ιδανική όταν ενδιαφέρει η σειρά εμφάνισης των σωστών προβλέψεων, κάτι πολύ συνηθισμένο στη θεματική ευρετηρίαση.

Στο Annif, οι μετρικές MAP και nDCG χρησιμοποιούνται για να αξιολογηθεί η κατάταξη των προτεινόμενων εννοιών από το μοντέλο σε σχέση με τις αναμενόμενες. Οι τιμές MAP και nDCG κυμαίνονται από 0 έως 1, με το 1 να αντιστοιχεί σε ιδανική κατάταξη.

Συμπερασματικά, η ροή χρήσης του Annif προσφέρει ένα δομημένο και επεκτάσιμο πλαίσιο για αυτόματη θεματική ευρετηρίαση. Η δυνατότητα συνδυασμού διαφορετικών αλγορίθμων, η υποστήριξη SKOS οντολογιών και η ενσωμάτωση αξιολόγησης καθιστούν το Annif ιδανικό εργαλείο για έρευνες σε πληροφοριακά και επιστημονικά συστήματα.

## 4.11 Επίλογος

Στο παρόν κεφάλαιο παρουσιάστηκαν οι βασικοί αλγόριθμοι μηχανικής μάθησης που υποστηρίζει το Annif και η συμβολή τους στην αυτόματη θεματική ευρετηρίαση. Οι αλγόριθμοι αυτοί διαφέρουν ως προς την προσέγγιση και τις ιδιότητές τους.

Ο TF-IDF παρέχει απλότητα και ερμηνευσιμότητα, ο fastText αξιοποιεί υπολεξικές αναπαραστάσεις για μεγαλύτερη γενίκευση, το Omikuji προσφέρει κλιμακωσιμότητα σε σενάρια Extreme Multi-Label, ενώ το nn_ensemble συνδυάζει διαφορετικά μοντέλα ώστε να βελτιστοποιήσει τη συνολική απόδοση.

Η μελέτη αυτών των αλγορίθμων δείχνει ότι κάθε προσέγγιση έχει τα δικά της πλεονεκτήματα και περιορισμούς, και η επιλογή τους εξαρτάται από τις απαιτήσεις του εκάστοτε προβλήματος. Παράλληλα, παρουσιάστηκαν οι βασικές αρχές της εκπαίδευσης και αξιολόγησης συστημάτων ταξινόμησης κειμένου, καθώς και οι μετρικές που χρησιμοποιούνται για τη μέτρηση της απόδοσης, όπως οι Precision, Recall, F1-score, MAP και nDCG.

Συνολικά το κεφάλαιο ανέδειξε, πως η ενσωμάτωση διαφορετικών αλγοριθμικών προσεγγίσεων και η αξιολόγησή τους μέσα από κατάλληλες μετρικές, αποτελούν θεμελιώδη βήματα για την ανάπτυξη αποτελεσματικών συστημάτων αυτόματης θεματικής ευρετηρίασης.

# Κεφάλαιο 5ο: Μεθοδολογία

## 5.1. Εισαγωγή

Η μεθοδολογία που ακολουθήθηκε για την ανάπτυξη, εκπαίδευση και αξιολόγηση του μοντέλου αυτόματης θεματικής ευρετηρίασης βασίστηκε στο εργαλείο Annif (έκδοση 1.3.1) και σχεδιάστηκε με σκοπό να είναι αναπαραγώγιμη, επεκτάσιμη και συμβατή με διεθνή πρότυπα. Το κεφάλαιο παρουσιάζει αναλυτικά τα στάδια της διαδικασίας, το περιβάλλον ανάπτυξης, τη συλλογή και προετοιμασία δεδομένων, την εκπαίδευση των μοντέλων, καθώς και την αξιολόγηση της απόδοσής τους.

Η συνολική διαδικασία ακολούθησε διακριτά βήματα: Εγκατάσταση απαιτούμενων λογισμικών, συλλογή δεδομένων, προεπεξεργασία δεδομένων, μετατροπή σε SKOS, εκπαίδευση μοντέλων, αξιολόγηση αποτελεσμάτων. Η σχεδίαση αυτή επέτρεψε την τμηματική παρακολούθηση της προόδου, την ανίχνευση σφαλμάτων και τη συνεχή βελτιστοποίηση.

## 5.2. Εγκατάσταση και ρύθμιση του Annif στο PyCharm

Η ανάπτυξη και εκτέλεση των πειραμάτων πραγματοποιήθηκε σε περιβάλλον Windows, με Python 3.10 και IDE το PyCharm. Το PyCharm είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης για τη γλώσσα Python, το οποίο παρέχει δυνατότητες διαχείρισης εικονικών περιβαλλόντων, ενσωματωμένο τερματικό και εργαλεία αποσφαλμάτωσης. Η εγκατάσταση του Annif, ενός εργαλείου ανοιχτού κώδικα για την αυτόματη θεματική ευρετηρίαση, πραγματοποιήθηκε μέσω της ενσωματωμένης γραμμής εντολών του PyCharm και ακολούθησε τα παρακάτω βήματα:

- Δημιουργία έργου και εικονικού περιβάλλοντος (virtual environment): Στο PyCharm δημιουργήθηκε ένα νέο έργο και επιλέχθηκε η χρήση εικονικού περιβάλλοντος Python (venv) για την απομόνωση των εξαρτήσεων. Η χρήση venv εξασφαλίζει ότι οι βιβλιοθήκες που εγκαθίστανται δεν επηρεάζουν άλλα έργα του συστήματος.
- Εγκατάσταση του Annif και απαραίτητων εξαρτήσεων: Μέσα από το ενσωματωμένο τερματικό του PyCharm, εκτελέστηκε η εντολή: pip install Annif
- Ρύθμιση του εκτελέσιμου Annif στο περιβάλλον: Μετά την εγκατάσταση, το εκτελέσιμο αρχείο annif είναι διαθέσιμο μέσα στο φάκελο Scripts του εικονικού περιβάλλοντος (σε Windows: .venv\Scripts\annif.exe). Το PyCharm ρυθμίστηκε ώστε οι εντολές να εκτελούνται με αυτό το περιβάλλον.
- Έλεγχος της εγκατάστασης: Η επιτυχής εγκατάσταση επιβεβαιώθηκε με την εκτέλεση:
  annif –version
  η οποία εμφάνισε την τρέχουσα έκδοση του εργαλείου (στην παρούσα εργασία: 1.3.1).
- Δημιουργία δομής έργων Annif: Στο πλαίσιο του PyCharm δημιουργήθηκε ένας κεντρικός φάκελος έργων (annif-projects), στον οποίο αποθηκεύτηκαν οι φάκελοι κάθε μοντέλου Annif, περιλαμβάνοντας τα αρχεία ρύθμισης (project.cfg), το λεξιλόγιο (SKOS RDF) και τα σύνολα δεδομένων εκπαίδευσης/ελέγχου (train.tsv, test.tsv).

Η παραπάνω διαδικασία επέτρεψε την απρόσκοπτη ενσωμάτωση του Annif στο αναπτυξιακό περιβάλλον του PyCharm, παρέχοντας τη δυνατότητα εκτέλεσης, αποσφαλμάτωσης και διαχείρισης των πειραμάτων χωρίς την ανάγκη εξωτερικών εργαλείων.

## 5.3 Συλλογή και Προετοιμασία Δεδομένων

Η διαδικασία ξεκίνησε με τη συλλογή δεδομένων από το σύνολο CVE (Common Vulnerabilities and Exposures), το οποίο περιλαμβάνει περιγραφές ευπαθειών στον τομέα της κυβερνοασφάλειας. Κάθε εγγραφή περιείχε περιγραφή κειμένου και μία ή περισσότερες κατηγορίες CWE (Common Weakness Enumeration).

Η συλλογή των δεδομένων πραγματοποιήθηκε μέσω του NVD (National Vulnerability Database) API, το οποίο παρέχει πρόσβαση σε δομημένες πληροφορίες για ευπάθειες κυβερνοασφάλειας (CVE entries). Το API key παρέχεται δωρεάν από το NVD: https://nvd.nist.gov/developers/request-an-api-key [37].

Χρησιμοποιήθηκε αυτοματοποιημένο Python script που καλεί επαναληπτικά το API, αξιοποιώντας παραμέτρους φιλτραρίσματος (π.χ. χρονικά διαστήματα, λέξεις-κλειδιά) και αποθηκεύει τα αποτελέσματα σε μορφή xlsx.

```
API_KEY = "d80b6eb2-9d7a-419d-a4e6-[redacted]"

def fetch_cve_data_interval(start_date, end_date, results_per_page=1000, api_key=None):
    base_url = "https://services.nvd.nist.gov/rest/json/cves/2.0"
    start_index = 0
    all_items = []

    headers = {"apiKey": api_key} if api_key else {}
```

*Σχήμα 28 Στιγμιότυπο από script*

```
def save_to_excel(data, filename):
    df = pd.DataFrame(data, columns=["Keywords", "Description"])
    df.to_excel(filename, index=False)
    print(f"Saved {len(data)} records to {filename}")

if __name__ == "__main__":
    api_key = API_KEY
    all_filtered = []
    seen_descriptions = set()

    start_year = 2016
    end_year = 2025
    interval_days = 7
```

*Σχήμα 29 Φιλτράρισμα εγγραφών από το 2016 έως την ημέρα υλοποίησης ανά εβδομάδα*

Για τη θεματική κατηγοριοποίηση των ευπαθειών χρησιμοποιήθηκε το επίσημο CWE λεξικό του οργανισμού MITRE, το οποίο είναι διαθέσιμο σε μορφή XML. Η ανάκτηση έγινε μέσω του δημόσιου αποθετηρίου του MITRE, εξασφαλίζοντας ότι χρησιμοποιείται η πλέον ενημερωμένη έκδοση του λεξικού.

Στη συνέχεια, εφαρμόστηκε διαδικασία μετατροπής του αρχείου XML σε μορφή RDF (Resource Description Framework), ώστε να είναι συμβατό με το πρότυπο SKOS (Simple Knowledge Organization System) που χρησιμοποιεί το Annif για την αναπαράσταση εννοιών.

Το παραγόμενο RDF λεξικό χρησιμοποιήθηκε ως λεξιλόγιο αναφοράς στην εκπαίδευση και στις προβλέψεις του Annif, επιτρέποντας την αντιστοίχιση των περιγραφών ευπαθειών με τις αντίστοιχες κατηγορίες CWE.

Για να μπορέσει να γίνει αντιστοίχιση των κατηγοριών CWE έγινε μετατροπή των κατηγοριών CWE στη συλλογή δεδομένων από το σύνολο CVE σε URIs. Στη συνέχεια, τα δεδομένα μετατράπηκαν σε μορφή TSV κατάλληλη για την εκπαίδευση του Annif, διατηρώντας τα πεδία περιγραφή της ευπάθειας και κατηγορία CWE σε URIs σχετική με το λεξικό που ανακτήθηκε από το MITRE.

Η διαδικασία αυτή εξασφάλισε ότι το σύνολο εκπαίδευσης θα είναι ενημερωμένο, πλήρες και συμβατό με τις απαιτήσεις του συστήματος ταξινόμησης. Πραγματοποιήθηκε καθαρισμός των δεδομένων, αφαίρεση θορύβου και έλεγχος για απουσία τιμών (null values). Για τον διαχωρισμό των δεδομένων σε εκπαίδευση (train) και αξιολόγηση (test), εφαρμόστηκε αναλογικός καταμερισμός ανά κατηγορία CWE, ώστε να διατηρηθεί η αναλογία παραδειγμάτων ανά κατηγορία.

Έπειτα διενεργήθηκε καθαρισμός και προεπεξεργασία με:

- Αφαίρεση εγγραφών με κενά πεδία.
- Κανονικοποίηση URIs (αφαίρεση γωνιακών αγκυλών, διαχωρισμός με βάση το σύμβολο #, μετατροπή σε κεφαλαία).
- Έλεγχος και ενοποίηση πολλαπλών ετικετών ανά εγγραφή.
- Εφαρμόστηκε αναλογική δειγματοληψία ανά κατηγορία CWE ώστε να διατηρηθεί η ισορροπία μεταξύ των κατηγοριών στο train/test split.

## 5.4 Εκπαίδευση Μοντέλων

Η δημιουργία του project στο Annif περιλάμβανε την αρχικοποίηση της δομής φακέλων και τη δημιουργία του αρχείου ρυθμίσεων project.cfg. Το αρχείο RDF (cve-project.rdf) της οντολογίας φορτώθηκε στον φάκελο του project και χρησιμοποιήθηκε ως λεξιλόγιο (vocabulary) για τη θεματική ευρετηρίαση.

Η εκπαίδευση πραγματοποιήθηκε με χρήση διαφορετικών backends που υποστηρίζει το Annif. Για κάθε backend δημιουργήθηκαν ξεχωριστά scripts εκπαίδευσης, τα οποία διαχειρίζονταν τη διαγραφή προηγούμενων εκπαιδευμένων μοντέλων, τη φόρτωση νέων δεδομένων και την αποθήκευση των αποτελεσμάτων. Οι παράμετροι εκπαίδευσης ρυθμίστηκαν ανάλογα με τον αλγόριθμο (π.χ. learning rate, epoch, max depth).

Η διαδικασία περιλάμβανε εκπαίδευση μεμονωμένων μοντέλων (π.χ. Omikuji, fastText) καθώς και συνδυαστικών (nn_ensemble), με στόχο τη βελτίωση της απόδοσης μέσω meta-learning. Ιδιαίτερη προσοχή δόθηκε στην αποφυγή overfitting μέσω stratified split και fine-tuning των υπερπαραμέτρων.

## 5.5 Διαδικασία Αξιολόγησης

Η αξιολόγηση των μοντέλων πραγματοποιήθηκε με την ανάπτυξη Python scripts που υλοποιούν μαζικές προβλέψεις (batch predictions) αξιοποιώντας παραλληλισμό (multiprocessing) για μείωση του χρόνου εκτέλεσης. Υπολογίστηκαν οι μετρικές Precision, Recall, F1-score, MAP και nDCG, οι οποίες θεωρούνται καθιερωμένα μέτρα για multi-label classification. Τα αποτελέσματα εξήχθησαν σε αρχεία Excel (.xlsx) για συστηματική καταγραφή και σύγκριση, τα οποία περιλάμβαναν τις περιγραφές, τις πραγματικές και τις προβλεπόμενες ετικέτες, καθώς και τις τιμές των μετρικών και το χρησιμοποιούμενο TOP_K (1, 2, 3 και 5).

Κατά τη διάρκεια της ανάπτυξης, εντοπίστηκαν και αντιμετωπίστηκαν διάφορα προβλήματα. Ένα συχνό σφάλμα ήταν το UnicodeDecodeError, το οποίο επιλύθηκε με τον ορισμό encoding='utf-8' στις κλήσεις subprocess και την ανάγνωση αρχείων. Επιπλέον, διορθώθηκαν περιπτώσεις όπου παρέμεναν προσωρινά αρχεία vectorizer από προηγούμενες εκτελέσεις, προκαλώντας σφάλματα FileExistsError. Η βελτιστοποίηση των παραμέτρων έγινε επαναληπτικά, με δοκιμές διαφορετικών τιμών και σύγκριση της απόδοσης.

Η διαδικασία αυτοματοποιήθηκε μέσω scripts που:

- Καθαρίζουν προηγούμενα μοντέλα και προσωρινά αρχεία.
- Εκτελούν εκπαίδευση.
- Υλοποιούν παράλληλη πρόβλεψη και υπολογισμό μετρικών.
- Παράγουν αναλυτικό αρχείο .xlsx με αποτελέσματα και στατιστικά.

Παρακάτω παρουσιάζεται η μεθοδολογία σε μορφή διαγράμματος.

*Πίνακας 1 Μεθοδολογία εργασίας*

| |
|---|
| **Συλλογή Δεδομένων (CVE Descriptions)** |
| ↓ |
| **Προεπεξεργασία Κειμένων (Καθαρισμός, Tokenization, Stopwords Removal, Stemming)** |
| ↓ |
| **Αντιστοίχιση με SKOS Οντολογία (CWE URIs)** |
| ↓ |
| **Εκπαίδευση Μοντέλων Μηχανικής Μάθησης (Omikuji, FastText, NN Ensemble)** |
| ↓ |
| **Αξιολόγηση Αποτελεσμάτων (Precision, Recall, F1, MAP, nDCG)** |
| ↓ |
| **Συμπεράσματα & Μελλοντικές Βελτιώσεις** |

Η μεθοδολογία που ακολουθήθηκε επέτρεψε την ανάπτυξη ενός συστήματος αυτόματης θεματικής ευρετηρίασης στον τομέα της κυβερνοασφάλειας, με βάση το Annif και SKOS οντολογίες. Τα αποτελέσματα έδειξαν ικανοποιητική ακρίβεια, με δυνατότητα περαιτέρω βελτίωσης μέσω εμπλουτισμού του συνόλου δεδομένων, βελτίωσης της ποιότητας των περιγραφών και πειραματισμού με επιπλέον backends και συνδυαστικά μοντέλα.

Η ανάπτυξη και υλοποίηση της παρούσας διπλωματικής εργασίας υποστηρίχθηκε από τη χρήση της πλατφόρμας GitLab, η οποία αποτέλεσε βασικό εργαλείο για τη διαχείριση του κώδικα, των δεδομένων και της συνολικής εξέλιξης του έργου. Το GitLab προσφέρει ένα ολοκληρωμένο περιβάλλον για έλεγχο εκδόσεων (version control) μέσω του συστήματος Git, καθώς και λειτουργικότητες που σχετίζονται με τη συνεργασία, την παρακολούθηση αλλαγών και την ασφαλή αποθήκευση αρχείων.

Η χρήση του GitLab επέτρεψε την παρακολούθηση των αλλαγών στον κώδικα και την τεκμηρίωση της εξέλιξης της εργασίας μέσω commit messages, ενώ παράλληλα διασφάλισε την ύπαρξη εφεδρικών αντιγράφων (backup) σε περίπτωση τεχνικών προβλημάτων. Επιπλέον, η δυνατότητα συγχρονισμού με το PyCharm επέτρεψε την απευθείας σύνδεση του περιβάλλοντος ανάπτυξης με το αποθετήριο, διευκολύνοντας την οργάνωση και τη συνέπεια του κώδικα.

Για λόγους πληρότητας, στο Παράρτημα Α παρουσιάζεται ενδεικτικά η δομή του αποθετηρίου στο GitLab, με τα κυριότερα αρχεία και φακέλους που χρησιμοποιήθηκαν. Η επιλογή του GitLab ως εργαλείου διαχείρισης κώδικα συνέβαλε καθοριστικά στη διασφάλιση της αναπαραγωγιμότητας και της διαφάνειας της μεθοδολογίας που ακολουθήθηκε.

## 5.6 Περιορισμοί και Προτάσεις

Η μεθοδολογία που ακολουθήθηκε επέτρεψε την ανάπτυξη ενός συστήματος αυτόματης θεματικής ευρετηρίασης στον τομέα της κυβερνοασφάλειας. Ωστόσο, υπάρχουν περιορισμοί που σχετίζονται με την ποιότητα των περιγραφών στο CVE dataset, την ανισορροπία μεταξύ κατηγοριών και τις

δυνατότητες παραμετροποίησης των backends. Μελλοντικές βελτιώσεις θα μπορούσαν να περιλαμβάνουν:

- Εμπλουτισμό του dataset με πρόσθετες πηγές (π.χ. Scopus, τεχνικές αναφορές).
- Πειραματισμό με πιο σύνθετα νευρωνικά μοντέλα και deep learning backends.
- Ενσωμάτωση semi-supervised και active learning τεχνικών για καλύτερη αξιοποίηση δεδομένων.
- Διερεύνηση cross-lingual μοντέλων για πολυγλωσσική θεματική ευρετηρίαση.

## 5.7 Επίλογος

Η μεθοδολογία που παρουσιάστηκε στο παρόν κεφάλαιο αποτέλεσε το θεμέλιο για την υλοποίηση της έρευνας. Μέσα από την οργανωμένη διαδικασία εγκατάστασης, συλλογής και προετοιμασίας δεδομένων, εκπαίδευσης μοντέλων και αξιολόγησης αποτελεσμάτων, διασφαλίστηκε η αναπαραγωγιμότητα και η επιστημονική εγκυρότητα του πειράματος. Η χρήση του Annif σε συνδυασμό με οντολογίες SKOS και πρότυπα όπως τα CVE και CWE επέτρεψε τη δομημένη θεματική ευρετηρίαση στον τομέα της κυβερνοασφάλειας, ενισχύοντας τη σημασιολογική οργάνωση και την ακρίβεια της ανάκτησης πληροφορίας. Επιπλέον, η αξιοποίηση εργαλείων ανάπτυξης και συνεργασίας, όπως το PyCharm και το GitLab, διευκόλυνε τη διαχείριση του κώδικα και των δεδομένων, ενώ παράλληλα ενίσχυσε τη διαφάνεια και την αναπαραγωγιμότητα της ερευνητικής διαδικασίας.

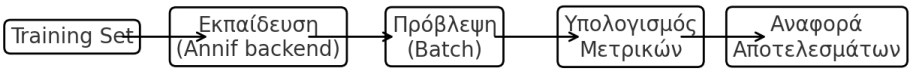
Συνολικά, η μεθοδολογία που ακολουθήθηκε επιτρέπει την εφαρμογή της προτεινόμενης προσέγγισης σε πραγματικά σενάρια και θέτει τις βάσεις για περαιτέρω έρευνα και βελτίωση σε συναφή πεδία.

# Κεφάλαιο 6ο: Αποτελέσματα

## 6.1 Εισαγωγή

Το παρόν κεφάλαιο παρουσιάζει τα αποτελέσματα της εκπαίδευσης και αξιολόγησης των μοντέλων μηχανικής μάθησης που υλοποιήθηκαν μέσω του εργαλείου Annif. Στο πλαίσιο της παρούσας εργασίας πραγματοποιήθηκε εκτεταμένη αξιολόγηση διαφορετικών backends του Annif, με χρήση συνόλων δοκιμών (test sets) διαφόρων μεγεθών και διαφορετικών στρατηγικών εκπαίδευσης.

Η αξιολόγηση βασίστηκε σε μετρικές όπως Precision, Recall, F1-score, MAP και nDCG, οι οποίες είναι ιδιαίτερα σημαντικές για προβλήματα πολυετικετικής ταξινόμησης (multi-label classification) για διάφορες τιμές της παραμέτρου TOP_K (1,2,3 και 5). Η ανάλυση των αποτελεσμάτων επιτρέπει τη σύγκριση των επιμέρους αλγορίθμων και την κατανόηση των πλεονεκτημάτων και μειονεκτημάτων τους.




*Σχήμα 30 Annif: Training → Prediction → Evaluation*

## 6.2 Αποτελέσματα ανά backend

### 6.2.1 Omikuji

Ο αλγόριθμος Omikuji, σχεδιασμένος για extreme multi-label classification, επέδειξε υψηλή απόδοση στην ανάκτηση κατηγοριών CWE με έμφαση στη μεγιστοποίηση του Recall. Στα πειράματα, ο Omikuji παρουσίασε υψηλές τιμές Recall, γεγονός που σημαίνει ότι εντόπιζε την πλειοψηφία των σχετικών ετικετών, αν και η Precision ήταν χαμηλότερη λόγω αυξημένων False Positives.

### 6.2.2 fastText

Ο αλγόριθμος fastText παρουσίασε ισορροπημένη συμπεριφορά μεταξύ Precision και Recall. Χάρη στη χρήση subword embeddings, μπόρεσε να γενικεύσει καλύτερα σε σπάνιους ή νέους όρους, κάτι που είναι ιδιαίτερα σημαντικό στον χώρο της κυβερνοασφάλειας όπου συχνά εμφανίζονται νέες ευπάθειες με καινοφανείς όρους. Η ταχύτητα εκπαίδευσης και πρόβλεψης αποτέλεσε επίσης σημαντικό πλεονέκτημα.

### 6.2.3 TF-IDF

Ο αλγόριθμος TF-IDF χρησιμοποιήθηκε ως baseline για τη σύγκριση των πιο σύνθετων αλγορίθμων. Παρότι είναι απλούστερο μοντέλο, προσέφερε ικανοποιητική ακρίβεια σε περιπτώσεις όπου οι όροι-κλειδιά ήταν σαφώς διακριτοί. Ωστόσο, η έλλειψη σημασιολογικής κατανόησης περιόρισε την απόδοσή του σε πιο σύνθετα κείμενα.

### 6.2.4 NN Ensemble

Ο αλγόριθμος NN Ensemble, που συνδυάζει τα αποτελέσματα του Omikuji και του fastText, εμφάνισε την καλύτερη συνολική απόδοση. Ο συνδυασμός των δύο αλγορίθμων επέτρεψε την εκμετάλλευση των πλεονεκτημάτων τους: ο Omikuji παρείχε υψηλή Recall, ενώ ο fastText υψηλότερη Precision. Ως αποτέλεσμα, το NN Ensemble πέτυχε βελτιωμένες τιμές σε μετρικές όπως MAP και nDCG, οι οποίες δίνουν έμφαση στη σωστή κατάταξη των ετικετών.

Ένας σημαντικός παράγοντας στην αξιολόγηση των εκπαιδευμένων μοντέλων ήταν η χρήση προσαρμοσμένων συνόλων εκπαίδευσης, η οποία είχε μεγάλη επίδραση στις μετρικές.

- Custom TOP 30 CWE – 1000 Descriptions per CWE: Βελτίωση όλων των μετρικών (π.χ. TOP_K=1, Precision=0.6867) σε σχέση με τα baseline μοντέλα.
- Custom TOP 10 CWE – 4000 Descriptions per CWE: Σημαντική αύξηση απόδοσης (TOP_K=1, Precision=0.8550, Recall=0.8550, F1=0.8550), υποδεικνύοντας ότι η εστίαση σε λιγότερες κατηγορίες με μεγαλύτερο αριθμό παραδειγμάτων ανά κατηγορία βελτιώνει δραστικά την ακρίβεια.
- Full Test 3964 entries: Ο Omikuji και το fastText έδωσαν υψηλές επιδόσεις, με τον nn_ensemble να φτάνει σε F1-score=0.7279.

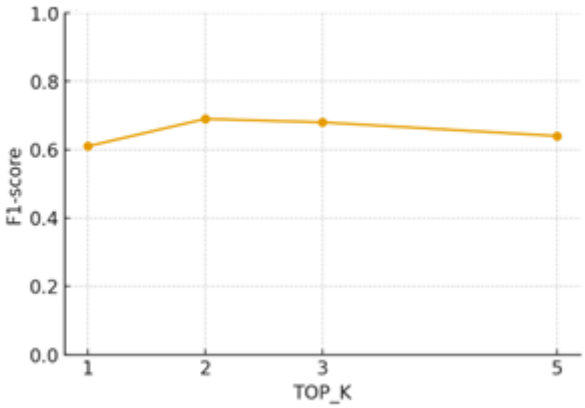
*Πίνακας 1 Αξιολόγηση Αλγορίθμων Annif - Custom Training (Top 10 CWE)*

| Metric | omikuji | fasttext | nn-ensemble (omikuji + fasttext) |
|---|---|---|---|
| TOP_K | 2 | 2 | 2 |
| Precision | 0.4717 | 0.4564 | 0.5948 |
| Recall | 0.9432 | 0.9127 | 0.9377 |
| F1-score | 0.6289 | 0.6085 | 0.7279 |
| MAP | 0.9067 | 0.864 | 0.9039 |
| nDCG | 0.9162 | 0.8768 | 0.9127 |
| Παράμετροι εκπαίδευσης | hyper-parameters n_trees=3, min_branch_size=100, max_depth=20, hinge_loss (c=1.0, eps=0.1), clustering (k=2, eps=0.0001) | dim=200, lr=0.3, epoch=10, loss=hs, minn=2, maxn=5, wordNgrams=2, minCount=2, threads=4 | nodes=100, dropout=0.2, epochs=10, learn-epochs=1, optimizer=adam |

Ο παραπάνω πίνακας περιέχει τις τιμές των μετρικών για TOP_K=2 που προέκυψαν από την αξιολόγηση των αλγορίθμων Annif με Custom Training (Top 10 CWE). Η εκπαίδευση έγινε με 4000 περιγραφές ανά CWE και η αξιολόγηση σε 3964 test εγγραφές.

### 6.3 Συζήτηση για TOP_K

Η επιλογή του πλήθους των προτεινόμενων ετικετών (TOP_K) είχε σημαντική επίδραση στην απόδοση των μοντέλων. Για μικρότερα K (π.χ. TOP_1), η Precision ήταν υψηλότερη καθώς το μοντέλο πρότεινε μόνο την πιο πιθανή ετικέτα. Αντίθετα, για μεγαλύτερες τιμές (π.χ. TOP_5), αυξανόταν η Recall αλλά μειωνόταν η Precision λόγω περισσότερων λανθασμένων προτάσεων. Η καλύτερη ισορροπία παρατηρήθηκε στις περιπτώσεις TOP_2 και TOP_3, όπου οι τιμές F1, MAP και nDCG παρουσίασαν βελτιστοποιημένα αποτελέσματα.




*Σχήμα 31 Μεταβολή F1-score ανάλογα με το TOP_K*

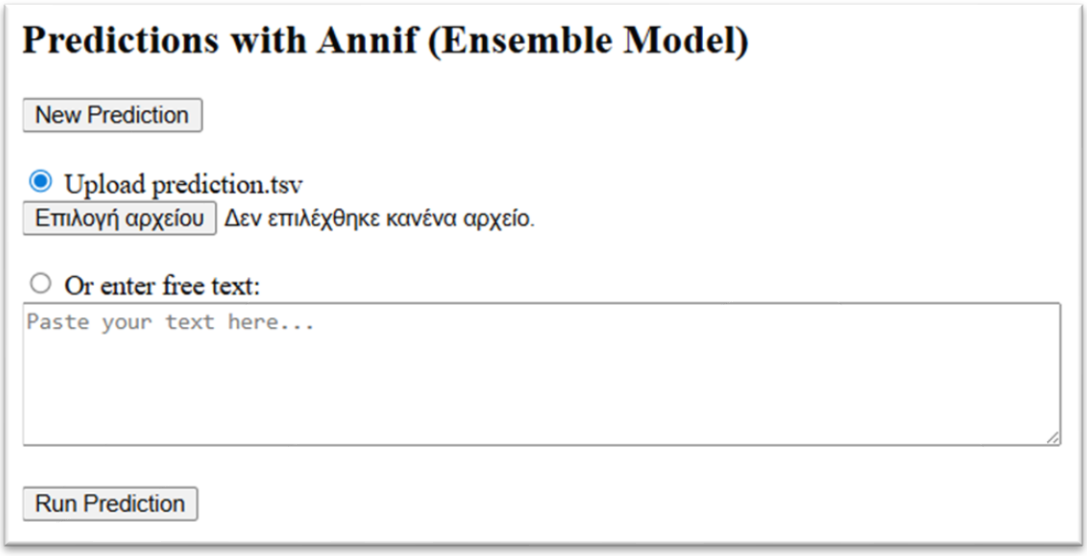
Αξίζει να σημειωθεί ότι για πειραματικούς λόγους (εγκυρότητας της εκπαίδευσης) χρησιμοποιήθηκε επίσης το `train.tsv` σε αξιολόγηση εκπαιδευμένου μοντέλου, εκτοξεύοντας τις μετρικές (MAP=0.9826, nDCG=0.9858), γεγονός που αναμενόμενα οφείλεται στην ταύτιση δεδομένων εκπαίδευσης και αξιολόγησης.

Ερμηνεία των μετρικών και παρατηρήσεις:

- Η Precision αυξάνεται αισθητά όταν περιορίζεται το πλήθος των κατηγοριών και αυξάνεται το πλήθος παραδειγμάτων ανά κατηγορία, κάτι που αποδεικνύει την ισχυρή συσχέτιση ποσότητας δεδομένων ανά label με την απόδοση.
- Η Recall διατηρείται σε υψηλά επίπεδα (>0.9) για TOP_K=2 και πάνω σε σενάρια με πολλά training examples, υποδεικνύοντας ότι το μοντέλο αναγνωρίζει σχεδόν όλες τις σωστές κατηγορίες στις πρώτες θέσεις κατάταξης.
- Οι μετρικές MAP και nDCG είναι ιδιαίτερα υψηλές (>0.9) στα σενάρια με περιορισμένες κατηγορίες και μεγάλο πλήθος παραδειγμάτων, υποδεικνύοντας ότι η διάταξη των σωστών κατηγοριών στις προβλέψεις είναι βελτιστοποιημένη.
- Η σύγκριση των αποτελεσμάτων δείχνει ότι ο αλγόριθμος Omikuji υπερτερεί ελαφρώς του fastText σε επίπεδο ακρίβειας (Precision) και γενικής ισορροπίας μετρικών, ενώ η χρήση προσαρμοσμένων συνόλων εκπαίδευσης με περιορισμένο αριθμό κατηγοριών και αυξημένο πλήθος παραδειγμάτων ανά κατηγορία (π.χ. Custom TOP 10 CWE – 4000 περιγραφές) οδηγεί σε σημαντική βελτίωση της απόδοσης. Η μέθοδος nn_ensemble, που συνδυάζει τους Omikuji και fastText, παρουσιάζει συχνά την καλύτερη ισορροπία μεταξύ ακρίβειας και ανάκλησης.
- Ο παραδοσιακός TF-IDF backend επιβεβαίωσε ότι οι μέθοδοι που βασίζονται αποκλειστικά σε συχνότητες λέξεων υπολείπονται σημαντικά σε σύγχρονες μεθόδους μηχανικής μάθησης, ενώ οι μετρικές MAP και nDCG υποδεικνύουν ότι η σειρά κατάταξης των σχετικών κατηγοριών είναι ιδιαίτερα βελτιστοποιημένη σε σενάρια με πλούσια και ισορροπημένα δεδομένα εκπαίδευσης.

### 6.4 Ανάπτυξη Διαδικτυακού Γραφικού Περιβάλλοντος (Web UI)

Στα πλαίσια των στόχων της εργασίας, αναπτύχθηκε ένα διαδικτυακό γραφικό περιβάλλον (web-based user interface – Web UI) με σκοπό να διευκολύνει τη χρήση του εργαλείου Annif για την αυτόματη θεματική ευρετηρίαση και την εξαγωγή λέξεων-κλειδιών από περιγραφές κυβερνοευπαθειών. Το περιβάλλον αυτό προσφέρει έναν εύχρηστο τρόπο αλληλεπίδρασης με το σύστημα, αποφεύγοντας την ανάγκη χρήσης γραμμής εντολών, και καθιστά το εργαλείο προσβάσιμο και σε χρήστες χωρίς τεχνικό υπόβαθρο.




*Σχήμα 32 UI προβλέψεων με επιλογή αρχείου ή ελεύθερου κειμένου*

Η διεπαφή σχεδιάστηκε με γνώμονα τη φιλικότητα προς τον χρήστη και τη λειτουργικότητα, παρέχοντας τις ακόλουθες δυνατότητες:

- Επιλογή εισόδου: Ο χρήστης μπορεί είτε να εισάγει ελεύθερο κείμενο σε κατάλληλο πεδίο, είτε να ανεβάσει αρχείο σε μορφή .tsv, το οποίο περιέχει πολλαπλές περιγραφές ευπαθειών.
- Εκτέλεση πρόβλεψης: Η εφαρμογή ενεργοποιεί ένα Python script που αξιοποιεί το προεκπαιδευμένο μοντέλο του Annif για την πρόβλεψη των σχετικών εννοιών.
- Λήψη αποτελεσμάτων: Τα αποτελέσματα είναι διαθέσιμα προς λήψη σε αρχείο .tsv, διευκολύνοντας την αποθήκευση και περαιτέρω ανάλυσή τους.
- Προεπισκόπηση αποτελεσμάτων: Οι προβλέψεις εμφανίζονται άμεσα στο UI, συνοδευόμενες από το URI της έννοιας, την αντίστοιχη ετικέτα (label) και το confidence score, προσφέροντας μια καθαρή και εύληπτη αναπαράσταση των εξόδων του συστήματος.

Η υλοποίηση της εφαρμογής βασίστηκε στο Flask, ένα ελαφρύ Python web framework, ενώ το γραφικό περιβάλλον αναπτύχθηκε με HTML και CSS. Το backend είναι υπεύθυνο για:

- Την αποθήκευση προσωρινών αρχείων εισόδου,
- Την εκτέλεση scripts μέσω της μεθόδου subprocess.run,
- Την ανάγνωση των αποτελεσμάτων από αρχεία .tsv και τη μετατροπή τους σε μορφή κατάλληλη για προβολή στο περιβάλλον χρήστη.

Για την αποφυγή σφαλμάτων κωδικοποίησης κατά την ανάγνωση και εγγραφή των αρχείων, εφαρμόστηκε UTF-8 encoding σε όλα τα στάδια της διαδικασίας.

Η ανάπτυξη του Web UI καθιστά τη χρήση του Annif πιο προσβάσιμη και πρακτική, επιτρέποντας την αξιολόγηση και ενσωμάτωσή του σε πραγματικά σενάρια εφαρμογής. Ενδεικτικά, η διεπαφή αυτή θα μπορούσε να χρησιμοποιηθεί από ερευνητές, αναλυτές κυβερνοασφάλειας ή βιβλιοθηκονόμους για την αυτόματη ευρετηρίαση μεγάλων συλλογών περιγραφών ευπαθειών, χωρίς να απαιτείται εξειδικευμένη γνώση προγραμματισμού.


*Σχήμα 33 Αποτελέσματα πρόβλεψης*

## 6.5 Επίλογος

Το παρόν κεφάλαιο παρουσίασε τα αποτελέσματα της εκπαίδευσης και αξιολόγησης των αλγορίθμων που χρησιμοποιήθηκαν στο εργαλείο Annif, καθώς και την ανάπτυξη μιας φιλικής διεπαφής για τη διευκόλυνση της χρήσης του. Η συγκριτική ανάλυση των διαφορετικών backends (Omikuji, fastText, TF-IDF και NN Ensemble) ανέδειξε τα πλεονεκτήματα και μειονεκτήματα κάθε προσέγγισης, επιβεβαιώνοντας τη σημασία της επιλογής κατάλληλου αλγορίθμου και συνόλου δεδομένων ανάλογα με το πρόβλημα.

Ιδιαίτερη έμφαση δόθηκε στον ρόλο των παραμέτρων TOP_K, καθώς και στην επίδραση της ποσότητας και ποιότητας των δεδομένων εκπαίδευσης στις τελικές μετρικές απόδοσης. Η χρήση προσαρμοσμένων συνόλων εκπαίδευσης απέδειξε ότι η εστίαση σε ισορροπημένα και επαρκώς μεγάλα δείγματα ανά κατηγορία μπορεί να βελτιώσει σημαντικά την ακρίβεια και την αξιοπιστία του συστήματος.

Τέλος, η ανάπτυξη του διαδικτυακού γραφικού περιβάλλοντος κατέδειξε πώς η ερευνητική υλοποίηση μπορεί να ενσωματωθεί σε ένα πρακτικό εργαλείο, προσβάσιμο ακόμη και σε χρήστες χωρίς εξειδικευμένες τεχνικές γνώσεις. Με αυτόν τον τρόπο, η εργασία δεν περιορίζεται μόνο σε θεωρητικό και πειραματικό επίπεδο, αλλά προσφέρει και μια πρακτική εφαρμογή με δυνατότητες αξιοποίησης σε πραγματικά σενάρια στον τομέα της κυβερνοασφάλειας.

# Κεφάλαιο 7ο: Συζήτηση Αποτελεσμάτων και Συμπεράσματα

## 7.1 Συζήτηση Αποτελεσμάτων

Στο κεφάλαιο αυτό πραγματοποιείται η συνολική ανάλυση και ερμηνεία των αποτελεσμάτων που παρουσιάστηκαν στο προηγούμενο κεφάλαιο. Κάθε backend εμφάνισε διαφορετικά πλεονεκτήματα και αδυναμίες, ανάλογα με το μέγεθος και την πολυπλοκότητα των δεδομένων, καθώς και με τις απαιτήσεις της ταξινόμησης.

Ο αλγόριθμος Omikuji απέδειξε την καταλληλότητά του σε προβλήματα extreme multi-label classification, επιτυγχάνοντας υψηλές τιμές Recall. Ωστόσο, η αυξημένη ανάκληση συνοδεύτηκε από χαμηλότερη Precision, γεγονός που σημαίνει ότι το μοντέλο εντόπιζε μεγάλο αριθμό σωστών ετικετών αλλά ταυτόχρονα παρήγαγε και περισσότερα False Positives.

Ο fastText, με την αξιοποίηση subword embeddings, προσέφερε καλύτερη ισορροπία μεταξύ Precision και Recall, ενώ η ταχύτητά του στην εκπαίδευση και την πρόβλεψη τον καθιστά ιδιαίτερα αποδοτικό για πραγματικές εφαρμογές. Η ικανότητά του να γενικεύει σε σπάνιες ή νέες λέξεις είναι κρίσιμη σε ένα δυναμικό περιβάλλον όπως η κυβερνοασφάλεια.

Ο TF-IDF, αν και απλούστερος αλγόριθμος, λειτούργησε ως ισχυρή baseline μέθοδος. Οι τιμές του ήταν ικανοποιητικές σε περιπτώσεις όπου οι όροι-κλειδιά ήταν διακριτοί, αλλά δεν κατάφερε να αποδώσει σε περισσότερο σύνθετα ή σημασιολογικά φορτισμένα κείμενα.

Το NN Ensemble συνδύασε τα αποτελέσματα των Omikuji και fastText, επιτυγχάνοντας βελτιωμένες τιμές σε μετρικές όπως MAP και nDCG. Ο συνδυασμός των πλεονεκτημάτων των δύο μοντέλων ανέδειξε την αξία των υβριδικών προσεγγίσεων, οι οποίες μπορούν να εκμεταλλευτούν διαφορετικά χαρακτηριστικά των δεδομένων.

Συνολικά, η επιλογή του κατάλληλου μοντέλου εξαρτάται από τις απαιτήσεις της εκάστοτε εφαρμογής. Σε περιβάλλοντα όπου η ανάκληση είναι κρίσιμη, όπως η πρόληψη κυβερνοαπειλών, ο Omikuji μπορεί να είναι προτιμότερος. Αντίθετα, όταν η ακρίβεια είναι πιο σημαντική, ο fastText αποδίδει καλύτερα. Το NN Ensemble αποτελεί μια συμβιβαστική λύση που μεγιστοποιεί τη συνολική απόδοση.

Η αξιολόγηση της απόδοσης του συστήματος αυτόματης θεματικής ευρετηρίασης πραγματοποιήθηκε τόσο με ποσοτικές μεθόδους όσο και με ποιοτική ανάλυση. Οι ποσοτικές μετρήσεις βασίστηκαν στις καθιερωμένες μετρικές Precision, Recall, F1-score, Mean Average Precision (MAP) και Normalized Discounted Cumulative Gain (nDCG), ενώ η ποιοτική αξιολόγηση στηρίχθηκε στη χειροκίνητη επισκόπηση προβλέψεων σε επιλεγμένα παραδείγματα κειμένων ευπαθειών.

Τα αποτελέσματα υπήρξαν ιδιαίτερα ενθαρρυντικά. Οι υψηλές τιμές MAP και nDCG καταδεικνύουν ότι οι σημαντικότερες (θεματικά συναφείς) ετικέτες τοποθετούνται ψηλά στη λίστα των προβλέψεων, γεγονός κρίσιμο για πραγματικά σενάρια χρήσης όπου ο χρήστης εστιάζει κυρίως στις κορυφαίες προτάσεις.

Η ακρίβεια του συστήματος αποδείχθηκε ικανοποιητική ακόμη και για κείμενα που περιέχουν σύνθετες ή τεχνικές πληροφορίες, υπό την προϋπόθεση ότι η κατηγορία εκπροσωπείται επαρκώς στο σύνολο εκπαίδευσης.

Η ποιότητα των παραγόμενων λέξεων-κλειδιών αποδείχθηκε ότι επηρεάζεται άμεσα από τους εξής παράγοντες:

- Την πληρότητα και την ακρίβεια του λεξιλογίου SKOS. Η ύπαρξη πλούσιων prefLabel, altLabel και definition ενισχύει τη σημασιολογική αντιστοίχιση.
- Η πολυπλοκότητα και ποικιλομορφία των κειμένων εισόδου: Μακροσκελή ή αμφίσημα κείμενα απαιτούν πιο εξελιγμένα μοντέλα.
- Η κατανομή των παραδειγμάτων στις κατηγορίες κατά την εκπαίδευση: Οι κατηγορίες με περισσότερα και πιο αντιπροσωπευτικά παραδείγματα παράγουν καλύτερες προβλέψεις.

Ιδιαίτερη βελτίωση στην απόδοση παρατηρήθηκε όταν χρησιμοποιήθηκε ένα custom balanced dataset με τα Top 10 CWE, με 4000 παραδείγματα ανά κατηγορία. Η εξισορρόπηση αυτή επέτρεψε στα μοντέλα να μάθουν πιο αντιπροσωπευτικά χαρακτηριστικά για κάθε ετικέτα.

Συμπερασματικά το Annif προσφέρει σημαντικά πλεονεκτήματα όπως:

- ευελιξία στην επιλογή και συνδυασμό backends (ensemble).
- υποστήριξη SKOS λεξιλογίων
- CLI και REST API, που καθιστούν εύκολη την ενσωμάτωση του Annif σε ροές εργασίας, pipelines και UI εφαρμογές.

Ωστόσο, σε περιπτώσεις με πολύ μικρά σύνολα εκπαίδευσης ή ακραία ανισορροπία κατηγοριών, η απόδοση του Annif δεν διαφέρει σημαντικά από απλούστερες μεθόδους. Επίσης η απόδοση εξαρτάται σημαντικά από την ποιότητα του λεξιλογίου. Λάθη στο RDF, απουσία εναλλακτικών όρων ή φτωχή τεκμηρίωση σε ετικέτες οδηγούν σε ελλιπείς ή λανθασμένες προβλέψεις.

Κατά την υλοποίηση και αξιολόγηση της εφαρμογής, εντοπίστηκαν συγκεκριμένοι περιορισμοί:

- Στο backend Omikuji, πολλές εσωτερικές παράμετροι δεν είναι διαθέσιμες για τροποποίηση μέσω του αρχείου ρυθμίσεων του έργου.
- Εξάρτηση από την ποιότητα του λεξιλογίου: Λάθη ή ελλείψεις στο SKOS RDF οδηγούν σε αντίστοιχες ελλείψεις στις προβλέψεις.
- Ανισορροπία δεδομένων εκπαίδευσης: Υπάρχουν αρκετές κατηγορίες με λίγα παραδείγματα.

Η εργασία αυτή μπορεί να αποτελέσει τη βάση για περαιτέρω επεκτάσεις, μεταξύ των οποίων:

- Εμπλουτισμός του λεξιλογίου SKOS με επιπλέον altLabel, synonyms, definitions και context.
- Χρήση τεχνικών data augmentation (π.χ. paraphrasing, synonym replacement) για ενίσχυση των υποεκπροσωπούμενων κατηγοριών.
- Διερεύνηση υβριδικών προσεγγίσεων, όπου το Annif συνδυάζεται με μεγάλα γλωσσικά μοντέλα (LLMs), ώστε να ενισχυθεί η σημασιολογική κατανόηση και η κάλυψη σε σπάνιες περιπτώσεις.

Το προτεινόμενο σύστημα απέδειξε ότι ένα εργαλείο όπως το Annif, σε συνδυασμό με καλά δομημένα λεξιλόγια και ισορροπημένα σύνολα δεδομένων, μπορεί να επιτύχει ποιοτική και ακριβή αυτόματη θεματική ευρετηρίαση στον τομέα της κυβερνοασφάλειας. Η προσθήκη διαδραστικού UI ενισχύει την αξιοποίηση του συστήματος από τελικούς χρήστες χωρίς τεχνικές γνώσεις, καθιστώντας την παρούσα λύση μια ελκυστική πρόταση για εφαρμογές όπως βιβλιοθηκονομικά συστήματα, διαχείριση ευπαθειών και θεματική κατηγοριοποίηση κάθε συμβάντος που επηρεάζει ή διαταράσσει τη φυσιολογική λειτουργία ενός πληροφοριακού συστήματος.

## 7.2 Συμπεράσματα

Από την ανάλυση προκύπτει ότι το Annif μπορεί να αποτελέσει ένα ιδιαίτερα χρήσιμο εργαλείο για την αυτόματη θεματική ευρετηρίαση στον τομέα της κυβερνοασφάλειας. Η ενσωμάτωση SKOS οντολογιών, όπως το CWE, διασφαλίζει ότι οι προβλέψεις του συστήματος βασίζονται σε τυποποιημένα και διεθνώς αναγνωρισμένα λεξιλόγια. Αυτό ενισχύει τη διαλειτουργικότητα και επιτρέπει την εύκολη ανταλλαγή και αξιοποίηση των αποτελεσμάτων μεταξύ διαφορετικών συστημάτων και οργανισμών.

Η χρήση τεχνικών μηχανικής μάθησης επέτρεψε την αποτελεσματική κατηγοριοποίηση περιγραφών ευπαθειών CVE σε κατηγορίες CWE. Η προσέγγιση της πολυετικετικής ταξινόμησης απέδειξε την καταλληλότητά της για το πεδίο της κυβερνοασφάλειας, καθώς κάθε ευπάθεια μπορεί να σχετίζεται με περισσότερες από μία κατηγορίες.

Η μεθοδολογία που εφαρμόστηκε είναι αναπαραγώγιμη και επεκτάσιμη, κάτι που σημαίνει ότι μπορεί να εφαρμοστεί και σε άλλα πεδία όπου απαιτείται αυτόματη θεματική ευρετηρίαση μεγάλων συνόλων δεδομένων. Η συμβολή της εργασίας εντοπίζεται τόσο στο θεωρητικό επίπεδο, μέσω της σύνδεσης οντολογιών και αλγορίθμων, όσο και στο πρακτικό, με την ανάπτυξη ενός λειτουργικού συστήματος ταξινόμησης.

## 7.3 Μελλοντικές Κατευθύνσεις

Η παρούσα εργασία άνοιξε τον δρόμο για περαιτέρω έρευνα και ανάπτυξη στον τομέα της αυτόματης θεματικής ευρετηρίασης στην κυβερνοασφάλεια. Μελλοντικές βελτιώσεις μπορούν να εστιάσουν στα εξής:

- Εμπλουτισμός του dataset με πρόσθετες πηγές, όπως ακαδημαϊκές βάσεις δεδομένων (π.χ. Scopus) ή τεχνικές αναφορές.
- Πειραματισμός με πιο σύνθετα μοντέλα βαθιάς μάθησης, όπως BERT ή άλλα transformer-based μοντέλα.
- Ανάπτυξη πολυγλωσσικών μοντέλων για cross-lingual θεματική ευρετηρίαση.
- Ενσωμάτωση semi-supervised και active learning τεχνικών για καλύτερη αξιοποίηση μη ετικεταρισμένων δεδομένων.
- Διερεύνηση explainable AI μεθόδων για μεγαλύτερη ερμηνευσιμότητα των προβλέψεων.

Η συνεισφορά της εργασίας δείχνει ότι η σύνδεση μεταξύ οντολογιών SKOS και μοντέλων μηχανικής μάθησης είναι μια αποτελεσματική στρατηγική, αλλά ταυτόχρονα αναδεικνύει την ανάγκη για περαιτέρω έρευνα ώστε να αντιμετωπιστούν προκλήσεις όπως η ανισορροπία δεδομένων, η σημασιολογική κατανόηση και η προσαρμογή σε νέες γλώσσες ή πεδία εφαρμογής.

# Βιβλιογραφία


[1] European Union Agency for Cybersecurity., *ENISA threat landscape 2024: July 2023 to June 2024.* LU: Publications Office, 2024. Accessed: Aug. 13, 2025. [Online]. Available: https://data.europa.eu/doi/10.2824/0710888

[2] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, vol. 1, no. 1, p. 80, 2011.

[3] S. Ghafur, S. Kristensen, K. Honeyford, G. Martin, A. Darzi, and P. Aylin, "A retrospective impact analysis of the WannaCry cyberattack on the NHS," *NPJ digital medicine*, vol. 2, no. 1, p. 98, 2019.

[4] R. Alkhadra, J. Abuzaid, M. AlShammari, and N. Mohammad, "Solar winds hack: In-depth analysis and countermeasures," in *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, 2021, pp. 1–7. Accessed: Aug. 19, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9579611/

[5] A. Miles and S. Bechhofer, "SKOS simple knowledge organization system reference," 2009, Accessed: Aug. 07, 2025. [Online]. Available: https://research.manchester.ac.uk/en/publications/skos-simple-knowledge-organization-system-reference

[6] "CWE - Common Weakness Enumeration." Accessed: Aug. 13, 2025. [Online]. Available: https://cwe.mitre.org/

[7] "MITRE ATT&CK®." Accessed: Aug. 13, 2025. [Online]. Available: https://attack.mitre.org/

[8] C. D. Manning, *Introduction to information retrieval*. Syngress Publishing, 2008. Accessed: Aug. 19, 2025. [Online]. Available: http://diglib.globalcollege.edu.et:8080/xmlui/bitstream/handle/123456789/1096/Manning_introduction_to_information_retrieval.pdf?sequence=1&isAllowed=y

[9] C. N. Mooers, *Making information retrieval pay*. Zator Company, 1951.

[10] G. Salton and M. E. Lesk, "Computer Evaluation of Indexing and Text Processing," *J. ACM*, vol. 15, no. 1, pp. 8–36, Jan. 1968, doi: 10.1145/321439.321441.

[11] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[12] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[13] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[14] J. Ramos, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, New Jersey, USA, 2003, pp. 29–48. Accessed: Aug. 19, 2025. [Online]. Available: https://www.researchgate.net/profile/Farshad-Madani/post/In_information_retrieval_tf-idf_calculation_why_we_dont_divide_tf_by_the_length_of_the_related_document/attachment/59d6446679197b807799fae0/AS%3A448525403201536%401483948197307/download/Using+TF-IDF+to+Determine+Word+Relevance+in+Document+Queries.pdf

[15] R. Nogueira and K. Cho, "Passage Re-ranking with BERT," Apr. 14, 2020, *arXiv*: arXiv:1901.04085. doi: 10.48550/arXiv.1901.04085.

[16] P. Fothergill, "LANCASTER, FW Indexing and abstracting in theory and practice. London: Facet Publishing, 2003. ISBN 1-85604-482-3:\pounds 39.95. xix, 451 p," *Legal Information Management*, vol. 4, no. 2, pp. 147–147, 2004.

[17] J. May, "Broughton, V.(2015). Essential classification . London: Facet Publishing." 2017. Accessed: Aug. 06, 2025. [Online]. Available: https://www.lirgjournal.org.uk/index.php/lir/article/download/707/754

[18] P. Mayr and V. Petras, "Cross-concordances: terminology mapping and its effectiveness for information retrieval," 2008, *arXiv*. doi: 10.48550/ARXIV.0806.3765.

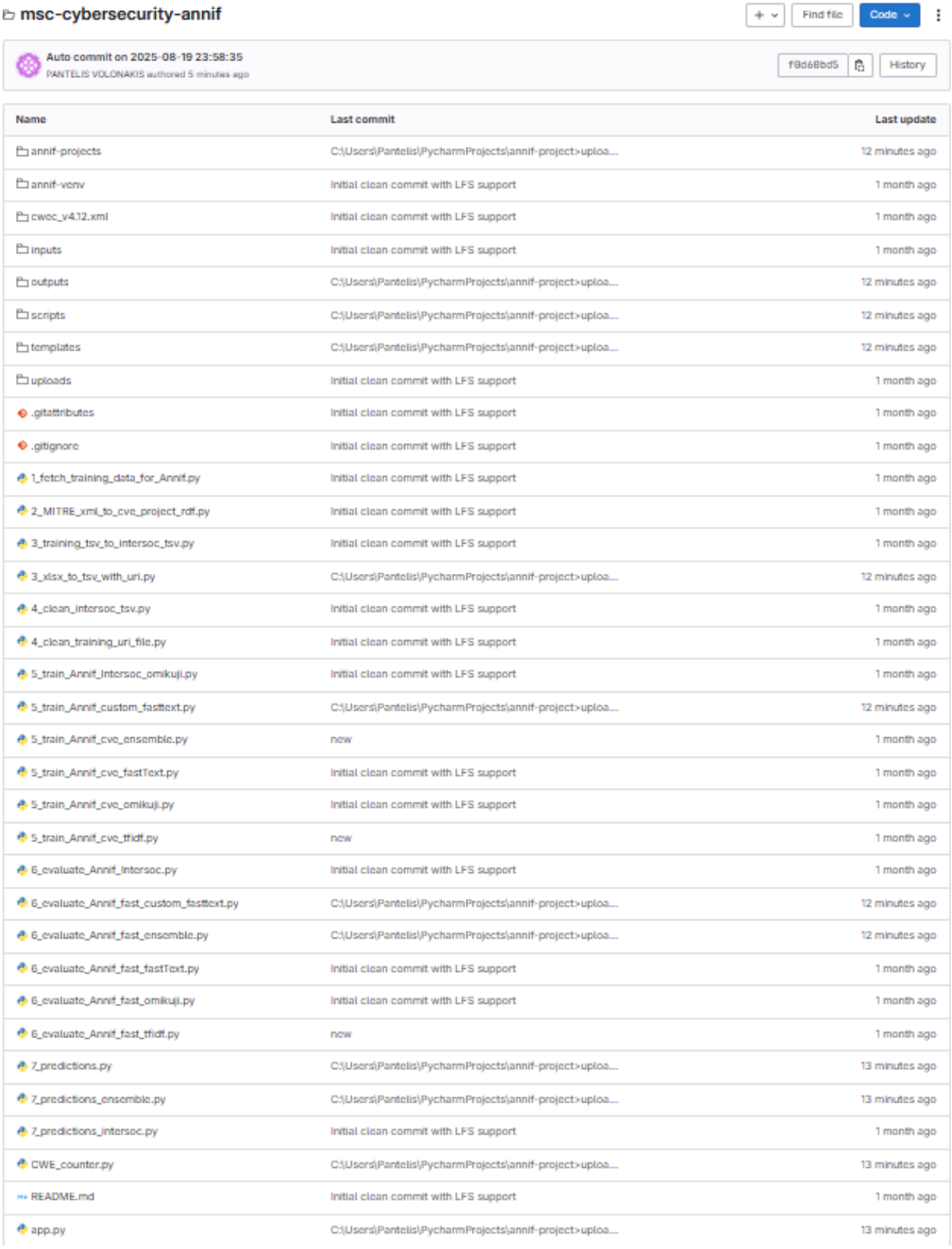
[19] O. Suominen, "Annif: DIY automated subject indexing using multiple algorithms," *LIBER Quarterly: The Journal of the Association of European Research Libraries*, vol. 29, no. 1, pp. 1–25, 2019.

[20] Y. Sure, S. Staab, and R. Studer, “Ontology Engineering Methodology,” in *Handbook on Ontologies*, S. Staab and R. Studer, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 135–152. doi: 10.1007/978-3-540-92673-3_6.
[21] T. Berners-Lee and J. Hendler, “Publishing on the semantic web,” *Nature*, vol. 410, no. 6832, pp. 1023–1024, 2001.
[22] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, “Mitre att&ck: Design and philosophy,” in *Technical report*, The MITRE Corporation, 2018. Accessed: Aug. 19, 2025. [Online]. Available: https://www.mitre.org/sites/default/files/2021-11/prs-19-01075-28-mitre-attack-design-and-philosophy.pdf
[23] N. Noy and D. L. McGuinness, “Ontology development 101,” *Knowledge Systems Laboratory, Stanford University*, vol. 2001, pp. 1–18, 2001.
[24] S. Primer, “SKOS Simple Knowledge Organization System Primer,” *World Wide Web Consortium*, vol. 18, 2009.
[25] M. Lei Zeng and L. Mai Chan, “Trends and issues in establishing interoperability among knowledge organization systems,” *J. Am. Soc. Inf. Sci.*, vol. 55, no. 5, pp. 377–395, Mar. 2004, doi: 10.1002/asi.10387.
[26] A. Isaac and E. Summers, “SKOS simple knowledge organization system primer,” *Working Group Note, W3C*, 2009, Accessed: Aug. 07, 2025. [Online]. Available: https://travesia.mcu.es/bitstream/10421/7465/1/skos.pdf
[27] P. Mayr, P. Mutschke, V. Petras, P. Schaer, and Y. Sure, “Applying Science Models for Search,” Jan. 09, 2011, *arXiv*: arXiv:1101.1639. doi: 10.48550/arXiv.1101.1639.
[28] G. Bueno-de-la-Fuente, “The Simple Knowledge Organization System (SKOS): a situation report for the HIVE Project,” 2008, Accessed: Aug. 07, 2025. [Online]. Available: https://e-archivo.uc3m.es/entities/publication/c07e5f5c-070f-4f56-b6fb-983533ed6529
[29] W. Stallings, “Effective cybersecurity: Understanding and using standards and best practices,” *(No Title)*, 2019, Accessed: Aug. 07, 2025. [Online]. Available: https://cir.nii.ac.jp/crid/1130000794369507200
[30] M. E. Whitman and H. J. Mattord, *Management of Information Security*. Cengage Learning, 2018. [Online]. Available: https://books.google.gr/books?id=TuNhEAAAQBAJ
[31] R. Von Solms and J. Van Niekerk, “From information security to cyber security,” *Computers & Security*, vol. 38, pp. 97–102, Oct. 2013, doi: 10.1016/j.cose.2013.04.004.
[32] “Von Solms και Van Niekerk - 2013 - From information security to cyber security.pdf.” Accessed: Sept. 01, 2025. [Online]. Available: https://profsandhu.com/cs6393_s20/Solms-Niekerk-2013.pdf
[33] C. Tankard, “Advanced persistent threats and how to monitor and deter them,” *Network security*, vol. 2011, no. 8, pp. 16–19, 2011.
[34] K. A. Scarfone and P. M. Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS),” National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-94, 2007. doi: 10.6028/NIST.SP.800-94.
[35] National Institute of Standards and Technology, “Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1,” National Institute of Standards and Technology, Gaithersburg, MD, NIST CSWP 04162018, Apr. 2018. doi: 10.6028/NIST.CSWP.04162018.
[36] P. Mell, K. Scarfone, and S. Romanosky, “A complete guide to the common vulnerability scoring system version 2.0,” in *Published by FIRST-forum of incident response and security teams*, 2007, p. 23. Accessed: Aug. 07, 2025. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=51198
[37] “CVE: Common Vulnerabilities and Exposures.” Accessed: Aug. 07, 2025. [Online]. Available: https://www.cve.org/
[38] C. Sabottke, O. Suciu, and T. Dumitraș, “Vulnerability disclosure in the age of social media: Exploiting twitter for predicting {Real-World} exploits,” in *24th USENIX security symposium (USENIX security 15)*, 2015, pp. 1041–1056. Accessed: Aug. 07, 2025. [Online]. Available: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/sabottke
[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. in Adaptive Computation and Machine Learning series. Cambridge, MA, USA: MIT Press, 2016. Accessed: Aug. 19, 2025. [Online]. Available: https://mitpress.mit.edu/9780262035613/deep-learning/

[40] A. Vaswani *et al.*, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017, Accessed: Aug. 19, 2025. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
[41] E. Cambria and B. White, “Jumping NLP curves: A review of natural language processing research,” *IEEE Computational intelligence magazine*, vol. 9, no. 2, pp. 48–57, 2014.
[42] D. Kelly and C. R. Sugimoto, “A systematic review of interactive information retrieval evaluation studies, 1967–2006,” *J Am Soc Inf Sci Tec*, vol. 64, no. 4, pp. 745–770, Apr. 2013, doi: 10.1002/asi.22799.
[43] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” Sept. 03, 2014, *arXiv*: arXiv:1408.5882. doi: 10.48550/arXiv.1408.5882.
[44] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186. Accessed: Aug. 19, 2025. [Online]. Available: https://aclanthology.org/N19-1423/?utm_campaign=The+Batch&utm_source=hs_email&utm_medium=email&_hsenc=p2ANqtz-_m9bbH_7ECE1h3lZ3D61TYg52rKpifVNjL4fvJ85uqggrXsWDBTB7YooFLJeNXHWqhvOyC
[45] Y. Liu *et al.*, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” July 26, 2019, *arXiv*: arXiv:1907.11692. doi: 10.48550/arXiv.1907.11692.
[46] T. Ruotsalo, G. Jacucci, P. Myllymäki, and S. Kaski, “Interactive intent modeling: information discovery beyond search,” *Commun. ACM*, vol. 58, no. 1, pp. 86–92, Jan. 2015, doi: 10.1145/2656334.
[47] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. in Adaptive Computation and Machine Learning series. Cambridge, MA, USA: MIT Press, 2018. Accessed: Sept. 01, 2025. [Online]. Available: https://mitpress.mit.edu/9780262039246/reinforcement-learning/
[48] R. Babbar and B. Schölkopf, “Data scarcity, robustness and extreme multi-label classification,” *Mach Learn*, vol. 108, no. 8–9, pp. 1329–1351, Sept. 2019, doi: 10.1007/s10994-019-05791-5.
[49] A. Joulin, M. Cissé, D. Grangier, and H. Jégou, “Efficient softmax approximation for GPUs,” in *International conference on machine learning*, PMLR, 2017, pp. 1302–1310. Accessed: Sept. 04, 2025. [Online]. Available: http://proceedings.mlr.press/v70/grave17a.html?ref=https://githubhelp.com

# Παράρτημα Α – Δομή GitLab Repository

Η δομή του αποθετηρίου GitLab που χρησιμοποιήθηκε για την υλοποίηση της εργασίας παρουσιάζεται ενδεικτικά παρακάτω:



msc-cybersecurity-annif

Find file | Code

Auto commit on 2025-08-19 23:58:35
PANTELIS VOLONAKIS authored 5 minutes ago

f8d68bd5 | History

| Name | Last commit | Last update |
|---|---|---|
| annif-projects | C:\Users\Pantelis\PycharmProjects\annif-project>uploa... | 12 minutes ago |
| annif-venv | Initial clean commit with LFS support | 1 month ago |
| cwec_v4.12.xml | Initial clean commit with LFS support | 1 month ago |
| inputs | Initial clean commit with LFS support | 1 month ago |
| outputs | C:\Users\Pantelis\PycharmProjects\annif-project>uploa... | 12 minutes ago |
| scripts | C:\Users\Pantelis\PycharmProjects\annif-project>uploa... | 12 minutes ago |
| templates | C:\Users\Pantelis\PycharmProjects\annif-project>uploa... | 12 minutes ago |
| uploads | Initial clean commit with LFS support | 1 month ago |
| .gitattributes | Initial clean commit with LFS support | 1 month ago |
| .gitignore | Initial clean commit with LFS support | 1 month ago |
| 1_fetch_training_data_for_Annif.py | Initial clean commit with LFS support | 1 month ago |
| 2_MITRE_xml_to_cve_project_rdf.py | Initial clean commit with LFS support | 1 month ago |
| 3_training_tsv_to_intersoc_tsv.py | Initial clean commit with LFS support | 1 month ago |
| 3_xlsx_to_tsv_with_uri.py | C:\Users\Pantelis\PycharmProjects\annif-project>uploa... | 12 minutes ago |
| 4_clean_intersoc_tsv.py | Initial clean commit with LFS support | 1 month ago |
| 4_clean_training_uri_file.py | Initial clean commit with LFS support | 1 month ago |
| 5_train_Annif_Intersoc_omikuji.py | Initial clean commit with LFS support | 1 month ago |
| 5_train_Annif_custom_fasttext.py | C:\Users\Pantelis\PycharmProjects\annif-project>uploa... | 12 minutes ago |
| 5_train_Annif_cve_ensemble.py | new | 1 month ago |
| 5_train_Annif_cve_fastText.py | Initial clean commit with LFS support | 1 month ago |
| 5_train_Annif_cve_omikuji.py | Initial clean commit with LFS support | 1 month ago |
| 5_train_Annif_cve_tfidf.py | new | 1 month ago |
| 6_evaluate_Annif_Intersoc.py | Initial clean commit with LFS support | 1 month ago |
| 6_evaluate_Annif_fast_custom_fasttext.py | C:\Users\Pantelis\PycharmProjects\annif-project>uploa... | 12 minutes ago |
| 6_evaluate_Annif_fast_ensemble.py | C:\Users\Pantelis\PycharmProjects\annif-project>uploa... | 12 minutes ago |
| 6_evaluate_Annif_fast_fastText.py | Initial clean commit with LFS support | 1 month ago |
| 6_evaluate_Annif_fast_omikuji.py | Initial clean commit with LFS support | 1 month ago |
| 6_evaluate_Annif_fast_tfidf.py | new | 1 month ago |
| 7_predictions.py | C:\Users\Pantelis\PycharmProjects\annif-project>uploa... | 13 minutes ago |
| 7_predictions_ensemble.py | C:\Users\Pantelis\PycharmProjects\annif-project>uploa... | 13 minutes ago |
| 7_predictions_intersoc.py | Initial clean commit with LFS support | 1 month ago |
| CWE_counter.py | C:\Users\Pantelis\PycharmProjects\annif-project>uploa... | 13 minutes ago |
| README.md | Initial clean commit with LFS support | 1 month ago |
| app.py | C:\Users\Pantelis\PycharmProjects\annif-project>uploa... | 13 minutes ago |

# Παράρτημα Β – Σύντομα αποσπάσματα κώδικα Python

Σε αυτό το παράρτημα παρουσιάζονται σύντομα αποσπάσματα (50 γραμμές ανά αρχείο) από τα Python scripts που χρησιμοποιήθηκαν. Τα πλήρη αρχεία βρίσκονται στο GitLab.

**1_fetch_training_data_for_Annif.py**

```
Εξαγωγή δεδομένων εκπαίδευσης μοντέλων.
import requests
import csv
import time
import pandas as pd
from datetime import datetime, timedelta
# Fixed API key directly in code (not recommended for production use)
API_KEY = "d80b6eb2-9d7a-419d-a4e6-************"
def fetch_cve_data_interval(start_date, end_date, results_per_page=1000, api_key=None):
    base_url = "https://services.nvd.nist.gov/rest/json/cves/2.0"
    start_index = 0
    all_items = []

    headers = {"apiKey": api_key} if api_key else {}

    while True:
        params = {
            "resultsPerPage": str(results_per_page),
            "startIndex": str(start_index),
            "pubStartDate": start_date,
            "pubEndDate": end_date
        }
        print(f"Fetching {start_date} to {end_date} at index {start_index}...")
        response = requests.get(base_url, headers=headers, params=params)

        if response.status_code != 200:
            print(f"Error fetching data at index {start_index}: {response.status_code}")
            print("Response:", response.text)
            break

        data = response.json()
        vulnerabilities = data.get("vulnerabilities", [])
        if not vulnerabilities:
            break

        all_items.extend(vulnerabilities)

        if len(vulnerabilities) < results_per_page:
            break

        start_index += results_per_page
        time.sleep(1.2)  # polite delay

    return {"vulnerabilities": all_items}

def extract_filtered_cve_info(cve_data, seen_descriptions):
    extracted_filtered = []
    for item in cve_data.get("vulnerabilities", []):
…
```

**2_MITRE_xml_to_cve_project_rdf.py**

```
Μετασχηματισμός xml αρχείου από MITRE, σε SKOS οντολογία.
import xml.etree.ElementTree as ET
from rdflib import Graph, Namespace, URIRef, Literal
from rdflib.namespace import RDF, SKOS

# Load XML
tree = ET.parse("cwec_v4.12.xml/cwec_v4.12.xml")
root = tree.getroot()

# Define namespaces
ns = {'cwe': 'http://cwe.mitre.org/cwe-7'}
CWE = Namespace("http://cwe.mitre.org/data/cwe#")

# Create RDF graph
g = Graph()
g.bind("skos", SKOS)
g.bind("cwe", CWE)

# Map CWE ID ➜ URI for broader relations
id_to_uri = {}

# Process CWE Weaknesses
for weakness in root.findall(".//cwe:Weakness", ns):
    cwe_id = weakness.get("ID")
    name = weakness.get("Name")
    desc_elem = weakness.find("cwe:Description", ns)
    description = desc_elem.text.strip() if desc_elem is not None and desc_elem.text else ""

    concept_uri = CWE[f"CWE-{cwe_id}"]
    id_to_uri[cwe_id] = concept_uri

    g.add((concept_uri, RDF.type, SKOS.Concept))
    g.add((concept_uri, SKOS.prefLabel, Literal(name, lang="en")))
    if description:
        g.add((concept_uri, SKOS.definition, Literal(description, lang="en")))

# Add broader relationships
for weakness in root.findall(".//cwe:Weakness", ns):
    child_id = weakness.get("ID")
    concept_uri = id_to_uri.get(child_id)
    rels = weakness.find("cwe:Related_Weaknesses", ns)
    if rels is not None:
        for rel in rels.findall("cwe:Related_Weakness", ns):
            if rel.get("Nature") == "ChildOf":
                parent_id = rel.get("CWE_ID")
                parent_uri = id_to_uri.get(parent_id)
                if parent_uri:
                    g.add((concept_uri, SKOS.broader, parent_uri))

# Save as RDF/XML
g.serialize(destination="cve-project.rdf", format="xml")

…
```

**3_xlsx_to_tsv_with_uri.py**

Μετατροπή δεδομένων μεταξύ μορφών (Excel/CSV/TSV) με ενσωμάτωση URIs.

```
import pandas as pd
from rdflib import Graph
import os

# === Input and output files ===
XLSX_FILE = "top10_cwe_4000_each.xlsx"
RDF_FILE = "cve-project.rdf"
OUTPUT_TSV = "training_uri.tsv"

# === Filtering parameters ===
MIN_COUNT = 5      # Minimum number of examples per URI to keep it
MAX_COUNT = 4100   # Maximum number of examples per URI

# === Load RDF vocabulary ===
print("Loading RDF vocabulary...")
g = Graph()
g.parse(RDF_FILE, format="xml")

# === Extract available CWE codes from RDF ===
cwe_uri_map = {}
for subj in g.subjects():
    if isinstance(subj, str) and "CWE-" in subj:
        cwe_code = subj.split("#")[-1]  # e.g. CWE-1037
        cwe_uri_map[cwe_code] = f"<{subj}>"

print(f"Loaded {len(cwe_uri_map)} CWE URIs from RDF.")

# === Load Excel data ===
print("Reading Excel file...")
df = pd.read_excel(XLSX_FILE)
df = df.dropna(subset=["Keywords", "Description"])
df = df.astype(str)

# === Expand rows: one row per keyword per description ===
expanded_rows = []
not_found = set()

for _, row in df.iterrows():
    description = row["Description"].strip()
    keywords = [kw.strip() for kw in row["Keywords"].split(";") if kw.strip()]

    for kw in keywords:
        uri = cwe_uri_map.get(kw)
        if uri:
            expanded_rows.append((uri, description))
        else:
            not_found.add(kw)

# === Convert to DataFrame ===
df_expanded = pd.DataFrame(expanded_rows, columns=["subject", "text"])

…
```

**4_clean_training_uri_file.py**

Προεπεξεργασία κειμένων (καθαρισμός, tokenization, stopwords, stemming/lemmatization).

```
import pandas as pd
from rdflib import Graph
import re

# === File settings ===
INPUT_TSV = "training_uri.tsv"
RDF_FILE = "cve-project.rdf"
OUTPUT_TSV = "training_final_clean.tsv"
MIN_OCCURRENCES = 3

# === Load RDF vocabulary ===
print("Loading RDF...")
g = Graph()
g.parse(RDF_FILE)
valid_uris = set(str(s) for s in g.subjects())
print(f"Found {len(valid_uris)} unique URIs in RDF.")

# === Basic cleaning of descriptions ===
def clean_text(text):
    if pd.isna(text):
        return ""
    text = text.lower()
    text = text.replace("\t", " ")  # Safe removal of TAB characters
    text = re.sub(r"\s+", " ", text)
    return text.strip()

# === Load TSV ===
print("Loading TSV...")
rows = []
with open(INPUT_TSV, "r", encoding="utf-8", errors="replace") as f:
    for line in f:
        parts = line.strip().strip("").split("\t", 1)
        if len(parts) != 2:
            continue  # skip malformed lines
        text = clean_text(parts[0])            # First column = text
        uri = parts[1].strip("<>").strip()     # Second column = URI
        if len(text.split()) >= 3:
            rows.append((uri, text))

print(f"Initial valid entries: {len(rows)}")

# === Create DataFrame ===
df = pd.DataFrame(rows, columns=["uri", "text"])

# === Filter URIs that exist in the RDF vocabulary ===
df = df[df["uri"].isin(valid_uris)]

# === Filter by URI occurrence frequency ===
valid_uris_counts = df["uri"].value_counts()
df = df[df["uri"].isin(valid_uris_counts[valid_uris_counts >= MIN_OCCURRENCES].index)]

…
```

**5_train_Annif_cve_omikuji.py**

```
Εκπαίδευση με Annif - backend omikuji.
import os
import subprocess
import shutil
import pandas as pd
from sklearn.model_selection import train_test_split
from openpyxl import Workbook
from datetime import datetime

# === SETTINGS ===
PROJECT_ID = "cve-project"
PROJECT_DIR = os.path.join("annif-projects", PROJECT_ID)
TRAINING_TSV = "training_final_clean.tsv"
SKOS_SOURCE_FILE = "cve-project.rdf"
SKOS_FILE = os.path.join(PROJECT_DIR, "cve-project.rdf")
TRAIN_FILE = os.path.join(PROJECT_DIR, "train.tsv")
TEST_FILE = os.path.join(PROJECT_DIR, "test.tsv")
ANNIF_EXECUTABLE = r"C:\Users\Pantelis\PycharmProjects\annif-project\.venv\Scripts\annif.exe"
TEMPLATE = "attentionxml"  # Επιλογή από: normal, bonsai, attentionxml

# === Ensure project folder ===
os.makedirs(PROJECT_DIR, exist_ok=True)

# === Copy RDF file ===
if not os.path.exists(SKOS_FILE):
    shutil.copyfile(SKOS_SOURCE_FILE, SKOS_FILE)
    print("Copied RDF file to project.")
else:
    print("RDF file already exists.")

# === Load training data ===
df = pd.read_csv(TRAINING_TSV, sep="\t", header=None, names=["text", "subject"])
df.dropna(inplace=True)

# === Train/test split ===
train_df, test_df = train_test_split(df, test_size=0.1, stratify=df["subject"], random_state=42)
train_df.to_csv(TRAIN_FILE, sep="\t", index=False, header=False)
test_df.to_csv(TEST_FILE, sep="\t", index=False, header=False)
print(f"Train: {len(train_df)} - Test: {len(test_df)}")

# === Templates ===
TEMPLATES = {
    "normal": {},
    "bonsai": {
        "cluster_balanced": "false",
        "cluster_k": "100",
        "max_depth": "3"
    },
    "attentionxml": {
        "cluster_balanced": "false",
        "cluster_k": "2",

…
```

**5_train_Annif_cve_fastText.py**

```
Εκπαίδευση με Annif - backend fasttext.
import os
import subprocess
import shutil
import pandas as pd
from sklearn.model_selection import train_test_split
from openpyxl import Workbook
from datetime import datetime
import glob

# === SETTINGS ===
PROJECT_ID = "cve-project-fastText"
PROJECT_DIR = os.path.join("annif-projects", PROJECT_ID)
TRAINING_TSV = "training_final_clean.tsv"
SKOS_SOURCE_FILE = "cve-project.rdf"
SKOS_FILE = os.path.join(PROJECT_DIR, "cve-project.rdf")
TRAIN_FILE = os.path.join(PROJECT_DIR, "train.tsv")
TEST_FILE = os.path.join(PROJECT_DIR, "test.tsv")
ANNIF_EXECUTABLE = r"C:\\Users\\Pantelis\\PycharmProjects\\annif-project\\.venv\\Scripts\\annif.exe"
TEMPLATE = "enrich"  # Choose from: normal, forum, enrich

# === Ensure project folder ===
os.makedirs(PROJECT_DIR, exist_ok=True)

# === Copy RDF file ===
if not os.path.exists(SKOS_FILE):
    shutil.copyfile(SKOS_SOURCE_FILE, SKOS_FILE)
    print("Copied RDF file to project.")
else:
    print("RDF file already exists.")

# === Load training data ===
df = pd.read_csv(TRAINING_TSV, sep="\t", header=None, names=["text", "subject"])
df.dropna(inplace=True)

# === Train/test split ===
train_df, test_df = train_test_split(df, test_size=0.1, stratify=df["subject"], random_state=42)
train_df.to_csv(TRAIN_FILE, sep="\t", index=False, header=False)
test_df.to_csv(TEST_FILE, sep="\t", index=False, header=False)
print(f"Train: {len(train_df)} - Test: {len(test_df)}")

# === Templates ===
TEMPLATES = {
    "normal": {},
    "forum": {
        "dim": "100",
        "lr": "0.25",
        "epoch": "5",
        "loss": "hs",
        "limit": "100",
        "chunksize": "24"

…
```

**5_train_Annif_cve_ensemble.py**

```
Εκπαίδευση με Annif - backend ensemble (omikuji + fasttext).
import os
import subprocess
import datetime
import pandas as pd
import re

# === Configuration ===
PROJECT_ID = "cve-project-ensemble"
PROJECT_DIR = "annif-projects"
TRAIN_FILE = os.path.join(PROJECT_DIR, PROJECT_ID, "train.tsv")
ANNIF_EXECUTABLE = os.path.join(".venv", "Scripts", "annif.exe")
LOG_FILE = os.path.join(PROJECT_DIR, PROJECT_ID, "training_logs.xlsx")

# === Function to remove illegal characters for Excel ===
def clean_string(text):
    if not text:
        return ""
    # Remove control characters except newline and tab
    return re.sub(r"[\x00-\x08\x0B\x0C\x0E-\x1F\x7F]", "", text)

# === Main training and logging function ===
def train_and_log():
    os.environ["ANNIF_PROJECT_DIR"] = PROJECT_DIR
    print(f"Starting training for project '{PROJECT_ID}'...")

    # Run the Annif training command
    result = subprocess.run(
        [ANNIF_EXECUTABLE, "train", PROJECT_ID, TRAIN_FILE],
        stdout=subprocess.PIPE,
        stderr=subprocess.PIPE,
        text=True,
        encoding="utf-8",
        errors="replace"
    )

    # Clean output for Excel compatibility
    cleaned_stdout = clean_string(result.stdout.strip())
    cleaned_stderr = clean_string(result.stderr.strip())

    # Create a new log entry
    log_entry = {
        "Timestamp": datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
        "Project": PROJECT_ID,
        "Train File": os.path.basename(TRAIN_FILE),
        "Status": "Success" if result.returncode == 0 else "Error",
        "Parameters": "nodes=100, dropout=0.2, epochs=10",
        "Stdout": cleaned_stdout,
        "Stderr": cleaned_stderr
    }

…
```

**6_evaluate_Annif_fast_omikuji.py**

Αξιολόγηση μοντέλου (backend omikuji) και εξαγωγή μετρικών (Precision, Recall, F1, MAP, nDCG).

```
import os
import pandas as pd
import subprocess
import uuid
from multiprocessing import Pool, cpu_count
from tqdm import tqdm
import math
from datetime import datetime

PROJECT_ID = "cve-project"
PROJECT_DIR = "annif-projects"
TEST_FILE = os.path.join(PROJECT_DIR, PROJECT_ID, "test.tsv")
ANNIF_EXECUTABLE = r"C:\Users\Pantelis\PycharmProjects\annif-project\.venv\Scripts\annif.exe"
TOP_K = 2
TEMP_DIR = os.path.join(PROJECT_DIR, PROJECT_ID, "tmp_eval_inputs")
os.makedirs(TEMP_DIR, exist_ok=True)

def normalize_uri(uri):
    return uri.strip().strip("<>").split("#")[-1].strip().upper()

def ap_score(true_set, pred_list):
    """Mean Average Precision for one sample"""
    score, hits = 0.0, 0
    for i, label in enumerate(pred_list):
        if label in true_set:
            hits += 1
            score += hits / (i + 1)
    return score / len(true_set) if true_set else 0.0

def ndcg_score(true_set, pred_list):
    """Normalized Discounted Cumulative Gain"""
    dcg = 0.0
    for i, label in enumerate(pred_list):
        if label in true_set:
            dcg += 1 / math.log2(i + 2)  # i+2 because log2(1) = 0
    idcg = sum(1 / math.log2(i + 2) for i in range(min(len(true_set), len(pred_list))))
    return dcg / idcg if idcg > 0 else 0.0

def predict_instance(args):
    text, index = args
    temp_path = os.path.join(TEMP_DIR, f"{uuid.uuid4()}.txt")
    with open(temp_path, "w", encoding="utf-8") as f:
        f.write(text)

    proc = subprocess.run(
        [ANNIF_EXECUTABLE, "suggest", PROJECT_ID, temp_path, "--limit", str(TOP_K), "-L", "en"],
        capture_output=True,
        text=True,
        encoding="utf-8"
    )

…
```

**6_evaluate_Annif_fast_fastText.py**

Αξιολόγηση μοντέλου (backend fasttext) και εξαγωγή μετρικών.

```
import os
import pandas as pd
import subprocess
import uuid
from multiprocessing import Pool, cpu_count
from tqdm import tqdm
import math
from datetime import datetime

PROJECT_ID = "cve-project-fastText"
PROJECT_DIR = "annif-projects"
TEST_FILE = os.path.join(PROJECT_DIR, PROJECT_ID, "test.tsv")
ANNIF_EXECUTABLE = r"C:\Users\Pantelis\PycharmProjects\annif-project\.venv\Scripts\annif.exe"
TOP_K = 2
TEMP_DIR = os.path.join(PROJECT_DIR, PROJECT_ID, "tmp_eval_inputs")
os.makedirs(TEMP_DIR, exist_ok=True)

def normalize_uri(uri):
    return uri.strip().strip("<>").split("#")[-1].strip().upper()

def ap_score(true_set, pred_list):
    """Mean Average Precision for one sample"""
    score, hits = 0.0, 0
    for i, label in enumerate(pred_list):
        if label in true_set:
            hits += 1
            score += hits / (i + 1)
    return score / len(true_set) if true_set else 0.0

def ndcg_score(true_set, pred_list):
    """Normalized Discounted Cumulative Gain"""
    dcg = 0.0
    for i, label in enumerate(pred_list):
        if label in true_set:
            dcg += 1 / math.log2(i + 2)  # i+2 because log2(1) = 0
    idcg = sum(1 / math.log2(i + 2) for i in range(min(len(true_set), len(pred_list))))
    return dcg / idcg if idcg > 0 else 0.0

def predict_instance(args):
    text, index = args
    temp_path = os.path.join(TEMP_DIR, f"{uuid.uuid4()}.txt")
    with open(temp_path, "w", encoding="utf-8") as f:
        f.write(text)

    proc = subprocess.run(
        [ANNIF_EXECUTABLE, "suggest", PROJECT_ID, temp_path, "--limit", str(TOP_K), "-L", "en"],
        capture_output=True,
        text=True,
        encoding="utf-8"
    )

…
```

**6_evaluate_Annif_fast_ensemble.py**

Αξιολόγηση μοντέλου (backend ensemble) και εξαγωγή μετρικών.

```
import os
import pandas as pd
import subprocess
import uuid
from multiprocessing import Pool, cpu_count
from tqdm import tqdm
import math
from datetime import datetime

# === Configuration ===
PROJECT_ID = "cve-project-ensemble"
PROJECT_DIR = "annif-projects"
TEST_FILE = os.path.join(PROJECT_DIR, PROJECT_ID, "test.tsv")
ANNIF_EXECUTABLE = r"C:\Users\Pantelis\PycharmProjects\annif-project\.venv\Scripts\annif.exe"
TOP_K = 2
TEMP_DIR = os.path.join(PROJECT_DIR, PROJECT_ID, "tmp_eval_inputs")
os.makedirs(TEMP_DIR, exist_ok=True)
os.makedirs("outputs", exist_ok=True)

# === Helper functions ===
def normalize_uri(uri):
    return uri.strip().strip("<>").split("#")[-1].strip().upper()

def ap_score(true_set, pred_list):
    score, hits = 0.0, 0
    for i, label in enumerate(pred_list):
        if label in true_set:
            hits += 1
            score += hits / (i + 1)
    return score / len(true_set) if true_set else 0.0

def ndcg_score(true_set, pred_list):
    dcg = 0.0
    for i, label in enumerate(pred_list):
        if label in true_set:
            dcg += 1 / math.log2(i + 2)
    idcg = sum(1 / math.log2(i + 2) for i in range(min(len(true_set), len(pred_list))))
    return dcg / idcg if idcg > 0 else 0.0

def predict_instance(args):
    text, index = args
    temp_path = os.path.join(TEMP_DIR, f"{uuid.uuid4()}.txt")
    with open(temp_path, "w", encoding="utf-8") as f:
        f.write(text)

    proc = subprocess.run(
        [ANNIF_EXECUTABLE, "suggest", PROJECT_ID, temp_path, "--limit", str(TOP_K), "-L", "en"],
        capture_output=True,
        text=True,
        encoding="utf-8"

…
```

**7_predictions_ensemble.py**

Πρόβλεψη/Ετικετοθέτηση νέων κειμένων με εκπαιδευμένα μοντέλα ensemble (omikuji + fasttext).

```
import os
import sys
import pandas as pd
import subprocess
from tqdm import tqdm

# === SETTINGS ===
PROJECT_ID = "cve-project-ensemble"
PROJECT_DIR = "annif-projects"
ANNIF_EXECUTABLE = r"C:\Users\Pantelis\PycharmProjects\annif-project\.venv\Scripts\annif.exe"
OUTPUT_FILE = os.path.join("outputs", "predictions_result_cve.tsv")

# === Check for correct usage ===
if len(sys.argv) != 2:
    print("Usage: python 7_predictions.py <input_tsv_path>")
    sys.exit(1)

INPUT_FILE = sys.argv[1]

# === READ INPUT FILE ===
df = pd.read_csv(INPUT_FILE, sep="\t")
if "text" not in df.columns:
    df.columns = ["text"]

print(f"Loaded {len(df)} rows from {INPUT_FILE}")

# === RUN PREDICTIONS ===
predicted_uris = []
predicted_labels = []
confidence_scores = []

for i, row in tqdm(df.iterrows(), total=len(df), desc="Running predictions"):
    text = str(row["text"])
    try:
        result = subprocess.run(
            [ANNIF_EXECUTABLE, "suggest", PROJECT_ID, "--limit", "5"],
            input=text,
            capture_output=True,
            text=True
        )
        lines = result.stdout.strip().split("\n")
        uris, labels, scores = [], [], []
        for line in lines:
            parts = line.strip().split("\t")
            if len(parts) >= 3:
                uris.append(parts[0].strip("<>"))
                labels.append(parts[1])
                scores.append(parts[2])
        predicted_uris.append(", ".join(uris))
        predicted_labels.append(", ".join(labels))

…
```

**app.py**

Λειτουργικότητα Python σχετική με το pipeline της εργασίας.

```
from flask import Flask, render_template, request, send_file
import os
import subprocess
import pandas as pd

app = Flask(__name__)

# === Script and output file paths ===
SCRIPT_FOLDER = "scripts"
OUTPUT_FOLDER = "outputs"
SCRIPT_NAME = "7_predictions_ensemble.py"
PREDICTION_FILE = os.path.join(OUTPUT_FOLDER, "predictions_result_cve.tsv")
TEMP_INPUT_FILE = "temp_uploaded.tsv"


def run_prediction(input_path):
    """Run the prediction script and parse the output file if it exists."""
    os.makedirs(OUTPUT_FOLDER, exist_ok=True)
    try:
        result = subprocess.run(
            ["python", os.path.join(SCRIPT_FOLDER, SCRIPT_NAME), input_path],
            stdout=subprocess.PIPE,
            stderr=subprocess.STDOUT,
            text=True,
            encoding="utf-8"
        )
        output = result.stdout
    except Exception as e:
        return f"Subprocess error: {e}", None, ""

    if os.path.exists(PREDICTION_FILE):
        try:
            df = pd.read_csv(PREDICTION_FILE, sep="\t")
            preview_data = []
            for _, row in df.iterrows():
                predictions = []
                uris = str(row.get("predicted_uris", "")).split(", ")
                labels = str(row.get("predicted_labels", "")).split(", ")
                scores = str(row.get("confidence_scores", "")).split(", ")
                for uri, label, score in zip(uris, labels, scores):
                    predictions.append({"uri": uri, "label": label, "score": score})
                preview_data.append({"text": row["text"], "predictions": predictions})
            return output, preview_data, PREDICTION_FILE
        except Exception as e:
            return f"{output}\nError reading output file: {e}", None, ""
    else:
        return f"{output}\nOutput file not found: {PREDICTION_FILE}", None, ""


@app.route("/", methods=["GET", "POST"])

…
```

**index.html**

HTML αρχείο που καθορίζει τη δομή της διεπαφής χρήστη UI (User Interface).

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Predictions with Annif</title>
    <style>
        .accordion-button {
            background-color: #f0f0f0;
            cursor: pointer;
            padding: 10px;
            width: 100%;
            border: none;
            outline: none;
            text-align: left;
            font-weight: bold;
            border-bottom: 1px solid #ccc;
        }
        .accordion-content {
            display: none;
            padding: 10px;
            border: 1px solid #ccc;
            border-top: none;
            background-color: #fafafa;
        }
        table { border-collapse: collapse; width: 100%; }
        th, td { border: 1px solid #ccc; padding: 6px; text-align: left; }
        th { background-color: #e0e0e0; }
        input[type="text"] { width: 300px; margin-bottom: 10px; }
    </style>
    <script>
        function toggleAccordion(btn) {
            const content = btn.nextElementSibling;
```

```
            content.style.display = content.style.display === "block" ? "none" : "block";
        }

        function filterAccordions() {
            let filter = document.getElementById("filterInput").value.toLowerCase();
            let items = document.querySelectorAll(".accordion-item");
            items.forEach(item => {
                const header = item.querySelector(".accordion-button").innerText.toLowerCase();
                item.style.display = header.includes(filter) ? "" : "none";
            });
        }
    </script>
</head>
<body>
    <h2>Predictions with Annif (Ensemble Model)</h2>
    <form action="/" method="GET" style="display:inline;">
        <button type="submit">New Prediction</button>
    </form>

…
```