| | | | |
|---|---|---|---|
| Τίτλος Έργου | Προηγμένη ανίχνευση εισβολών και παράνομου ηλεκτρονικού περιεχομένου για τη διασφάλιση επιχειρησιακής και λειτουργικής ασφάλειας σε κατανεμημένα συστήματα συστημάτων | | |
| Ακρωνύμιο Έργου | ΜΥΡΙΩΠΟΣ | | |
| Κωδικός Έργου | Τ2ΕΔΚ-04665 | | |
| Πρόσκληση / Θέμα | ΕΡΕΥΝΩ – ΔΗΜΙΟΥΡΓΩ - ΚΑΙΝΟΤΟΜΩ Β' ΚΥΚΛΟΣ / ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ, ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ & ΚΑΙΝΟΤΟΜΙΑ | | |
| Ημερομηνία Εκκίνησης | 16/06/2020 | Διάρκεια | 30 Μήνες |

# Π3.2 Προηγμένοι αλγόριθμοι και μηχανισμοί ανίχνευσης απειλών πραγματικού χρόνου από μεγάλα και μη αξιόπιστα δεδομένα

| | |
|---|---|
| **Ενότητα Εργασίας** | **ΕΕ3 Κατανεμημένη Βαθιά Ανάλυση και Νοημοσύνη Κυβερνοαπειλών** |
| **Επικεφαλής** | **UBITECH** |
| **Συνεισφορές** | **MAGGIOLI** |

# Περιεχόμενα



# Συντομογραφίες

| | |
|---|---|
| AE | Autoencoders - Αυτοκωδικοποιητές |
| API | Application Programming Interface – Διεπαφή Προγραμματισμού Εφαρμογής |
| CNN | Convolutional Neural Networks - Συνελικτικά Νευρωνικά Δίκτυα |
| DFNN | Deep Feedforward Neural Network – Απλό Νευρωνικό Δίκτυο |
| DoS | Denial-of-Service Attack – Επίθεση Άρνησης Υπηρεσιών |
| FPR | False Positive Rate - Ποσοστό Λανθασμένων προβλέψεων ως Απειλή |
| HTTP | Hypertext Transfer Protocol |
| ICMP | Internet Control Message Protocol |
| IDS | Intrusion Detection System - Συστήματα Ανίχνευσης Εισβολής |

| | |
|---|---|
| IP | Internet Protocol |
| NIDS | Network Intrusion Detection System - Συστήματα Ανίχνευσης Δικτυακής Εισβολής |
| RNN | Recurrent Neural Network - Επαναλαμβανόμενο Νευρωνικό Δίκτυο |
| SL | Supervised Learning – Μάθηση με Επίβλεψη |
| SVM | Support Vector Machine |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| UL | Unsupervised Learning – Μάθηση χωρίς Επίβλεψη |
| U2R | User-to-Root Attack - Επίθεση για απόκτηση προνομίων κεντρικού χρήστη |

## Λίστα Εικόνων



## Λίστα Πινάκων



# Περίληψη

Το παραδοτέο Π3.2 περιγράφει τους αλγορίθμους, τους μηχανισμούς και τις υπηρεσίες λογισμικού που αναπτύχθηκαν για την ανίχνευση απειλών για την επιβολή πολιτικών ανάκαμψης σε πραγματικό χρόνο. Στο παραδοτέο Π3.2 παρουσιάζεται και αναλύεται όλη η διαδικασία εντοπισμού και ανάλυσης μέσω βαθιάς μηχανικής μάθησης μεγάλων δεδομένων. Τα δεδομένα αυτά συλλέγονται σε πραγματικό χρόνο από πράκτορες του δικτύου και μέσω ορισμού πολιτικών ανάκαμψης καταστέλλονται ύποπτες και επικίνδυνες ενέργειες για έναν οργανισμό. Επίσης, παρουσιάζονται οι αλγόριθμοι αναλυτικής για την εξαγωγή

προβλέψεων και γνώσης που στοχεύουν στη λήψη αποφάσεων για θέματα κυβερνοασφάλειας. Τέλος, παρουσιάζονται οι αναφορές και οι ενέργειες που εφαρμόζονται αυτόματα από το σύστημα ΜΥΡΙΩΠΟΣ για την καταστολή των επικίνδυνων συμβάντων σε πραγματικό χρόνο.

# 1 Εισαγωγικά Στοιχεία

## 1.1 Πεδίο Εφαρμογής και Στόχοι Παραδοτέου

Η μετάβαση στην ψηφιακή εποχή έχει αυξήσει ραγδαία τις κυβερνο-απειλές. Ως εκ τούτου, καθίσταται αναγκαία η θωράκιση των οργανισμών σε επιθέσεις και ύποπτες ενέργειες σε πραγματικό χρόνο. Η εντυπωσιακή ανάπτυξη της Τεχνητής Νοημοσύνης και ειδικά της Βαθιάς Μηχανική Μάθησης, έχει βοηθήσει στην αποτελεσματικότερη και πρωτίστως, στη δυναμικότερη θωράκιση των αμυντικών μηχανισμών από πιο έξυπνες και περίπλοκες απειλές και επιθέσεις. Η υιοθέτηση αλγορίθμων Τεχνητής Νοημοσύνης για ανάπτυξη και επιβολή πολιτικών καταστολής σε πραγματικό χρόνο στο ψηφιακό κόσμο, έναντι των πιο παραδοσιακών τρόπων (όπως π.χ., static based ή rule based prevention), ωφελείται κυρίως από το γεγονός ότι οι αλγόριθμοι μπορούν πιο εύκολα να προσαρμοστούν στη αναγνώριση προτύπων, συμπεριφορών, καινούργιων απειλών αλλά και επίσης να προσαρμοστούν πιο εύκολα σε ένα συγκεκριμένο περιβάλλον (π.χ., ένα περιβάλλον που είναι πιο προσαρμοσμένο στον εκάστοτε οργανισμό).

Η ανίχνευση και πρόληψη των εισβολών σε πραγματικό χρόνο είναι ένα σύστημα από υπηρεσίες ασφαλείας που παρακολουθεί, αποθηκεύσει και αναλύει την κυκλοφορία του δικτύου για κακόβουλη δραστηριότητα και λαμβάνει μέτρα για την αποτροπή επικίνδυνων συμβάντων. Τα συστήματα αυτά συνήθως χρησιμοποιούν μια ποικιλία μεθόδων για τον εντοπισμό των εισβολών, συμπεριλαμβανομένης της ανίχνευσης βάσει υπογραφών, της ανίχνευσης βάσει ανωμαλιών και της ανάλυσης πρωτοκόλλων κατάστασης μέσω τειχών ασφαλείας.

Τα συστήματα ανίχνευσης και πρόληψης εισβολών σε πραγματικό χρόνο μπορούν να αναπτυχθούν με διάφορους τρόπους. Μπορούν να χρησιμοποιηθούν για την παρακολούθηση ενός μεμονωμένου δικτύου ή μπορούν να αναπτυχθούν σε πολλά δίκτυα. Τα συστήματα αυτά μπορούν επίσης να χρησιμοποιηθούν για την παρακολούθηση εφαρμογών και υπηρεσιών που βασίζονται σε υπολογιστικό νέφος. Στα πλαίσια αυτού του παραδοτέου, η λύση αναπτύχθηκε για να εξυπηρετήσει ένα οργανισμό του οποίου οι υπηρεσίες στηρίζονται σε υπηρεσίες άκρου και υπολογιστικού νέφους. Τα οφέλη της προτεινόμενης λύσης είναι ότι:

- Μπορεί να βοηθήσει σε πραγματικό χρόνο την αποφυγή εισβολών προτού προκαλέσει ζημιά σε υπηρεσίες και υποδομές.
- Μπορεί να παρέχει είτε πολιτικές μετριασμού της επίθεσης πραγματικού χρόνου, είτε έγκαιρη προειδοποίηση για επιθέσεις, έτσι ώστε οι διαχειριστές να μπορούν να αναλάβουν δράση για να μετριάσουν τη ζημιά.
- Μπορεί να βοηθήσει στη βελτίωση της συνολικής ασφαλείας ενός δικτύου γιατί αξιοποιεί δεδομένα μεγάλου όγκου που συλλέγονται από δικτυακές υποδομές και υπηρεσίες υπολογιστικού νέφους και συσκευών άκρου.

## 1.2 Συσχέτιση με Δράσεις και Ενότητες Εργασίας του ΜΥΡΙΩΠΟΥ

Το παραδοτέο Π3.2 περιγράψει τους αλγορίθμους, τις υπηρεσίες λογισμικού και τους μηχανισμούς που αναπτύχθηκαν για τη θωράκιση ενός δικτύου και υπηρεσιών υπολογιστικού νέφους μέσω μεθόδων Τεχνητής

Νοημοσύνης. Επιπρόσθετα, περιγράφει πως αυτοί οι αλγόριθμοι ενοποιούνται με το σύστημα ΜΥΡΙΩΠΟΣ, με σκοπό την επιβολή πολιτικών δικτύου με σκοπό το μετριασμό των επιθέσεων. Το παραδοτέο Π3.2 παρουσιάζει τα αποτελέσματα σε επίπεδο σχεδιασμού της αρχιτεκτονικής λύσης και των υπηρεσιών λογισμικού που αναπτύχθηκαν στα πλαίσια της Δράσης Δ3.2 Αλγόριθμοι στατιστικής πραγματικού χρόνου για την ανίχνευση απειλών και της Δράσης Δ3.3 Μηχανική και βαθιά μάθηση σε μεγάλα και μη αξιόπιστα δεδομένα. Επίσης, είναι το δεύτερο (2ο) παραδοτέο της ενότητας εργασίας αναφορικά με την ΕΕ3 Κατανεμημένη Βαθιά Ανάλυση και Νοημοσύνη Κυβερνοαπειλών. Ως τμήμα του ενοποιημένου Πλαισίου ΜΥΡΙΩΠΟΣ τόσο οι Δράσεις 3.2 και 3.3 όσο και το παραδοτέο Π3.2 είναι υπεύθυνα και υποστηρίζουν το σύνολο των υφιστάμενων συστημάτων διαχείρισης δεδομένων, των προγραμματιστικών διεπαφών (APIs – Application Programming Interfaces) και των αλγορίθμων που είναι απαραίτητα για την αποτελεσματική καταγραφή, ανίχνευση και καταστολή επιθέσεων μέσω πολιτικών δικτύου.

## 1.3 Δομή Παραδοτέου

Το παραδοτέο Π3.2 περιγράφει στο **Κεφάλαιο 1 – Εισαγωγή**, τον γενικό σκοπό του παραδοτέου. Τεκμηριώνει επίσης τη θέση του παραδοτέου στο έργο, δηλαδή τη σχέση του συγκεκριμένου παραδοτέου με τα άλλα παραδοτέα, δράσεις και ενότητες εργασίας, καθώς και πώς η γνώση που παράγεται στα άλλα παραδοτέα και ενότητες εργασίας χρησιμεύουν ως είσοδο στο εν λόγω παραδοτέο.

Στη συνέχεια στο **Κεφάλαιο 2 – Επισκόπηση Μεθόδων Αναλυτικής για Υπηρεσίες Κυβερνοασφάλειας,** περιγράφονται οι διάφορες μέθοδοι μοντελοποίησης και ανάλυσης συστημάτων ανίχνευσης εισβολών σε ένα δίκτυο λαμβάνοντας υπόψιν διαφορετικούς αλγορίθμους Τεχνητής Νοημοσύνης.

Το **Κεφάλαιο 3 - Διαχείριση Πολιτικών Δικτύου**, περιγράφει τις διαφορετικές κατηγορίες πολιτικών δικτύου που ορίστηκαν και εφαρμόζονται από το Σύστημα ΜΥΡΙΩΠΟΣ.

Στη συνέχεια στο **Κεφάλαιο 4 - Μέθοδοι Βαθιάς Μηχανικής Μάθησης**, γίνεται η αναλυτική περιγραφή των μοντέλων Βαθιάς Μηχανικής Μάθησης, καθώς και όλα τα επιμέρους υπο-μοντέλα που χρειάζονται για την επεξεργασία και ανάλυση. Περιλαμβάνει τη συλλογή και περιγραφή των δεδομένων, την επιλογή των σημαντικών χαρακτηριστικών, την εξαγωγή χαρακτηριστικών και τα τελικά μοντέλα που χρησιμοποιήθηκαν για τη πρόβλεψη (καθώς και σύντομη περιγραφή αυτών).

Στη συνέχεια στο **Κεφάλαιο 5 – Ανίχνευση Απειλών, Δημιουργία Αναφορών και Καταστολή Συμβάντων στο Σύστημα ΜΥΡΙΩΠΟΣ** περιγράφουμε την εμπειρία του χρήστη στο ενοποιημένο σύστημα ΜΥΡΙΩΠΟΣ καθώς και τη ροή εργασιών που χρειάζεται να ακολουθήσει για τον εντοπισμό και την καταστολή ύποπτων συμβάντων.

Τέλος, το παραδοτέο ολοκληρώνεται με το **Κεφάλαιο 6 – Επίλογος**, το οποίο αναφέρει τα κύρια αποτελέσματα και ευρήματα του παραδοτέου και παρέχει ένα πλάνο που θα καθοδηγήσει τις μελλοντικές ερευνητικές και τεχνολογικές προσπάθειες της σύμπραξης.

# 2 Επισκόπηση Μεθόδων Αναλυτικής για Υπηρεσίες Κυβερνοασφάλειας

Η σημασία της κυβερνοασφάλειας είναι αδιαμφισβήτητη. Η έρευνα σχετικά με τα Συστήματα Ανίχνευσης Εισβολής (Intrusion Detection System - IDS) είναι ένα γνωστό ερευνητικό πεδίο που, ειδικά με την πρόσφατη αύξηση της εξάρτησης από το διαδίκτυο και των σχετικών απειλών, έχει γίνει ένα αναπόσπαστο κομμάτι οποιουδήποτε συστήματος. Ανάλογα με τον τύπο του IDS, μπορούν να χρησιμοποιηθούν διαφορετικές στρατηγικές για την ανίχνευση πιθανών επιθέσεων αξιοποιώντας μεγάλα δεδομένα καταγραφής της κίνησης του δικτύου. Μπορούμε να κατηγοριοποιήσουμε τα IDS σε τρεις κύριες κατηγορίες με βάση τους αλγόριθμους που χρησιμοποιούν για τον έγκαιρο εντοπισμό ύποπτων συμβάντων:

- **Rule based:** Η ιδέα πίσω από τέτοια συστήματα είναι να παρακολουθούν τα γεγονότα και να τα συγκρίνουν με τα μοτίβα και τις υπογραφές που συλλέγονται βάσει γνωστών επιθέσεων, ευπαθειών και καθορισμένων κανόνων πολιτικής. Αυτές οι μέθοδοι μπορούν να χαρακτηριστούν από υψηλό ποσοστό ακρίβειας και χαμηλό αριθμό ψευδών θετικών αναφορών για τις γνωστές και καθιερωμένες επιθέσεις. Ωστόσο, είναι αναποτελεσματικές όταν συμβούν νέες (δίχως γνώση σε παρελθοντικά μοτίβα, ή αλλιώς zero-day) ή τροποποιημένες επιθέσεις: κάθε τροποποίηση της επίθεσης απαιτεί εντελώς νέους ή προσαρμογή των ήδη υπαρχόντων κανόνων. Αυτό σημαίνει επίσης ότι τα IDS βασισμένα σε κανόνες απαιτούν πολύ περισσότερη συντήρηση και τακτικές ενημερώσεις για να παραμείνουν συνεχώς επίκαιρα [1].

- **Static based:** Τέτοια συστήματα ανιχνεύουν επιθέσεις δημιουργώντας στατιστικές κατανομές μοτίβων διείσδυσης / επίθεσης. Μετρώντας μια απόκλιση της κατανομής τμήματος της τρέχουσας κίνησης από τη μετρημένη (υποτιθέμενη αλήθεια) ανακαλύπτουν τις επιθέσεις. Ωστόσο, ανάλογα με τα IDS βασισμένα σε κανόνες (π.χ., rule based), χρειάζονται τακτικές ενημερώσεις για να μπορούν να εντοπίσουν ακριβώς την κίνηση. Επιπλέον, προκειμένου να δημιουργηθεί ένα στατιστικό μοντέλο για σύγκριση, τα δεδομένα χρειάζεται να είναι καθαρά και χωρίς θόρυβο, ώστε οι κατανομές να αντιπροσωπεύουν ρεαλιστική κίνηση από πραγματικά συστήματα [2].

- **Machine Learning based:** Με την ραγδαία ανάπτυξη της Τεχνητής Νοημοσύνης που οφείλεται στη εξέλιξη της Μηχανικής Μάθησης, δημιουργούνται συστήματα που χρησιμοποιούν Μηχανική Μάθηση και Βαθιά Μηχανική Μάθηση. Αυτές οι μέθοδοι χρησιμοποιούν διάφορα μοντέλα για να διακρίνουν και να ταξινομήσουν τις επιθέσεις από την κανονική κίνηση. Παρότι απαιτούν σημαντικό όγκο δεδομένων για να εκπαιδευτούν εκτενώς, με την πρόσφατη αύξηση του όγκου των δεδομένων που ανταλλάσσονται στο Διαδίκτυο, αποκτούν σημαντική δημοτικότητα και αποτελούν το επίκεντρο της ερευνητικής κοινότητας. Ο στόχος τους είναι να ανιχνεύουν μια ανωμαλία στα δεδομένα, κυρίως στη στατιστική ανάλυση τους στο χρόνο, επομένως, είναι κατάλληλες για την ανίχνευση άγνωστων επιθέσεων δίχως προηγούμενη γνώση από παρελθοντικά μοτίβα [3].

Από το σημείο αυτό και στη συνέχεια, το παραδοτέο Π3.2 επικεντρώνεται σε αλγορίθμους μηχανικής μάθησης. Μέσα στις μεθόδους που βασίζονται στην Μηχανική Μάθηση, μπορούμε να διακρίνουμε δύο υποκατηγορίες τεχνικών:

- Τους "κλασικούς" αλγόριθμους Μηχανικής Μάθησης [1], όπως οι K-Nearest Neighbour, K-Means, Hidden Markov Models, Support Vector Machines (SVMs); και
- Την Βαθιά Μηχανική Μάθηση (Deep Learning), έναν αρκετά νέο κλάδο της Μηχανικής Μάθησης, όπου οι τεχνικές περιλαμβάνουν Αυτοκωδικοποιητές (Autoencoders), Συνελικτικά Νευρωνικά Δίκτυα (CNN) και άλλα [2], [3].

Ενώ οι κλασικοί αλγόριθμοι Μηχανικής Μάθησης απαιτούν λιγότερα δεδομένα και υπολογιστική ισχύ, συχνά δεν είναι σε θέση να μοντελοποιήσουν την πολυπλοκότητα του προβλήματος. Οι τεχνικές της Βαθιάς Μηχανικής Μάθησης είναι πιο προσαρμοσμένες σε μεγάλους όγκους δομημένων και αδόμητων δεδομένων, όπως η κίνηση στο δίκτυο [3]. Και γενικά έχουν στη πράξη αποδειχθεί πιο αποτελεσματικοί.

Κάποιες προηγούμενες ερευνητικές εργασίες όσον αφορά στα συστήματα ανίχνευσης εισβολών (IDS) με χρήση μοντέλων βαθιάς μηχανικής μάθησης περιλαμβάνουν:

- **Random Forest [4]:** Η εργασία αυτή προτείνει τον αλγόριθμο επιβλεπόμενης μάθησης τυχαίων δασών για την εκπαίδευση ενός μοντέλου που ανιχνεύει διάφορες επιθέσεις δικτύου. Για να αυξήσουν περαιτέρω την ακρίβεια ταξινόμησης του μοντέλου τους, χρησιμοποιήσαν τη διαδεδομένη τεχνική εξόρυξης δεδομένων μέσω της εξαγωγής χαρακτηριστικών. Έχει γίνει έξυπνη επιλογή χαρακτηριστικών από το σύνολο των δεδομένων χρησιμοποιώντας τον Συντελεστή Τζίνι για να μειωθεί η διάσταση και ως εκ τούτου ο αριθμός των χαρακτηριστικών. Τα πειραματικά αποτελέσματα δείχνουν ότι το μοντέλο τους τρέχει γρήγορα και παρουσιάζει υψηλή ακρίβεια.
- **Autoendoder [5]:** Η εργασία αυτή παρουσιάζει το Kitsune: ένα εύκολο στην εγκατάσταση και χρήση NIDS (Network Intrusion Detection System) που ανιχνεύει επιθέσεις στο τοπικό δίκτυο, χωρίς επίβλεψη και με αποδοτικό τρόπο σε πραγματικό χρόνο. Ο πυρήνας αλγορίθμου του Kitsune (KitNET) χρησιμοποιεί ένα σύνολο νευρωνικών δικτύων που ονομάζονται Αυτοκωδικοποιητές (Autoencoders) για να διαχωρίσει συλλογικά τα κανονικά και ύποπτα μοτίβα κίνησης. Το KitNET υποστηρίζεται από ένα πλαίσιο εξαγωγής χαρακτηριστικών που ανιχνεύει αποδοτικά τα μοτίβα κάθε δικτυακού καναλιού. Οι αξιολογήσεις δείχνουν ότι Kitsune μπορεί να ανιχνεύει διάφορες επιθέσεις με απόδοση που συγκρίνεται με ανιχνευτές ανωμαλιών εκτός τοπικού δικτύου, ακόμη και σε ένα Raspberry PI. Αυτό δείχνει ότι το Kitsune μπορεί να αποτελέσει ένα ρεαλιστικό και αποτελεσματικό NIDS.
- **Συνελικτικά Νευρωνικά Δίκτυα CNN [6]:** Η εργασία παρουσιάζει ένα μοντέλο εισβολής βασισμένο σε βαθιά μάθηση (Deep Learning - DL) με ιδιαίτερη εστίαση στις επιθέσεις αποκλεισμού υπηρεσίας (Denial-of-Service ή ισοδύναμα DoS). Για το σύνολο δεδομένων εισβολής χρησιμοποίησαν το ανοικτό σύνολο δεδομένων KDD CUP 1999, το πιο ευρέως χρησιμοποιούμενο σύνολο δεδομένων για την αξιολόγηση των συστημάτων ανίχνευσης εισβολής (IDS). Το σύνολο δεδομένων KDD αποτελείται από τέσσερις κατηγορίες επιθέσεων, όπως οι επιθέσεις DoS, user to root (U2R), απομακρυσμένου προς τοπικού υπολογιστή (R2L) και ανίχνευση. Πολλές μελέτες σχετικά με το σύνολο δεδομένων KDD έχουν χρησιμοποιήσει αλγόριθμους μηχανικής μάθησης για την ταξινόμηση του συνόλου δεδομένων σε αυτές τις τέσσερις κατηγορίες (multi-class classification) ή σε δύο κατηγορίες (π.χ., binary classification), όπως επικίνδυνο – μη κανονικό και ακίνδυνο - κανονικό. Αντί να επικεντρωθούν στις ευρείες κατηγορίες, επικεντρωθήκαν σε διαφορετικές επιθέσεις που ανήκουν στην ίδια κατηγορία. Αντίθετα με τις άλλες κατηγορίες επιθέσεων του συνόλου δεδομένων KDD, η κατηγορία DoS έχει αρκετά δείγματα για εκπαίδευση σε κάθε επίθεση. Εκτός από το σύνολο δεδομένων KDD, χρησιμοποιήθηκε επίσης το CSE-CIC-IDS2018, το πιο ενημερωμένο σύνολο δεδομένων IDS. Το CSE-

CIC-IDS2018 περιλαμβάνει πιο προηγμένες επιθέσεις DoS από αυτές του KDD. Σε αυτήν την εργασία, επικεντρώθηκαν στην κατηγορία DoS και των δύο συνόλων δεδομένων και ανέπτυξαν ένα μοντέλο βαθιάς μηχανικής μάθησης για την ανίχνευση DoS, βασισμένο σε CNN και επαναλαμβανόμενο νευρωνικό δίκτυο (RNN), με το CNN να αποδίδει καλύτερα σύμφωνα με τα πειράματα τους.

# 3 Διαχείριση Πολιτικών Δικτύου

Όπως αναφέρθηκε πιο πάνω, στο σύστημα ΜΥΡΙΩΠΟΣ υπάρχει και εφαρμόζεται πληθώρα πολιτικών δικτυού με σκοπό την άμεση ανάδραση σε μια ανερχόμενη και αναπάντεχη κατάσταση κατά την παρακολούθησης της ροής του δικτύου. Συγκεκριμένα, για την εφαρμογή αυτών των πολιτικών, χρησιμοποιείται το Cilium, που χρησιμοποιείται για δικτύωση, ασφάλεια και παρακολούθηση εφαρμογών σε περιβάλλοντα υπολογιστικού νέφους, και κατά κύριο λόγο στο περιβάλλον Kubernetes (K8S).

**Πολιτικές Δικτύου:** Οι Πολιτικές Δικτύου του Cilium ορίζονται χρησιμοποιώντας τα αντικείμενα NetworkPolicy του Kubernetes. Επιτρέπουν τον καθορισμό κανόνων που ορίζουν ποια επικοινωνία επιτρέπεται ή απαγορεύεται μεταξύ των pods ή των namespaces, βάση διαφόρων κριτήριων όπως διευθύνσεις IP, namespaces, θύρες και πρωτόκολλα. Οι πολιτικές δικτύου που χρησιμοποιούνται έχουν διάφορους τρόπους που μπορούν να εφαρμοστούν ενώ παράλληλα αποτελούνται από διάφορους τύπους ανάλογα με το περιεχόμενο. Συγκεκριμένα οι πολιτικές διακρίνονται.

Ανάλογα με τη κατεύθυνση της κίνησης με την οποία εφαρμόζονται:

1. Egress: Εφαρμόζεται σε πακέτα δικτύου τα οποία εξέρχονται από το σύστημα που προστατεύεται.
2. Ingress: Εφαρμόζεται σε πακέτα δικτύου τα οποία εισέρχονται από το σύστημα που προστατεύεται.

Ανάλογα με το επίπεδο στο οποίο εφαρμόζονται:

1. **Layer-3 (Επίπεδο Δικτύου):** Το Cilium μπορεί να ορίσει πολιτικές βασιζόμενο σε διευθύνσεις IP και υποδίκτυα. Αυτό επιτρέπει τον έλεγχο της κίνησης του δικτύου μεταξύ των pods ή των namespaces σε επίπεδο IP.
2. **Layer-4 (Επίπεδο Μεταφοράς):** Το Cilium μπορεί να ορίσει πολιτικές βασιζόμενο σε θύρες και πρωτόκολλα. Μπορεί να καθοριστούν κανόνες που να επιτρέπουν ή να αποκλείουν την κίνηση βάσει συγκεκριμένων αριθμών θυρών ή πρωτοκόλλων, όπως TCP, UDP ή ICMP.
3. **Layer-7 (Επίπεδο Εφαρμογής):** Οι Πολιτικές Δικτύου του Cilium μπορούν να επιβάλλουν αποφάσεις πολιτικής στο Επίπεδο 7, πράγμα που σημαίνει ότι μπορεί να ορισθούν πολιτικές βασισμένες σε χαρακτηριστικά του επιπέδου εφαρμογής, όπως σε επικεφαλίδες HTTP ή διαδρομές URL. Αυτό επιτρέπει την επιβολή πολιτικών που βασίζονται σε πιο προηγμένη συμπεριφορά της εφαρμογής.

Σε αυτό το κεφάλαιο περιγράφεται η συνεισφορά μας, που αποτελείται από τρία (3) κύρια μέρη: α) το λογισμικό Εξαγωγής Χαρακτηριστικών (Feature Engineering), β) το λογισμικό Επιλογής Χαρακτηριστικών (Feature Engineering), και γ) το λογισμικό Βαθιάς Μηχανικής Μάθησης (Deep Learning).

Το λογισμικό Βαθιάς Μηχανικής Μάθησης είναι γραμμένο σε Python3 και χρησιμοποιεί αρκετές υπάρχουσες γνωστές βιβλιοθήκες. Για τη μοντελοποίηση των μεθόδων Βαθιάς Μηχανικής Μάθησης, χρησιμοποιήσαμε τις παρακάτω βιβλιοθήκες: Keras / Tensorflow και scikit-learn. Για επιπλέον επεξεργασία και ανάλυση δεδομένων, χρησιμοποιήσαμε τις βιβλιοθήκες Pandas, Numpy, seaborn, Matplotlib.

## 3.1 Εξαγωγή Χαρακτηριστικών

Αυτή η ενότητα έχει ως στόχο την επεξεργασία της ακατέργαστης ροής δεδομένων του δικτύου, έτσι ώστε να εξαχθεί η απαραίτητη πληροφορία καθώς και χαρακτηριστικά που απαιτούνται ώστε να μπορεί η επίθεση να ανιχνευθεί, αλλά και για να μπορέσει το μοντέλο να εκπαιδευτεί και να λειτουργήσει σωστά.

Συγκεκριμένα, τα δεδομένα έρχονται σαν ροή (π.χ., stream) από πακέτα δικτύου, αρά τα δεδομένα στην ουσία είναι ένας πίνακας από πακέτα δικτύου. Ωστόσο, το πακέτο από μόνο του δεν μπορεί να δώσει οποιαδήποτε πληροφορία για μια ενδεχόμενη κατάσταση δικτύου ή/και επίθεση.

Για αυτό, από τα χαρακτηριστικά των πακέτων που έρχονται εξάγονται ιδιότητες οι οποίες προσδιορίζουν τις συνδέσεις που έχει το μηχάνημα με άλλα μηνύματα και IPs. Συγκεκριμένα η τελική μορφή των δεδομένων είναι ένα ευρετήριο με όλες τις IP στις οποίες υπάρχει επικοινωνία με το μηχάνημα μας, μαζί με τα εκάστοτε χαρακτηριστικά των επιμέρους IP/συνδέσεων.

Η συλλογή της ροής των πακέτων εκτελείται από το Packetbeat, το οποίο έχει εγκατασταθεί στο Kubernetes Cluster. Τα δεδομένα από το Packetbeat αποθηκεύονται στο Apache Kafka ως ροή δεδομένων με διάρκεια ζωής 12 ωρών για την αποτελεσματικότερη χρήση των υπολογιστικών και αποθηκευτικών πόρων.

Τα πακέτα που συλλέγονται φέρουν τα ακόλουθα αρχικά χαρακτηριστικά:

1. **timestamp:** Η ώρα και ημερομηνία που καταγράφηκε το εν λόγω πακέτο.
2. **agent info:** Πληροφορίες για το Packetbeat (μηχάνημα που λειτουργεί, id, κλπ.).
3. **Source Host info:** Πληροφορίες για τον αποστολέα του πακέτου (IP, αριθμός πακέτου, αριθμός bytes, port).
4. **Destination Host info:** Πληροφορίες για τον παραλήπτη του πακέτου (IP, αριθμός πακέτου, αριθμός bytes, port).
5. **Host info:** Πληροφορίες για το μηχάνημα που τρέχει το Packetbeat (πληροφορίες για το λειτουργικό σύστημα, id, MAC, IP κλπ.).
6. **Network Info:** Πληροφορίες για το δίκτυο (αριθμός bytes, protocol).
7. **Specific Protocol Info:** Επιπρόσθετες πληροφορίες για κάποιο συγκεκριμένο πρωτόκολλο ή υπηρεσία (π.χ., ICMP, HTTP, Redis, κλπ.)
8. **Type:** Τύπος του πακέτου (HTTP, ICMP κλπ.)

Από τα προαναφερόμενα δεδομένα, αυτά που χρησιμοποιήθηκαν για την εξαγωγή των νέων χαρακτηριστικών του μοντέλου είναι τα χαρακτηριστικά 1, 3, 4 και 8.

Συγκεκριμένα τα δεδομένα αυτά συγκεντρώνονται ανά 2 δευτερόλεπτα ανά μοναδική IP που η εικονική μηχανή η οποία τρέχει το Packetbeat έχει σύνδεση, και περιέχουν διάφορες στατιστικές για την σύνδεση, ως ακολούθως:

- μέσος ορός bytes που το μηχάνημα που τρέχει το Packetbeat δέχεται από τον άλλο χρήστη στη σύνδεση
- πρωτόκολλο πακέτων στη σύνδεση (εάν είναι HTTP, ή ICMP κλπ.)
- μέσος ορός bytes που το μηχάνημα που τρέχει το Packetbeat στέλνει στον άλλο χρήστη στη σύνδεση,

- αριθμός πακέτων, και
- αλλά επιμέρους στατιστικά για τη σύνδεση με το ίδιο χρήστη στο διάστημα των 2 δευτερολέπτων.

Επιπρόσθετα, αφού πρώτα συλλέξουμε τις πρώτες αναλυτικές από τα παραπάνω δεδομένα, ενισχύουμε τις αναλυτικές και τις μετρικές μας, με υφιστάμενα δεδομένα από το σύνολο δεδομένων KDD99 τα οποία εμπλουτίζουν το συλλεχθέν σύνολο δεδομένων.

Επομένως τα τελικά χαρακτηριστικά πριν τη τελική ανάλυση παρουσιάζονται λεπτομερώς στη συνέχεια. Σε αυτό το σημείο, γίνεται η εξαγωγή των χαρακτηριστικών όπως φαίνεται παρακάτω. Αυτό γίνεται με τη εξαγωγή των χαρακτηριστικών από τη ροή δεδομένων από το Packetbeat, και με τη χρήση της βιβλιοθήκης Pandas. Γενικά, δημιουργείται ένα ευρετήριο με τις IP που το μηχάνημα έχει επικοινωνία, συγκεντρώνοντας τις πιο κάτω τιμές ανά διάστημα 2 δευτερολέπτων. Δηλαδή κάθε τιμή που καταγράφεται έχει χρονικό ορίζοντα τα 2 δευτερόλεπτα.

Επιπρόσθετα, λόγω ότι κάποιες τιμές είναι τύπου «Κατηγορίας», δηλαδή περιέχουν διακριτές μη αριθμητικές τιμές, επιπρόσθετη επεξεργασία είναι απαραίτητη, ώστε όλα τα χαρακτηριστικά να είναι σε αριθμητική τιμή. Συγκεκριμένα, για αυτές τις τιμές δημιουργούνται χαρακτηριστικά ίσα με το αριθμό των διαφορετικών τιμών που έχουν, στο πεδίο τους. Για παράδειγμα, service_http, service_icmp, κλπ. Όπου κάθε νέο χαρακτηριστικό φέρει την τιμή 1 εάν η αρχική τιμή «Κατηγορίας» ήταν αυτή η τιμή, αλλιώς 0. Ο Πίνακας 1 παρουσιάζει όλα τα χαρακτηριστικά, όπως όνομα, περιγραφή και τύπος που αναλύονται από το τελικό μοντέλο.

| Όνομα | Περιγραφή | Τύπος |
|---|---|---|
| **protocol** | τύπος πρωτοκόλλου, π.χ., tcp, udp, κλπ. | Διακριτή |
| **service** | υπηρεσία δικτύου, π.χ., http, telnet, κλπ. | Διακριτή |
| **dst_bytes** | αριθμός πακέτων από το παραλήπτη στον αποστολέα | Συνεχής |
| **src_bytes** | αριθμός πακέτων από τον αποστολέα στον παραλήπτη | Συνεχής |
| **count** | αριθμός συνδέσεων με τον ίδιο χρήστη | Συνεχής |
| **same_srv_rate** | % των συνδέσεων με την ιδιά υπηρεσία | Συνεχής |
| **diff_srv_rate** | % των συνδέσεων με διαφορετικές υπηρεσίες | Συνεχής |
| **srv_count** | % των συνδέσεων με την ίδια υπηρεσία στα 2 τελευταία δευτερόλεπτα | Συνεχής |
| **srv_diff_host_rate** | % των συνδέσεων με διαφορετικούς χρήστες | Συνεχής |
| **srv_serror_rate** | % των συνδέσεων που έχουν σφάλμα «REJ»' | Συνεχής |

**Πίνακας 1. Είδη Χαρακτηριστικών Εκπαίδευσης.**

## 3.2 Επιλογή Κατάλληλων Παραμέτρων

Από τα πιο πάνω χαρακτηριστικά, επιλέγονται αυτά που είναι τα πιο σημαντικά για τη πρόβλεψη σύνδεσης, ώστε να μειωθεί ο θόρυβος που έρχεται από ασήμαντα χαρακτηριστικά.

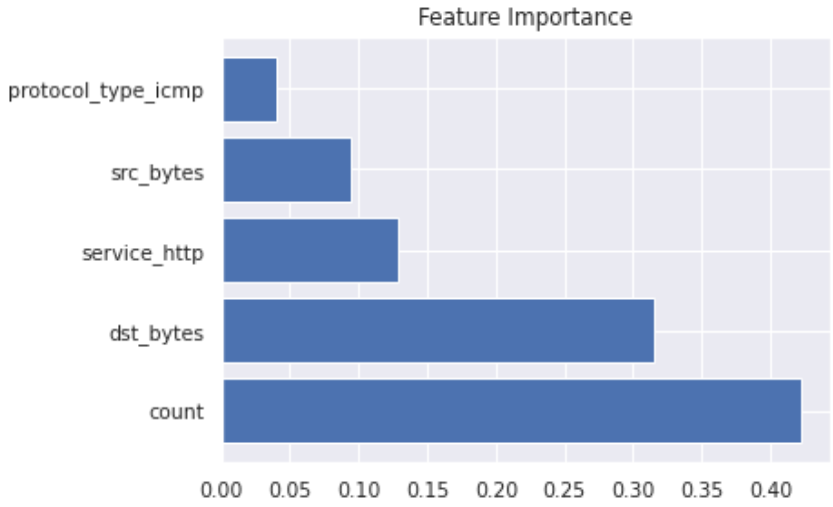
Συγκεκριμένα συνδυάζοντας τους παρακάτω τρόπους (βλ. Εικόνα 1 και Εικόνα 2) επιλεγούμε τα πιο κάτω χαρακτηριστικά. Τα αλλά χαρακτηριστικά που δεν είναι σημαντικά μπορούν να αγνοηθούν καθώς μπορούμε να δούμε ότι η απόδοση των μοντέλων δεν γίνεται χειρότερη και σε κάποια μοντέλα η απόδοση πάει στο μέγιστο (βλ. DFNN)

Ο πρώτος τρόπος χρησιμοποιεί την ιδιότητα του πρώτου μοντέλου (Random Forest) [10] για να δώσει τη σημασία των χαρακτηριστικών στη κατηγοριοποίηση. Στην ουσία υπολογίζεται ο μέσος όρος της αντίθετης τιμής impurity. Impurity είναι ένα μέτρο αταξίας που χρησιμοποιείται στους αλγόριθμους δέντρων απόφασης, συμπεριλαμβανομένου του Random Forest. Μετρά την πιθανότητα να γίνει λανθασμένη κατηγοριοποίηση ενός τυχαία επιλεγμένου στοιχείου σε ένα σύνολο δεδομένων, εάν είχε τυχαία ετικετοποιηθεί σύμφωνα με την κατανομή των κατηγοριών μέσα σε ένα υποσύνολο.

Ο δεύτερος τρόπος είναι η συσχέτιση (correlation). Στατιστικό που δείχνει τη γραμμική σχέση μεταξύ δύο μεταβλητών. Στη προκείμενη περίπτωση, υπολογίστηκε για όλα τα χαρακτηριστικά σε σχέση με τη κατηγορία της σύνδεσης (target) Κανονική (0) ή Ύποπτη (1). Όπου 1 σημαίνει μέγιστη γραμμική σχέση (εξίσου και το ένα με το εαυτό της) και -1 σημαίνει μέγιστη αντίστροφος ανάλογη σχέση.

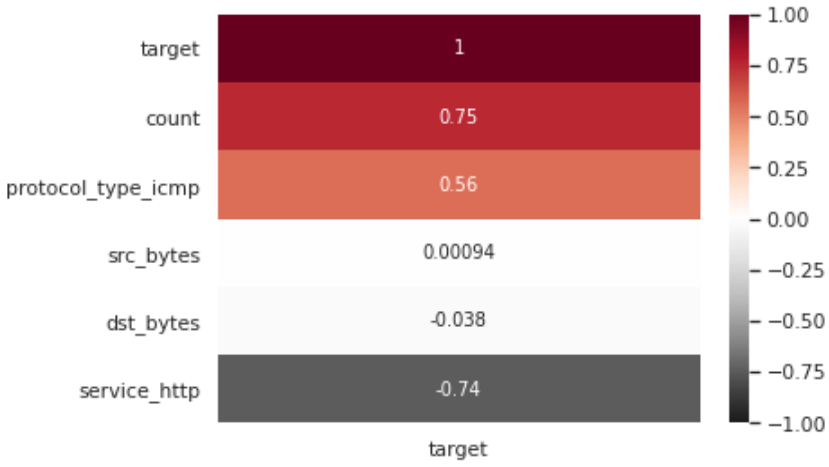
Το πιο σημαντικό κριτήριο επιλογής, ωστόσο, είναι ότι με τα επιλεγόμενα επιτυγχάνεται υψηλή απόδοση των μοντέλων. Μάλιστα, ένα από τα μοντέλα πετυχαίνει και 100% απόδοση.

Η Εικόνα 1 παρουσιάζει τη σημασία των χαρακτηριστικών (>0.02) από το Random Forest μοντέλο. Δείχνει το αντίστροφο impurity score το οποίο εκφράζει τη σημασία ενός χαρακτηριστικού στην μοντελοποίηση χρησιμοποιώντας το Random Forest μοντέλο. Στην ουσία δείχνει ποια χαρακτηριστικά λαμβάνονται περισσότερο υπόψη κατά την ταξινόμηση / κατηγοριοποίηση των δεδομένων. Χαρακτηριστικά που δεν έχουν σχετικά υψηλή τιμή απορρίπτονται και δεν χρησιμοποιούνται περεταίρω αφού στο τέλος μόνο θόρυβο προσθέτουν που δυσκολεύει το μοντέλο εκπαίδευσης.




**Εικόνα 1. Σημασία χαρακτηριστικών.**

Η Εικόνα 2 παρουσιάζει τη συσχέτιση των χαρακτηριστικών με μεταβλητή στόχο και πρόβλεψης το Κανονικό = 0 ή Ύποπτο = 1. Υψηλή συσχέτιση μπορεί να αποτελέσει ένδειξη ότι το χαρακτηριστικό συνεισφέρει στην πρόβλεψη του 0 (Κανονικό) ή του 1 (Ύποπτο), καθώς η διακύμανση στην τιμή του χαρακτηριστικού προς τα πάνω ή προς τα κάτω μπορεί να αλλάξει την τιμή της κατηγορίας από 0 σε 1 και αντίστροφα.




**Εικόνα 2. Συσχέτιση χαρακτηριστικών.**

## 3.3 Μέθοδοι Μάθησης από Μεγάλα Δεδομένα Καταγραφής Κίνησης Δικτυακών Υποδομών

Αφού γίνει η τελική επιλογή των σημαντικών χαρακτηριστικών μαζί με την κατηγοριοποίηση της εγγραφής / σύνδεσης σε Κανονική (0) η Ύποπτη (1), στη συνέχεια γίνεται η εκπαίδευση των μοντέλων με σκοπό την ανίχνευση μελλοντικών απειλών. Τα δεδομένα κίνησης που συλλέχθηκαν είχαν όγκο 500GB και ο χρόνος εκπαίδευσης διήρκησε μια (1) εβδομάδα.

Η **Μάθηση με Επίβλεψη** (Supervised Learning) είναι μια κατηγορία της μηχανικής μάθησης που εμπλέκει στην εκπαίδευση των μοντέλων ετικετοποιημένα (labelled) σύνολα δεδομένων. Στη μάθηση με επίβλεψη, το σύνολο δεδομένων αποτελείται από χαρακτηριστικά εισόδου και τις αντίστοιχες ετικέτες στόχου, οι οποίες χρησιμοποιούνται για να καθοδηγήσουν τη διαδικασία μάθησης. Ο απώτερος στόχος είναι να δημιουργηθεί μια συνάρτηση αντιστοίχισης που μπορεί να προβλέψει με ακρίβεια τις ετικέτες στόχου για νέα και άγνωστα δεδομένα εισόδου. Η διαδικασία μάθησης περιλαμβάνει την προσαρμογή των παραμέτρων του μοντέλου με στόχο την ελαχιστοποίηση της απόκλισης μεταξύ των προβλεπόμενων εξόδων και των πραγματικών ετικετών. Στη προκείμενη περίπτωση, το μοντέλο που δημιουργείται, εκπαιδεύεται στο να κατηγοριοποιεί τα δεδομένα σε Κανονικά (0) ή Ύποπτα (1).

Το Random Forest είναι ένας δημοφιλής αλγόριθμος μηχανικής μάθησης που χρησιμοποιείται τόσο για ταξινόμηση όσο και για προβλήματα παλινδρόμησης. Λειτουργεί κατασκευάζοντας ένα σύνολο από δέντρα αποφάσεων πραγματοποιώντας προβλέψεις βασισμένες στο σύνολο των εξόδων. Ο αλγόριθμος ξεκινάει επιλέγοντας τυχαία ένα υποσύνολο από τα δεδομένα εκπαίδευσης, γνωστό ως bootstrap δείγμα, και κατασκευάζει ένα δέντρο απόφασης χρησιμοποιώντας αυτό το υποσύνολο. Το δέντρο αποφάσεων

εκπαιδεύεται με αναδρομική διαμέριση των δεδομένων με βάση το διαχωρισμό των κλάσεων που βελτιστοποιείται όταν εκτελείται ταξινόμηση ή το σφάλμα πρόβλεψης μειώνεται όταν εκτελείται παλινδρόμηση. Αυτή η διαδικασία επαναλαμβάνεται για να δημιουργηθεί μια συλλογή από δέντρα αποφάσεων, καθένα εκπαιδευμένο σε διαφορετικό bootstrap δείγμα. Κατά την πρόβλεψη, ο αλγόριθμος Random Forest συγκεντρώνει τις ατομικές προβλέψεις από όλα τα δέντρα και επιλέγει την πλειοψηφία τους.

Το Deep Feedforward Neural Network (γνωστό και ως πολυεπίπεδο νευρωνικό δίκτυο προώθησης) είναι ένα είδος τεχνητού νευρωνικού δικτύου που αποτελείται από πολλά συνελικτικά επίπεδα (ή κρυφά επίπεδα) που ακολουθούνται από ένα τελικό επίπεδο εξόδου. Λειτουργεί με τρόπο ώστε τα δεδομένα εισόδου να προωθούνται μπροστά μέσω του δικτύου με την εφαρμογή γραμμικών και μη γραμμικών συναρτήσεων στα κρυφά επίπεδα και μετά επεξεργάζονται για να εξάγουν συγκεκριμένα χαρακτηριστικά. Κάθε κρυφό επίπεδο αποτελείται από πολλούς νευρώνες, οι οποίοι συνδέονται με όλους τους νευρώνες του προηγούμενου και του επόμενου επιπέδου. Τα βάρη και οι παράμετροι του δικτύου εκπαιδεύονται με χρήση αλγορίθμων όπως ο αλγόριθμος οπισθοδρόμησης, με σκοπό να μειωθεί η απόκλιση (δλδ. το σφάλμα/η διαφορά που απορρέει) μεταξύ των πραγματικών εξόδων και των εξόδων που παράγονται από το δίκτυο, που είναι και σκοπός της μάθησης με επίβλεψη.

Η Εικόνα 3 παρουσιάζει τη μορφή του νευρωνικού δικτύου. Φαίνονται τα διαφορετικά επίπεδα (π.χ. 3 στο σύνολο), ο τύπος του επιπέδου (dense που είναι ένα απλό επίπεδο από νευρώνες), ο αριθμός των νευρώνων σε κάθε επίπεδο καθώς και ο αριθμός των παραμέτρων (συνήθως τα βάρη των συνδέσεων μεταξύ νευρώνων).

```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_15 (Dense)             (None, 16)                112
_________________________________________________________________
dense_16 (Dense)             (None, 8)                 136
_________________________________________________________________
dense_17 (Dense)             (None, 1)                 9
=================================================================
Total params: 257
Trainable params: 257
Non-trainable params: 0
_________________________________________________________________
```

**Εικόνα 3. Εκπαίδευση Νευρωνικού Δικτύου.**

Η **Μάθηση χωρίς Επίβλεψη** (Unsupervised Learning) είναι μια κατηγορία αλγορίθμων μηχανικής μάθησης που διαχειρίζονται την ανάλυση αμέτρητων και μη ετικετοποιημένων δεδομένων. Αντίθετα με την επιβλεπόμενη μάθηση, δεν υπάρχουν ετικέτες ή οδηγοί που να καθοδηγούν τη διαδικασία μάθησης. Στόχος της μάθησης χωρίς επίβλεψη είναι να ανακαλύψει δομικά μοτίβα, κρυφές σχέσεις και σημαντικές πληροφορίες που είναι κρυμμένες στα δεδομένα. Οι αλγόριθμοι μάθησης χωρίς επίβλεψη μπορούν να εφαρμοστούν με διάφορους τρόπους, όπως ο αλγόριθμος K-Means για ομαδοποίηση [7], ο αλγόριθμος Hierarchical Clustering για την ανακάλυψη της ιεραρχίας των συστάδων [8] και ο αλγόριθμος των Autoencoders [9] για την εξαγωγή χαρακτηριστικών και της διάστασης των δεδομένων. Οι αλγόριθμοι μάθησης χωρίς επίβλεψη ανοίγουν τον δρόμο για την κατανόηση της δομής των δεδομένων και την ανακάλυψη κρυφών πληροφοριών. Στη προκείμενη περίπτωση, το μοντέλο που δημιουργείται, εκπαιδεύεται

σε δεδομένα που είναι Κανονικά (0), με σκοπό ο αλγόριθμος να μάθει τη κρυφή δομή και κατανομή τους, ώστε, να αναγνωρίζει δεδομένα που διαφοροποιούνται από αυτή τη δομή.

Ο Autoencoder είναι ένα είδος νευρωνικού δικτύου που χρησιμοποιείται για την αυτόματη εξαγωγή χαρακτηριστικών από μη επιβλεπόμενα δεδομένα. Το συγκεκριμένο μοντέλο αποτελείται από δύο κύρια τμήματα: τον κωδικοποιητή (encoder) και τον αποκωδικοποιητή (decoder). Ο κωδικοποιητής μετατρέπει τα εισερχόμενα δεδομένα σε μια πιο συμπιεσμένη αναπαράσταση, ονομάζοντας την "κωδικοποιημένη αναπαράσταση" ή "latent space". Στη συνέχεια, ο αποκωδικοποιητής αναδημιουργεί τα δεδομένα από τον χώρο των χαρακτηριστικών στον αρχικό χώρο. Κατά τη διάρκεια της εκπαίδευσης, ο Autoencoder προσπαθεί να μειώσει την απώλεια ανακατασκευής, που είναι η διαφορά μεταξύ των εισερχομένων δεδομένων και των ανακατασκευασμένων εξόδων. Αυτή η διαδικασία επιτρέπει στον Autoencoder να μάθει μια συμπιεσμένη αναπαράσταση των δεδομένων, αφαιρώντας τον θόρυβο και εξάγοντας τα σημαντικά χαρακτηριστικά.

Η Εικόνα 4 παρουσιάζει τη μορφή του νευρωνικού δικτύου, τύπου Autoencoder. Αποτελείται από 2 επίπεδα. Το πρώτο είναι αυτό που δέχεται την είσοδο και στην ουσία συμπιέζει και μεταμορφώνει τα δεδομένα σε ένα ενδιάμεσο χώρο. Το δεύτερο επίπεδο αποσυμπιέζει τη αναπαράσταση στον ενδιάμεσο χώρο, σε έξοδο που έχει το ίδιο μέγεθος με την είσοδο.

```
Model: "auto_encoder"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
sequential_6 (Sequential)    (None, 8)                 56
_________________________________________________________________
sequential_7 (Sequential)    (None, 6)                 126
=================================================================
Total params: 182
Trainable params: 182
Non-trainable params: 0
_________________________________________________________________
```

**Εικόνα 4. Αρχιτεκτονική νευρωνικού δικτύου Autoencoder.**

Στη συνέχεια φαίνονται τα αποτελέσματα κάθε μεθόδου που χρησιμοποιήθηκε. Ο Πίνακας 2 παρουσιάζει τις μετρικές με τις οποίες κάθε μοντέλο αξιολογείται για την απόδοση του. Συγκεκριμένα, για κάθε μοντέλο, παρουσιάζεται το Accuracy, Recall, False Positive Rate και Precision. Για την ολοκλήρωση στο Σύστημα ΜΥΡΙΩΠΟΣ επιλέξαμε τη μέθοδο που αποδίδει καλύτερα τόσο σε accuracy όσο και σε False Positive Rate. Στο ενοποιημένο Σύστημα ΜΥΡΙΩΠΟΣ επιλέξαμε το Random Forest.

Το Accuracy, δείχνει πόσο ακριβές είναι η πρόβλεψη του μοντέλου σε κάθε δείγμα. Το Recall, δείχνει πόσα ύποπτα δείγματα κατάφερε το μοντέλο να βρει από όλα τα ύποπτα δείγματα υπάρχουν στα δεδομένα. Το Precision, δείχνει πόσα από αυτά που προέβλεψε ως ύποπτα, είναι πράγματι ύποπτα. Το False Positive Rate, δείχνει το ποσοστό με το οποίο το μοντέλο προβλέπει ένα δείγμα ως ύποπτο, χωρίς ωστόσο να είναι. Αυτή η μετρική πρέπει να είναι όσο πιο μικρή όσο γίνεται.

| Model | Accuracy(%) | Recall(%) | FPR(%) | Precision(%) |
|---|---|---|---|---|
| **Random Forest** | 99.95 | 99.92 | 0.08 | 99.92 |
| **DFNN** | 100 | 100 | 0 | 100 |

| | | | | |
|---|---|---|---|---|
| **Autoencoder** | 99.4 | 98.5 | 1 | 99.6 |

**Πίνακας 2. Αξιολόγηση Μοντέλου.**

## 3.4 Μέθοδοι Επιβολής Πολιτικών Ασφάλειας και Καταστολής Επιθέσεων

Συγκεκριμένα, στο έργο ΜΥΡΙΩΠΟΣ, μέχρις στιγμής εφαρμόζονται 2 πολιτικές ασφάλειας για καταστολή των επιθέσεων. Οι πολικές αυτές εφαρμόζονται μέσω της Διεπαφής Χρήστη του ΜΥΡΙΩΠΟΣ όπως φαίνεται και στις παρακάτω εικόνες. Οι 2 πολίτικες είναι του ίδιου τύπου, και συγκείμενα είναι πολιτική **Ingress** (δηλ. εφαρμόζεται για επικοινωνία προς το μηχάνημα.) και είναι τύπου Layer 3 δηλαδή εφαρμόζεται με βάση την IP και τις πληροφορίες δικτύου. Οι 2 πολιτικές είναι στη ουσία το blocking ή unblocking μιας συγκεκριμένης IP. Ταυτόχρονα όμως, έτσι όπως χτίζεται η πολιτική μπορεί ευκολά να συμπεριλάβει και εύρος IP.

Πιο κάτω στην εικόνα φαίνεται ένα παράδειγμα του περιεχομένου μιας πολιτικής. Όπως διακρίνεται από την εικόνα, φαίνονται οι διάφορες πληροφορίες που μπορούν να προσδιοριστούν σε μια πολική, όπως για παράδειγμα το όνομα της και το namespace στο οποίο εφαρμόζεται και το οποίο καθορίζει και το εύρος που έχει ισχύει. Η πιο σημαντική πληροφορία, ωστόσο, βρίσκεται στις γραμμές 8-12, όπου καθορίζεται η κατεύθυνση της πολιτικής (ingress/egress) αλλά και το κύριο κομμάτι της πολιτικής που στη προκυμμένη περίπτωση, είναι η IP της οποίας απαγορεύεται η επικοινωνία. Το Cilium, δεν έχει άμεσο τρόπο να απαγορεύει κίνηση στο δίκτυο για μια συγκεκριμένη IP. Αυτό μπορεί να γινεί με το να αφήνει την επικοινωνία για όλες τις IP.( βλ. - cidr: 0.0.0.0/0) εκτός της IP που θα τείνει απαγόρευσης(βλ. Except : - 8.8.8.8/32). Πιο αναλυτικά, το 0.0.0.0/0 επιτρέπει όλες τις IP, ενώ το 8.8.8.8/32 υποδηλώνει μια συγκεκριμένη IP λόγω του /32. Στο παράδειγμα υποδηλώνει την 8.8.8.8. Η Εικόνα 5 παρουσιάζει τη διεπαφή με την όποια ο χρήστης αφενός, μπορεί να δει τις ύποπτες IP όπως αυτές προκύπτουν από τα ποιο πάνω μοντέλα, και αφετέρου, να εφαρμόζει τη πολιτική block (ή/και unblock) σε μια από αυτές τις IP. Αυτό μπορεί να γίνει με το να πατηθεί το κόκκινο κουμπί Block στη διεπαφή όπως φαίνεται στην εικόνα. Όλες οι IP που επηρεάζονται από τις πολιτικές εμφανίζονται στο πίνακα που βρίσκεται στο κάτω μέρος της διεπαφής (Blocked IPs).

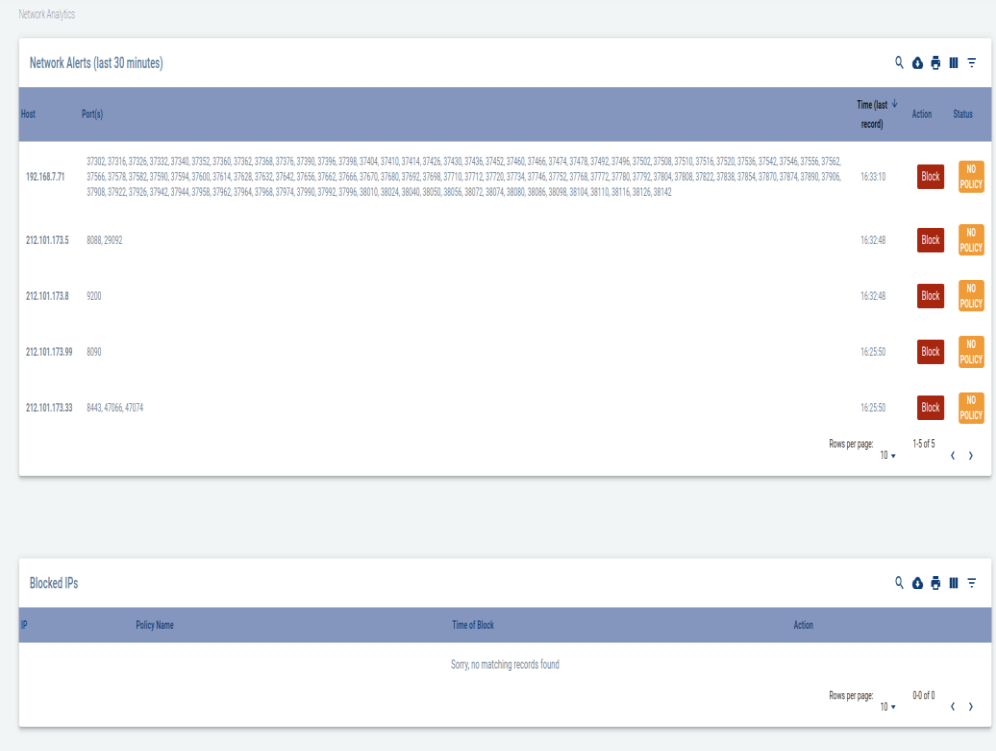
Ο Πίνακας 3 παρουσιάζει τον ορισμό των πολιτικών επιβολής μέσω ενός YAML αρχείου. Η επιλογή του namespace που θα εφαρμοστεί η πολιτική, το είδος της κίνησης στην οποία θα εφαρμοστεί (ingress/egress) και η IP που θα μπλοκαριστεί (π.χ. 8.8.8.8).

```yaml
apiVersion: cilium.io/v2
kind: CiliumNetworkPolicy
metadata:
  name: myriopos-policy
  namespace: myriopos
spec:
  endpointSelector: {}
  ingress:
    - fromCIDRSet:
        - cidr: 0.0.0.0/0
          except:
            - 8.8.8.8/32

```

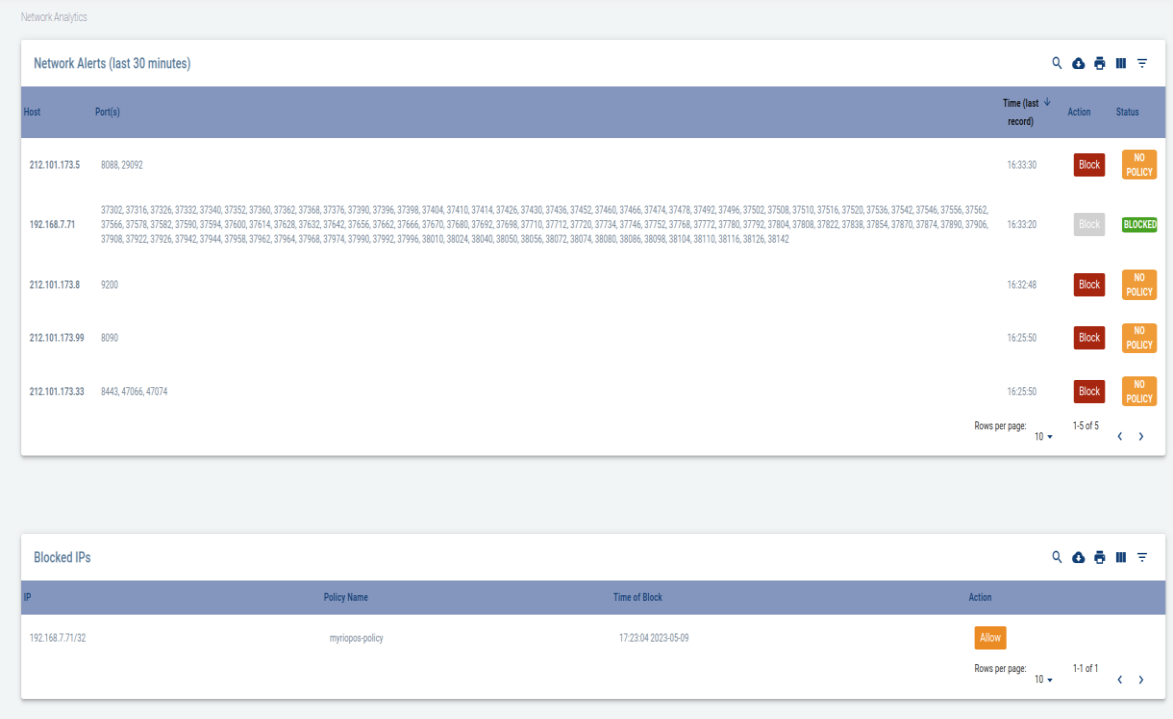
**Πίνακας 3. Ορισμός πολιτικής μέσω YAML αρχείο.**

Στην Εικόνα 5 διακρίνονται οι ύποπτες IP, τα ports που αφορούν σε κάθε IP, την ώρα αναφοράς, το κουμπί για εφαρμογή block πολιτικής (π.χ., μπλοκάρεται η συγκεκριμένη IP και port), καθώς και η κατάσταση κάθε πολιτικής. Επίσης, παρουσιάζεται η Διεπαφή Χρήστη στο Σύστημα ΜΥΡΙΩΠΟΣ. Διακρίνονται οι IP με τις οποίες το μηχάνημα έχει επικοινωνία, οι θύρες που είναι ανοικτές στην επικοινωνία, η ώρα ενημέρωσης της IP και οι επιλογή για block ή μη εφαρμογή πολιτικής. Στο πιο κάτω μέρος, φαίνεται το σύνολο των διευθύνσεων IP που έχουν μπλοκαριστεί.




**Εικόνα 5. Διεπαφή Χρήστη για Εφαρμογή Πολιτικών Ανάκαμψης.**

Στην Εικόνα 6 διακρίνεται, η Διεπαφή Χρήστη του ΜΥΡΙΩΠΟΣ, αφού μια IP γίνει blocked. Στο κάτω τμήμα του πίνακα παρουσιάζεται ως μπλοκαρισμένη ενώ παράλληλα αλλάζει η κατάστασή της και στο επάνω τμήμα του πίνακα που φαίνεται ως “BLOCKED”.




**Εικόνα 6. Διεπαφή Χρήστη – BLOCKED IP.**

Στην Εικόνα 7 φαίνεται η ενεργή απάντηση από το σύστημα που ελέγχει το ΜΥΡΙΩΠΟΣ, που υποδηλώνει ότι υπάρχει σύνδεση. Για τη δοκιμή εφαρμογής των πολιτικών, εφαρμόζουμε μια επίθεση (πριν η IP που τη δέχεται γίνει blocked). Φαίνεται αρχικά ότι η απάντηση της εικονικής μηχανής είναι ενεργή και υποδηλώνει επιτυχή επικοινωνία.

```
---------------- START ----------------
b'Hello World!'
----------------- END -----------------


Sending request... [2023-05-29 13:33:36.713849]
Response           [2023-05-29 13:33:36.713962]

---------------- START ----------------
b'Hello World!'
----------------- END -----------------


Sending request... [2023-05-29 13:33:36.735201]
Response           [2023-05-29 13:33:36.735482]

---------------- START ----------------
b'Hello World!'
----------------- END -----------------


Sending request... [2023-05-29 13:33:36.758335]
Response           [2023-05-29 13:33:36.758442]
```

**Εικόνα 7. Κατάσταση Επικοινωνίας πριν η IP που δέχεται Επίθεση γίνει blocked.**

Η Εικόνα 8 παρουσιάζει ότι δεν γίνεται σύνδεση προς το σύστημα που ελέγχει το ΜΥΡΙΩΠΟΣ, το οποίο υποδηλώνει ότι επιτυχώς η IP του επιτιθέμενου μπλοκαρίστηκε από το σύστημα. Όπως φαίνεται, δεν υπάρχει

ενεργή απάντηση από την εικονική μηχανή και διακρίνεται το “Connection Refused” που δείχνει την ανεπιτυχή επικοινωνία.

```
t=31652): Max retries exceeded with url: / (Caused by NewConnectionError('<urll
ib3.connection.HTTPConnection object at 0xffff8e1d7f10>: Failed to establish a
new connection: [Errno 110] Connection timed out'))

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "main.py", line 7, in <module>
    response = requests.get('http://212.101.173.33:31652/')
  File "/usr/lib/python3/dist-packages/requests/api.py", line 75, in get
    return request('get', url, params=params, **kwargs)
  File "/usr/lib/python3/dist-packages/requests/api.py", line 60, in request
    return session.request(method=method, url=url, **kwargs)
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 533, in requ
est
    resp = self.send(prep, **send_kwargs)
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 646, in send
    r = adapter.send(request, **kwargs)
  File "/usr/lib/python3/dist-packages/requests/adapters.py", line 516, in send
    raise ConnectionError(e, request=request)
requests.exceptions.ConnectionError: HTTPConnectionPool(host='212.101.173.33',
port=31652): Max retries exceeded with url: / (Caused by NewConnectionError('<u
rllib3.connection.HTTPConnection object at 0xffff8e1d7f10>: Failed to establish
 a new connection: [Errno 110] Connection timed out'))
```

**Εικόνα 8. Επικοινωνία αφότου η IP που δέχτηκε Επίθεση γίνει blocked.**

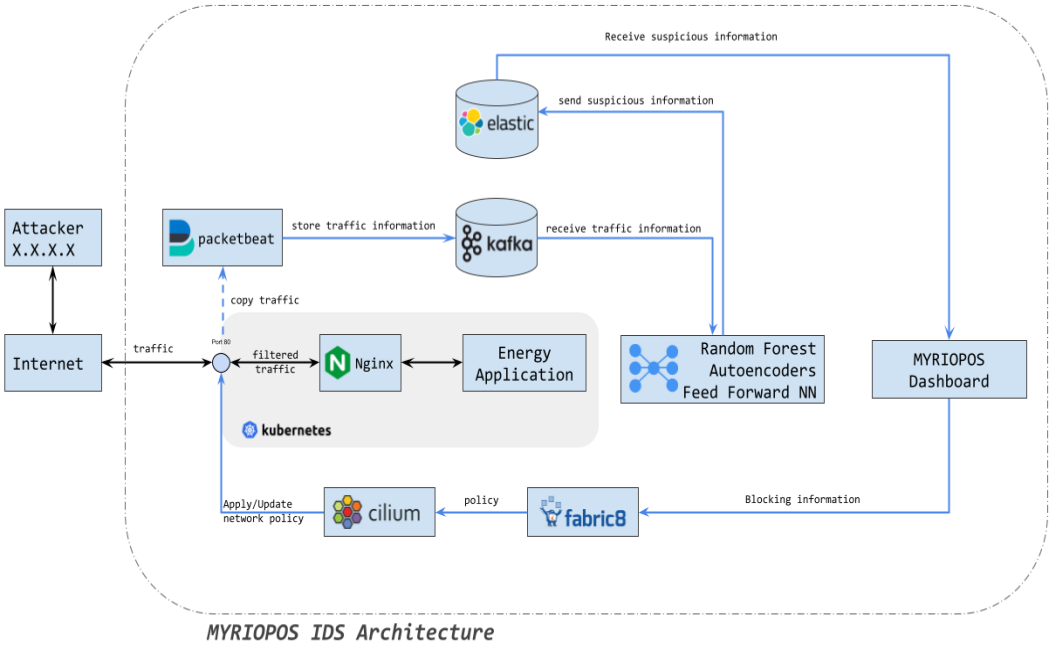
# 4 Ανίχνευση Απειλών, Δημιουργία Αναφορών και Καταστολή Συμβάντων στο Σύστημα ΜΥΡΙΩΠΟΣ

Η αρχιτεκτονική του ΜΥΡΙΩΠΟΥ παρουσιάζεται στην Εικόνα 9. Όπως φαίνεται από το σχήμα, αποτελείται από διάφορα επιμέρους τμήματα. Πρώτον, υπάρχει ένα περιβάλλον συλλογής δεδομένων που καταγράφει τα πακέτα δικτύου σε πραγματικό χρόνο ή από αποθηκευμένες ακολουθίες πακέτων (π.χ., ιστορικά δεδομένα που έχουν αποθηκευτεί). Αυτό γίνεται αφενός από το Packetbeat μου μαζεύει τα ακατέργαστα πακέτα και τα στέλνει σε μια υπηρεσία Kafka (βλ. ασύγχρονος διαχειριστής μηνυμάτων).

Αυτά τα πακέτα υπόκεινται σε προ επεξεργασία για την εξαγωγή σχετικών χαρακτηριστικών, όπως επικεφαλίδες πακέτων, χαρακτηριστικά σύνδεσης / πακέτων και στατιστικά ροής, σε μορφή ευρετηρίου με δείκτη τις IP, όπως παρουσιάστηκαν παραπάνω.

Ακολούθως τα εξαγόμενα χαρακτηριστικά τροφοδοτούν το AI μοντέλο για την πρόβλεψη ύποπτης ή μη ύποπτης σύνδεσης με μια IP. Οι ύποπτες IP αποθηκεύονται, με τη σειρά τους, σε μια υπηρεσία Elasticsearch, με σκοπό τη γρήγορη ανάκτηση τους, στην Διεπαφή Χρήστη του ΜΥΡΙΩΠΟΣ. Μέσα από τη διεπαφή, όπου φαίνονται οι ύποπτες IP, υπάρχει η επιλογή επιβολής των πολιτικών που αναφέρθηκαν πιο πάνω.

Εφόσον επιβληθεί μια πολιτική που μπλοκάρει την επικοινωνία από κάποια συγκεκριμένη IP ή εύρος IP, τότε η επικοινωνία από αυτές τις πηγές σταματάει να υφίσταται.




**Εικόνα 9. Αρχιτεκτονική ΜΥΡΙΩΠΟΣ για ανίχνευση και καταστολή απειλών.**

# 5 Επίλογος

Ο στόχος του Π3.2 ήταν να παρουσιάσει τις μεθόδους, τους μηχανισμούς και τις υπηρεσίες που αναπτύχθηκαν για την ανίχνευση απειλών σε πραγματικό χρόνο και την επιβολή πολιτικών ανάκαμψης. Στο παραδοτέο Π3.2 παρουσιάστηκε η διαδικασία εντοπισμού και ανάλυσης μέσω βαθιάς μηχανικής μάθησης (αλγόριθμοι Random Forest, DFNN, και Autoencoder). Τα δεδομένα που συλλέχθηκαν είχαν όγκο 500GB και ο χρόνος εκπαίδευσης διήρκησε μια (1) εβδομάδα. Τα δεδομένα αυτά συλλέχθηκαν σε πραγματικό χρόνο από πράκτορες του δικτύου και μέσω ορισμού πολιτικών ανάκαμψης τερματίστηκαν οι ύποπτες και επικίνδυνες ενέργειες. Τέλος παρουσιάστηκαν πειραματικά αποτελέσματα των μοντέλων AI που εκπαιδεύτηκαν και διευκρινίστηκε ότι στην τελική ενοποιημένη λύση του ΜΥΡΙΩΠΟΣ χρησιμοποιήθηκε το Random Forest γιατί αποδίδει καλύτερα τόσο σε accuracy όσο και σε False Positive Rate.

Το παραδοτέο Π3.2 ολοκληρώνει τόσο σε τεχνικό όσο και σε επίπεδο αναφοράς την ΕΕ3 Κατανεμημένη Βαθιά Ανάλυση και Νοημοσύνη Κυβερνοαπειλών. Συμπεραίνουμε ότι δεν είναι κοινό να αποκτήσεις γνώση για τις πιθανές κυβερνο-επιθέσεις σε δικτυακές υποδομές, υπηρεσίες υπολογιστικού νέφους και συσκευές άκρου, αλλά όταν υπάρχει η δυνατότητα να συλλεχθούν δεδομένα πραγματικού χρόνου από τις ίδιες τις εφαρμογές με ακριβείς καταγραφές μπορείς κανείς να εκπαιδεύσει σύνθετους αλγορίθμους, να πάρει ασφαλή αποτελέσματα και να οδηγηθεί σε σωστές αποφάσεις καταστολής, τερματισμού ή μπλοκαρίσματος με αυτόματο τρόπο.

# 6 Αναφορές

[1] Velan, P., Čermák, M., Čeleda, P., & Drašar, M. (2015). A survey of methods for encrypted traffic classification and analysis. International Journal of Network Management, 25(5), 355-374.

[2] Gümüşbaş, D., Yıldırım, T., Genovese, A., & Scotti, F. (2020). A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems. IEEE Systems Journal, 15(2), 1717-1731.

[3] Kwon, D., Kim, H., Kim, J., Suh, S. C., Kim, I., & Kim, K. J. (2019). A survey of deep learning-based network anomaly detection. Cluster Computing, 22, 949-961.

[4] Zhang, J., Zulkernine, M., & Haque, A. (2008). Random forest-based network intrusion detection systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38(5), 649-659.

[5] Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: an ensemble of autoencoders for online network intrusion detection. arXiv preprint arXiv:1802.09089.

[6] Li, Z., Qin, Z., Huang, K., Yang, X., & Ye, S. (2017, October). Intrusion detection using convolutional neural networks for representation learning. In International conference on neural information processing (pp. 858-866). Cham: Springer International Publishing.

[7] Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. Electronics, 9(8), 1295.

[8] Cohen-Addad, V., Kanade, V., Mallmann-Trenn, F., & Mathieu, C. (2019). Hierarchical clustering: Objective functions and algorithms. Journal of the ACM (JACM), 66(4), 1-42.

[9] Bank, D., Koenigstein, N., & Giryes, R. (2020). Autoencoders. arXiv preprint arXiv:2003.05991.

[10] Louppe, G. (2014). Understanding random forests: From theory to practice. arXiv preprint arXiv:1407.7502.