Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή
Στην Ασφάλεια Υπολογιστών και Δικτύων

Αντιμετώπιση Εσωτερικών Απειλών με χρήση Τεχνητής Νοημοσύνης σε ευρέως διαδεδομένο dataset

Βασίλειος Ν. Κουτσουβέλης

Επιβλέπων Καθηγητής
Δρ. Σταύρος Σιαηλής

Μάιος 2018

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Αντιμετώπιση Εσωτερικών Απειλών με χρήση Τεχνητής Νοημοσύνης σε ευρέως διαδεδομένο dataset

Βασίλειος Ν. Κουτσουβέλης

Επιβλέπων Καθηγητής
Δρ. Σταύρος Σιαηλής

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών
στην Ασφάλεια Υπολογιστών και Δικτύων

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Μάιος 2018

Περίληψη

Οι εσωτερικές απειλές, αποτελούν έναν από τους σημαντικότερους παράγοντες απώλειας, απάτης, κλοπής εμπιστευτικών ή εμπορικά πολύτιμων πληροφοριών, κλοπής πνευματικής ιδιοκτησίας ή δολιοφθοράς στα συστήματα πληροφορικής μιας Εταιρείας ή ενός Οργανισμού. Γι' αυτό το λόγο έχουν προκαλέσει το ενδιαφέρον της παγκόσμιας ερευνητικής κοινότητας, η οποία προσπαθεί να μελετήσει και να βρει αποτελεσματικούς τρόπους αντιμετώπισής τους. Ένα από τα πεδία, που εμπλέκονται στις έρευνες αυτές είναι της Τεχνητής Νοημοσύνης και των Νευρωνικών Δικτύων. Οι μέθοδοι αντιμετώπισης, που αναπτύσσονται μέσω αυτών, αποτελούν ένα νέο πεδίο για την έρευνα και έχουν ως στόχο να συμβάλουν στην έγκυρη ανίχνευση και αντιμετώπιση των εσωτερικών απειλών.

Η παρούσα μεταπτυχιακή διατριβή ερευνά την ανίχνευση των εσωτερικών απειλών, μελετώντας τα στοιχεία της δραστηριότητας των χρηστών ενός Πληροφοριακού Συστήματος για ένα ορισμένο χρονικό διάστημα και από ένα δεδομένο dataset. Αρχικά, παρουσιάζεται περιληπτικά ένα τμήμα της βιβλιογραφικής ανασκόπησης επί του ζητήματος αυτού, επιχειρείται μια σύντομη κριτική των ερευνών, που έχουν λάβει χώρα και επισημαίνεται η συμβολή της παρούσας μεταπτυχιακής διατριβής στο εν λόγο πεδίο έρευνας. Ακολουθεί το θεωρητικό τμήμα αυτής, που περιλαμβάνει δύο (2) κεφάλαια, τα οποία εστιάζουν την προσοχή τους σε δύο (2) κεντρικές περιοχές: στην Τεχνητή Νοημοσύνη και στις απειλές των πληροφοριακών συστημάτων. Στο δεύτερο μέρος της μεταπτυχιακής διατριβής, το ερευνητικό, περιγράφεται το πείραμα, που έχει διεξαχθεί σε τρία (3) στάδια: α) αρχικά με την εξαγωγή των στοιχείων της δραστηριότητας ανά χρήστη από τα αρχεία καταγραφής (log files) και την κατηγοριοποίηση αυτής μέσω ενός συνόλου προγραμμάτων σε γλώσσα προγραμματισμού Java, που δημιουργήσαμε, β) στη συνέχεια, την οπτικοποίησή της (visualization) με την εξαγωγή εικόνων μέσω της βιβλιοθήκης D3.js σε Javascript και γ) τέλος, την εκπαίδευση αλγορίθμου Convolutional Neural Network (CNN) μέσω Μηχανικής Μάθησης (Machine Learning) και την εν συνεχεία δοκιμή του για την κατηγοριοποίηση της δραστηριότητας των χρηστών και συγκεκριμένα την απάντηση στο ερώτημα εάν ένας χρήστης αποτελεί ή όχι εσωτερική απειλή για την Εταιρεία ή τον Οργανισμό. Στο τελευταίο κεφάλαιο της παρούσας μεταπτυχιακής διατριβής εξάγονται συμπεράσματα από την υλοποίηση της συγκεκριμένης μεθόδου, συγκρίνεται η εν λόγω μέθοδος με άλλες, που έχουν εφαρμοστεί και προτείνονται βελτιώσεις, οι οποίες θα συμβάλλουν στην εξέλιξη της έρευνας στο συγκεκριμένο πεδίο.

Summary

Insider threats are one of the most important factors of loss, fraud, theft of confidential or commercially valuable information, theft of intellectual property or sabotage of a Company's or an Organization's computer systems. For this reason they have attracted the interest of the global research community, which is trying to study them and find effective ways to deal with them. One of the fields which are involved in these researches is the Artificial Intelligence and Neural Networks. The countermeasures developed through them constitute a new field for research and their purpose is to contribute to prompt detection and response to internal threats.

This paper explores the detection of insider threats by studying the elements of the users' activity of an Information System for a certain period of time and for a given dataset. Initially, a section of the bibliographic review on this issue is briefly presented, a brief critique of the investigations that have taken place is attempted and the contribution of this paper to this field of research is pointed out. The theoretical part of this paper follows, which includes two (2) chapters, focusing on two (2) central regions: Artificial Intelligence and information systems' threats. In the second part of this paper - the research -, the experiment is described and carried out in three (3) stages: a) first by extracting activity data per user from the log files and categorizing it through a set of Java programs (b) then by visualizing it with the extraction of images through the D3.js library in Javascript; and (c) finally by training the Machine Learning Convolutional Neural Network (CNN) algorithm and testing it for the classification of users' activity, in order to answer the question whether or not a user is an insider threat to the Company or to the Organization. In the last chapter of this paper we draw conclusions from the implementation of this method, we compare this method with others that have been implemented and we propose improvements that will contribute to the development of the research in this field.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω ολόψυχα τον Καθηγητή μου κο Σταύρο Σιαηλή, αφ' ενός μεν για την προτροπή του να ασχοληθώ με τη συγκεκριμένη θεματική, αφ' ετέρου δε για την γενική του επίβλεψη, τον φίλο μου Γιώργο Θεοδωράκη για την πολύτιμη βοήθειά του και τέλος την οικογένειά μου και ιδιαίτερα τη σύζυγό μου Ιωάννα για τη στήριξη και την συμπαράστασή της, καθ' όλη τη διάρκεια του εν λόγω μεταπτυχιακού προγράμματος.

Περιεχόμενα

Πίνακας γραφικών παραστάσεων

Πίνακας εικόνων

Πίνακες

Συντομογραφίες

Συντομογραφίες	Ορισμοί
DNN	Deep Neural Networks
RNN	Recurrent Neural Networks
GBAD	Graph-Based Anomaly Detection
SA	Structural Anomaly Detection
PP	Psychological Profiling
GB	Giga Bytes
CNN	Convolutional Neural Network
EDVAC	Electronic Discrete Variable Automatic Computer
ICT	Security: Information and Communication Technology Security
CERT	Computer Emergency Response Team
SEI	Software Engineering Institute
DHS	Department of Homeland Security
FBI	Federal Bureau of Investigation
DOM	Document Object Model
SVG	Scalable Vector Graphics
ReLU	Rectified Linear Unit

Κεφάλαιο 1
Βιβλιογραφική Ανασκόπηση

1.1 Εισαγωγή

Είναι γεγονός ότι οι εσωτερικές απειλές, αποτελούν έναν από τους πιο σημαντικούς παράγοντες κινδύνου για το Πληροφοριακό Σύστημα και τα αγαθά μιας Εταιρείας ή ενός Οργανισμού. Η πρόληψη και αντιμετώπιση των εσωτερικών απειλών, έχει προκαλέσει το ενδιαφέρον της παγκόσμιας ερευνητικής ακαδημαϊκής κοινότητας, η οποία πραγματοποιεί σημαντικές έρευνες στον τομέα αυτό. Σκοπός της παρούσας βιβλιογραφικής ανασκόπησης είναι η παρουσίαση ορισμένων ερευνών στο εν λόγω πεδίο και η περιγραφή των πλεονεκτημάτων και μειονεκτημάτων τους, η κατάδειξη πιθανών κενών και προβλημάτων, καθώς και οι τρόποι αντιμετώπισης αυτών. Ο τελικός στόχος είναι να καταδειχθεί ο τρόπος, με τον οποίο η παρούσα έρευνα μπορεί να συμβάλει στην ευρύτερη ερευνητική κοινότητα για την αντιμετώπιση του κινδύνου των εσωτερικών απειλών.

1.2 Μέθοδοι ανίχνευσης εσωτερικών απειλών

Οι Breier και Branisova (Breier, Branisova 2015) προτείνουν μία μέθοδο ανίχνευσης απειλών, η οποία βασίζεται σε τεχνικές εξόρυξης δεδομένων για την ανάλυση των αρχείων καταγραφής του συστήματος (log analysis). Η προσέγγισή τους χρησιμοποιεί την τεχνική «Apache Hadoop», που επιτρέπει την επεξεργασία μεγάλου όγκου δεδομένων με παράλληλο τρόπο. Η μέθοδος αυτή ανιχνεύει νέους τύπους παραβιάσεων χωρίς την περαιτέρω ανθρώπινη παρέμβαση, ενώ το συνολικό ποσοστό σφάλματος κυμαίνεται κάτω από 10%. Σύμφωνα με τους μελετητές, το κύριο πλεονέκτημα αυτής της μεθόδου είναι η ικανότητά της να αποκαλύπτει νέους τύπους παραβιάσεων και να ελαχιστοποιεί την ανάγκη χειροκίνητης παρέμβασης. Οι ερευνητές διερεύνησαν επίσης τις δυνατότητες για αποτελεσματικότερη ανάλυση των μεγάλων όγκων δεδομένων λόγω του μεγέθους και της ποσότητας των αρχείων καταγραφής, τα οποία έχουν αυξανόμενη τάση. Με την εφαρμογή της τεχνολογίας «Apache Hadoop» δημιούργησαν ένα μοναδικό σύμπλεγμα κόμβων για παράλληλη επεξεργασία λογαρίθμων χρησιμοποιώντας τη μέθοδο «Map Reduce». Ο χρόνος, που απαιτήθηκε για την ανάλυση των αρχείων καταγραφής (log files) αυξήθηκε σημαντικά και ο αλγόριθμος, που εφαρμόστηκε στην μέθοδο «Hadoop» ήταν σε θέση να επεξεργάζεται δεδομένα ταχύτερα από τον κανονικό αλγόριθμο, που χρησιμοποιεί την δομή που βασίζεται σε δομή δέντρου. Επίσης, προσθέτοντας περισσότερους κόμβους υπολογιστών βελτιώθηκε ο χρόνος επεξεργασίας.

Οι ίδιοι ως άνω μελετητές σε μεταγενέστερη μελέτη τους (Breier, Branisova 2016) υπογραμμίζουν ότι σε σύγκριση με την υλοποίησή της στην Java, η εκτέλεση της Apache Hadoop εκτελείται πάνω από δέκα (10) φορές πιο γρήγορα και σημειώνουν ότι άλλη μια βελτιστοποίηση αφορά στην μετατροπή των δεδομένων σε δυαδική μορφή, με την οποία καθίσταται πιο αποτελεσματική η ανάλυση ανταλλαγής τους, ενώ επισημαίνουν ότι μελλοντικά θα ήθελαν να διερευνήσουν τις δυνατότητες προσδιορισμού συσχετίσεων μεταξύ διαφόρων συσκευών δικτύου με αυτοματοποιημένες μεθόδους.

Από την άλλη, οι Legg, Buckley, Goldsmith και Creese (Legg, Buckley, Goldsmith, Creese 2015) προτείνουν ένα σύστημα ανίχνευσης, απειλών εταιρικού χαρακτήρα, που ονομάζουν «Corporate Insider Threat Detection (CITD)», το οποίο είναι το αποτέλεσμα ενός διεπιστημονικού ερευνητικού έργου, που ενσωματώνει τεχνικές και συμπεριφορικές δραστηριότητες για την αξιολόγηση των απειλών, που προκαλείται από πρόσωπα. Συγκεκριμένα, το σύστημα αναγνωρίζει τους χρήστες και τα προφίλ που βασίζονται σε ρόλους και μετρά τον τρόπο με τον οποίο οι χρήστες αποκλίνουν από τις παρατηρούμενες συμπεριφορές τους για να εκτιμήσει την πιθανή απειλή που ενδέχεται να αποτελέσει μια σειρά από δραστηριότητες ενός χρήστη. Το εν λόγω σύστημα ανίχνευσης απειλών χρησιμοποιεί μια παράλληλη γραφική απεικόνιση συντεταγμένων και ενσωματώνει μια λίστα ειδοποιήσεων για να καταγράψει τους χρήστες, που επισημαίνονται από το σύστημα. Αυτό επιτρέπει στον αναλυτή να εκτιμήσει όχι μόνο ότι ένας χρήστης αποτελεί απειλή, αλλά επίσης και για ποιό λόγο, λαμβάνει χώρα αυτό. Επίσης, το ίδιο σύστημα υποστηρίζει έναν ενεργό βρόγχο εκμάθησής του, με τον οποίο ο αναλυτής μπορεί να δεχτεί ή να απορρίψει αποτελέσματα από τον κατάλογο προειδοποιήσεων, ο οποίος, στη συνέχεια, τροφοδοτεί το μοντέλο ανίχνευσης. Σύμφωνα με τους συγγραφείς, η μέθοδος αυτή θα πρέπει να αναπτυχθεί σε ένα αυτόνομο μηχάνημα έτσι ώστε να μην υπάρχει σύγκρουση ή παρέμβαση σε καμία υπάρχουσα υποδομή στον οργανισμό. Μάλιστα, προκειμένου να ελεγχθεί η εν λόγω προσέγγιση οι ερευνητές χρησιμοποίησαν τεχνητά σύνολα δεδομένων, που δημιουργήθηκαν από την CMU/CERT καθώς επίσης και εσωτερικά σύνολα δεδομένων, που αναπτύχθηκαν ξεχωριστά από την εργασία ανίχνευσης. Οι ερευνητές υπογραμμίζουν ότι γνωρίζουν πώς πρέπει να υλοποιηθεί το σύστημα, προκειμένου να ανιχνεύει κακόβουλες συμπεριφορές, που μπορεί να αποτελέσουν απειλή για έναν οργανισμό. Ωστόσο, επισημαίνεται ότι ο περιορισμός πολλών υφιστάμενων συστημάτων ανίχνευσης αθέμιτων πληροφοριών βρίσκεται στο ψευδώς θετικό ποσοστό. Για το λόγο αυτό και προκειμένου να μετριάσουν αυτό ενσωματώνουν μια ημι-εποπτευόμενη μαθησιακή προσέγγιση, η οποία αποκαλείται ως «ενεργός μάθηση», που επιτρέπει να ενσωματώνουν διαισθητικά την γνώση πίσω στο σύστημα,

βοηθώντας έτσι τον αλγόριθμο ανίχνευσης να βελτιώσει τον τρόπο εκτέλεσης της υποκείμενης ρουτίνας ανίχνευσης, σύμφωνα με το επιθυμητό αποτέλεσμα του χρήστη.

Σε έτερη μελέτη, οι Sanzgiri και Dasgupta (Sanzgiri, Dasgupta 2016) παρουσιάζουν εν συντομία τις τεχνικές, που έχουν αναπτυχθεί για την ανίχνευση των εσωτερικών απειλών, με αναφορά των μελετητών και των τεχνικών, που οι τελευταίοι χρησιμοποίησαν. Ειδικότερα, επισημάνθηκε ότι ο Hu πρότεινε την μοντελοποίηση των διαδικασιών και των δραστηριοτήτων των χρηστών για την επισήμανση της αποκλίνουσας συμπεριφοράς και χρησιμοποίησαν μεθόδους ελέγχου πρόσβασης, βάσει ρόλων για να δημιουργήσουν κανόνες αποκλίνουσας συμπεριφοράς για κάθε ρόλο. Ο Giordano, από την άλλη πλευρά, παρακολούθησε τη συμπεριφορά των χρηστών και την ανέλυσε προκειμένου να επαληθεύσει, εάν η συμπεριφορά αυτή είναι η αναμενόμενη. Άλλοι μελετητές πρότειναν την χρήση παγίδων honeypots και «δολωμάτων» για να παγιδεύουν και να ανιχνεύουν κακόβουλες εσωτερικές απειλές. Ο Armstrong χρησιμοποιεί το τρίγωνο Τεχνολογία, Διαδικασία και Άνθρωποι (TPP - Technology, Process and People) για να προτείνει μια τεχνική που χρησιμοποιεί δεδομένα κοινωνικών μέσων για την ανίχνευση των εσωτερικών απειλών. Ο Kandias επέκτεινε το μοντέλο για να εντοπίσει τους χρήστες, που είναι επιρρεπείς σε τέτοιες επιθέσεις και, συνεπώς, να προτείνει πρόσθετη παρακολούθηση αυτών των χρηστών. Ο Maybury χρησιμοποίησε συμπεριφορές χρηστών πέραν των συνηθισμένων ορίων, όπως για παράδειγμα απότομη οικονομική ευρωστία, συχνές μετακινήσεις στο εξωτερικό ως ενδεικτικό των αντιθετικών τάσεων στο μοντέλο της αντιφατικής συμπεριφοράς. Στο ίδιο πνεύμα κινήθηκε και ο Greitzer, ο οποίος χρησιμοποίησε αυτό το αντιφατικό μοντέλο για να προβλέψει τις απειλές εκ των έσω, χρησιμοποιώντας ένα συνδυασμό από ψυχολογικές ενδείξεις και ανωμαλίες πάνω στη χρήση του συστήματος. Ο τελευταίος χρησιμοποίησε μελέτες περίπτωσης και έρευνες για διάφορους παράγοντες, που σχετίζονται με πρόδρομες εκδηλώσεις συμπεριφοράς ατόμων, που διαπράττουν εγκλήματα στον κυβερνοχώρο. Ο Bishop χρησιμοποίησε μια παρόμοια προσέγγιση, αλλά επικεντρώθηκε περισσότερο στις διεργασίες των χρηστών και τα αντικείμενα μέσω της ανάλυσης ανεκτικών σφαλμάτων (FTA). Ο Axelrad χρησιμοποίησε μεθόδους στατιστικής ανάλυσης βασιζόμενες σε διαδικασίες του Beyes (Bayesian Networks) για να διαμορφώσει την αντιφατική συμπεριφορά και ισχυρίστηκε ότι πέτυχε μεγαλύτερη ακρίβεια στις προβλέψεις ενώ ο Nurse ανέπτυξε ένα πλαίσιο για τον χαρακτηρισμό των εσωτερικών επιθέσεων, το οποίο βασίστηκε σε πραγματικά γεγονότα για τον εντοπισμό βασικών χαρακτηριστικών των εσωτερικών απειλών. Ο Eldardiry ανέφερε ότι οι τρέχουσες τεχνικές ανίχνευσης είναι υπερβολικά εξειδικευμένες και δεν λαμβάνουν υπόψη τους ότι τα δεδομένα παρακολούθησης της συμπεριφοράς είναι ασαφή, γεγονός που έχει ως αποτέλεσμα ψευδή αρνητικά

αποτελέσματα. Ο Chinchani παρουσίασε μια θεωρία σχετικά με την αξιολόγηση των εσωτερικών απειλών, διαμορφώνοντας την οπτική του χρήστη για την οργάνωση και τις δυνατότητες της. Μια σημαντική πτυχή αυτής της εργασίας έγκειται στην αφαίρεση διαφόρων διαύλων επικοινωνίας και ως εκ τούτου καθίσταται ευέλικτη να αναπτύξει τρόπους για την ανίχνευση εσωτερικών απειλών. Οι Kramuller και Probst χρησιμοποίησαν οργανωτικές δομές για να προσδιορίσουν τις αλληλουχίες εκείνες που οδηγούν στις παραβιάσεις της πολιτικής ασφαλείας και να συμβάλουν στην πρόβλεψη εσωτερικών επιθέσεων με τον έλεγχο αυτών των παραβιάσεων. Τέλος, οι μελετητές σημειώνουν ότι ένας από τους λόγους για τους οποίους εξακολουθεί να είναι δύσκολος ο εντοπισμός των επιθέσεων από εσωτερικούς χρήστες είναι η έλλειψη αρκετών πραγματικών διαθέσιμων δεδομένων για την κατασκευή και τη δοκιμή μοντέλων και μηχανισμών ανίχνευσης εσωτερικών απειλών.

Περαιτέρω, οι Tuor, Kaplan και Hutchinson (Tuor, Kaplan, Hutchinson 2017) αναφέρονται σε ένα σύστημα βαθειάς γνώσης για το φιλτράρισμα των δεδομένων των αρχείων καταγραφής ενός συστήματος (logdata) και την εξέτασή τους από αναλυτή. Κατά τους συγγραφείς, επειδή η συμπεριφορά από εσωτερικές απειλές ποικίλλει ευρέως, ο μελετητής δεν επιχειρεί να διαμορφώσει σαφώς το μοντέλο της συμπεριφοράς, που αποτελεί απειλή. Αντ' αυτού, νέες παραλλαγές των Νευρωνικών δικτύων (DNN) και των επαναλαμβανόμενων νευρωνικών δικτύων (RNN) εκπαιδεύονται ώστε να αναγνωρίζουν την δραστηριότητα, που είναι χαρακτηριστική για κάθε χρήστη σε ένα δίκτυο και ταυτόχρονα αξιολογούν, εάν η συμπεριφορά του χρήστη είναι φυσιολογική ή ύποπτη. Όλα αυτά βέβαια συμβαίνουν σε πραγματικούς χρόνους ούτως ώστε το εν λόγω μοντέλο να προσαρμόζεται στα μεταβαλλόμενα πρότυπα των δεδομένων. Οι συγγραφείς σημειώνουν ότι η ανίχνευση κακόβουλων περιστατικών είναι ιδιαίτερα δύσκολη επειδή οι επιτιθέμενοι προσπαθούν συχνά να μιμηθούν στενά μια τυπική συμπεριφορά ενός φυσιολογικού χρήστη, καθ' όσον εξαιτίας του γεγονότος ότι η εσωτερική απειλή λαμβάνει νέες και διαφορετικές μορφές, δεν είναι πρακτικό να μοντελοποιηθεί ρητά. Η προσέγγιση αυτή έχει σχεδιαστεί για να υποστηρίζει το σενάριο συνεχούς ροής, επιτρέποντας το φιλτράρισμα ροών μεγάλου όγκου δεδομένων σε ένα διαχειρίσιμο αριθμό γεγονότων, τα οποία μπορούν να ελέγξουν οι αναλυτές. Περαιτέρω, τα αποτελέσματα των ελέγχων, επιτρέπουν στο συγκεκριμένο σύστημα να μεταδώσει γιατί αισθάνθηκε ότι ένας δεδομένος χρήστης ήταν κακόβουλος σε μια δεδομένη ημέρα (π.χ. επειδή ο χρήστης είχε έναν μη φυσιολογικό αριθμό μεταφορτώσεων αρχείων μεταξύ 18:00 και 12:00). Η ερμηνεία αυτή δόθηκε προκειμένου να βελτιωθεί η ταχύτητα και η ακρίβεια της ανάλυσης. Εκτός αυτού, μια άλλη ελπιδοφόρα διάσταση που επισημάνθηκε ήταν η εξερεύνηση διαφορετικών λεπτομερειών στον χρόνο.

Συγκεκριμένα, η τρέχουσα μελέτη αθροίζει ημερήσια χαρακτηριστικά από μεμονωμένους χρήστες. Με τον τρόπο αυτό επιτυγχάνεται η δυνατότητα να παραλειφθούν τα κακόβουλα σχέδια, που συμβαίνουν μέσα σε μια μέρα. Το εν λόγω μοντέλο – σύμφωνα με τους συγγραφείς - έχει τη μεγαλύτερη δυνατότητα γενίκευσης και θα μπορούσε να εφαρμοστεί σε μεμονωμένα γεγονότα/δεδομένα καταγραφής, χρησιμοποιώντας την κρυφή του κατάσταση ως μνήμη για την ανίχνευση ακολουθιών από κακόβουλες πράξεις. Με τον τρόπο αυτό θα μπορούσε επίσης να μειώσει δραματικά το σύνολο των μεμονωμένων γεγονότων που ένας αναλυτής πρέπει να ερευνήσει για να προσδιορίσει εάν η κακόβουλη συμπεριφορά συνιστά εσωτερική απειλή.

Πλην αυτών, οι Eberle και Holder (Eberle, Holder) χρησιμοποιούν προσεγγίσεις βασισμένες σε γραφήματα για την ανεύρεση κακόβουλων περιπτώσεων σε δομικά πρότυπα δεδομένων, που αντιπροσωπεύουν δραστηριότητα εσωτερικής απειλής. Οι προσεγγίσεις, που παρουσιάζονται αναζητούν δραστηριότητες, που εμφανίζουν ομοιότητες με τις φυσιολογικές συναλλαγές δεδομένων, αλλά στην πραγματικότητα είναι δομικά διαφορετικές από αυτές. Οι μελετητές υπογραμμίζουν τη χρησιμότητα της εφαρμογής προσεγγίσεων θεωρητικών γραφημάτων για την ανακάλυψη ύποπτης εσωτερικής δραστηριότητας, σε τομείς όπως οι αλληλογραφίες ηλεκτρονικού ταχυδρομείου, οι επιχειρηματικές διαδικασίες και το έγκλημα στον κυβερνοχώρο ενώ στη συνέχεια η μελέτη αναφέρεται σε μελλοντική έρευνα, που ασχολείται με το χειρισμό δυναμικών γραφημάτων, καθώς και με την εφαρμογή αυτών των αλγορίθμων ανίχνευσης κακόβουλων συμπεριφορών, που βασίζονται σε γραφήματα. Οι συγγραφείς υποστηρίζουν ότι η ικανότητα εξαγωγής δεδομένων για παράνομη δραστηριότητα καθίσταται δύσκολη λόγω της μιμητικότητας του χρήστη. Σημειώνουν ότι τα τελευταία χρόνια, οι εταιρείες αναλύουν τις λειτουργίες και τις διεργασίες τους στον τομέα της πληροφορικής με σκοπό την αποκάλυψη εσωτερικών απειλών, συνεπειών και εγκλήματος στον κυβερνοχώρο. Οι περισσότερες προσεγγίσεις χρησιμοποιούν είτε στατιστικά, είτε οπτικοποίηση των πόρων τους, τους οποίους μπορούν να παρακολουθούν για παράνομη πρόσβαση ή είσοδο. Μία πιθανή οδός για την ανίχνευση εσωτερικών απειλών σε δομικά περίπλοκα δεδομένα μπορεί να αποτελέσει η ανίχνευση κακόβουλης συμπεριφοράς, που βασίζεται σε γραφήματα. Η ιδέα πίσω από την προσέγγιση, που χρησιμοποιήθηκε είναι η εύρεση κακόβουλων συμπεριφορών σε δεδομένα, που βασίζονται σε γράφημα όπου η παράνομη υποδομή είναι μέρος, επισυνάπτεται ή απουσιάζει από ένα κανονιστικό πρότυπο, το οποίο στην εν λόγω εφαρμογή ελαχιστοποιεί την περιγραφή μήκους (MDL) ενός γραφήματος. Η μέθοδος αυτή είναι μια προσέγγιση χωρίς επίβλεψη, που στηρίζεται στην μέθοδο εντοπισμού γνώσης με βάση τη γραφική παράσταση «SUBDUE». Χρησιμοποιώντας τρεις (3) αλγόριθμους ανίχνευσης απειλών μέσω της παράστασης «SUBDUE»

αναζητείται η καλύτερη υποδομή ή ένα κανονιστικό μοτίβο σε ένα γράφημα εισόδου. Στα πλεονεκτήματα της μεθόδου αναφέρεται το μεγάλο ποσοστό επιτυχίας εύρεσης των απειλών (ποτέ κάτω από 95%), με ελάχιστες περιπτώσεις σφάλματος (καμία στις περισσότερες περιπτώσεις). Στον αντίποδα, οι μελετητές αναφέρουν πως η δυνατότητα ανεύρεσης κακόβουλων ενεργειών περιορίζεται μερικές φορές από τους πόρους, που κατανέμονται στον αλγόριθμο. Με δεδομένο ένα γράφημα όπου η ανώμαλη υποδομή αποκλίνει ελάχιστα από το κανονιστικό πρότυπο, εάν παρέχεται επαρκής χρόνος επεξεργασίας και μνήμη, όλοι αυτοί οι αλγόριθμοι θα ανακαλύψουν την μη φυσιολογική υποδομή στο δοθέν γράφημα χωρίς σφάλμα. Ωστόσο, η δυνατότητα ανίχνευσης ανωμαλιών (ανά ορισμό) παρεμποδίζεται επίσης από την ποσότητα θορύβου που υπάρχει στο γράφημα.

Οι ίδιοι ως άνω μελετητές σε έτερη μελέτη τους (Eberle, Holder 2009) σημειώνουν ότι η μέθοδος αυτή ανακαλύπτει κακόβουλες ενέργειες δομικών προτύπων σε δεδομένα, που αντιπροσωπεύουν οντότητες, σχέσεις και ενέργειες και σημειώνουν ότι αναπτύχθηκαν νέοι αλγορίθμοι για την ανάλυση υποσυνόλων γραφημάτων με σκοπό την αποκάλυψη και των τριών (3) τύπων κακόβουλων ενεργειών με βάση το γράφημα: τροποποιήσεις, εισαγωγές και διαγραφές. Επισημαίνουν ότι προκειμένου να ανακαλυφθεί κάθε ένας από τους πιθανούς κακόβουλους τύπους, υλοποιήθηκαν τρεις (3) αλγόριθμοι σε ένα σύστημα που το ονομάζουν GBAD. Έκαστος εξ αυτών έχει σκοπό την ανακάλυψη ενός συγκεκριμένου τύπου κακόβουλης συμπεριφοράς. Ως πλεονέκτημα της μεθόδου αναφέρεται ότι τα πειράματα με τη μέθοδο «GBAD» σε προσομοιωμένα σύνολα δεδομένων έχουν δείξει ποσοστό σχεδόν 100% αποκάλυψης για κάθε αλγόριθμο σε γραφήματα, που ποικίλλουν σε μέγεθος, ενώ η ακρίβεια και ο χρόνος λειτουργίας των αλγορίθμων σχετίζονται με το μέγεθος του χώρου αναζήτησης. Από την άλλη, σημειώνεται ότι στο μέλλον, οι βελτιώσεις για την επίδοση της μεθόδου θα επικεντρωθούν στην μείωση του χρόνου που δαπανάται στον υπολογισμό και την ταυτόχρονη επεξεργασία των δεδομένων για το GBAD.

Άλλη μέθοδος, που προτείνεται από τους Brdiczka, Liu, Price, Shen, Patil, Chow, Bart και Ducheneaut (Brdiczka, Liu, Price, Shen, Patil, Chow, Bart, Ducheneaut) συνδυάζει την Δομική Ανίχνευση απειλών (SA) από κοινωνικά και πληροφοριακά δίκτυα και το Ψυχολογικό Προφίλ (PP) των ατόμων. Η πρώτη χρησιμοποιεί τεχνολογίες, όπως η ανάλυση γραφημάτων, η δυναμική παρακολούθηση και η εκμάθηση μηχανών για την ανίχνευση δομικών κακόβουλων ενεργειών σε ευρείας κλίμακας δεδομένα δικτύου, ενώ το δεύτερο κατασκευάζει δυναμικά ψυχολογικά προφίλ από συμπεριφορικά πρότυπα. Οι απειλές εντοπίζονται τελικά από τον συνδυασμό και την κατάταξη των αποτελεσμάτων των ανωτέρω στοιχείων. Σε αυτή την μελέτη, η έρευνα δεν

λαμβάνει χώρα μετά το συμβάν αλλά επιδιώκεται ο προληπτικός εντοπισμός της κακόβουλης πρόθεσης, πριν αυτή η πρόθεση πραγματοποιηθεί. Η Ανίχνευση δομικών ανωμαλιών (SA) εξάγει δομικές πληροφορίες από ευρείας κλίμακας δεδομένα πληροφοριακού δικτύου πληροφοριών (κοινωνικά δίκτυα, μηνύματα, επισκέψεις στο διαδίκτυο κ.λπ.). Καθορίζει τις ομοιότητες μεταξύ των ατόμων, τα φυσικά πρότυπα και τις κακόβουλες ενέργειες και χρησιμοποιεί τεχνολογίες, που περιλαμβάνουν ανάλυση γραφημάτων και μηχανική μάθηση για την ανίχνευση ανωμαλιών στα δεδομένα. Η μέθοδος «SA» περιλαμβάνει μια σειρά τεχνικών στοιχείων. Εν πρώτοις, η ανάλυση της δομής του γραφήματος ανακαλύπτει τα ειδικά χαρακτηριστικά του δικτύου πληροφόρησης και τα εκμεταλλεύεται για αποτελεσματική αναπαράσταση δεδομένων και μαζική ελάττωσή τους. Δεύτερον, μετατρέπει δεδομένα από την γραφική παράσταση σε ένα χώρο χαρακτηριστικών, το οποίο είναι πιο βολικό για μεθόδους μηχανικής μάθησης.

Το ψυχολογικό προφίλ (PP) δημιουργεί ένα δυναμικό ψυχολογικό μοντέλο, που κατασκευάζεται και ενημερώνεται από τα συμπεριφορικά πρότυπα και τα δομικά δεδομένα του δικτύου πληροφοριών, το οποίο κατασκευάζει ψυχολογικά προφίλ και ανιχνεύει ψυχολογικές ανωμαλίες, παρέχοντας έτσι σημαντικές πληροφορίες για τα δεδομένα του δικτύου πληροφοριών. Το ψυχολογικό προφίλ αποτελείται από απλή μαθηματική και στατιστική μοντελοποίηση και συμπεράσματα: συνδυάζει ένα ψυχολογικό μοντέλο και παρατηρήσεις (δομικά δεδομένα συμπεριφορικού και πληροφοριακού δικτύου) για την εξαγωγή συμπερασμάτων σχετικά με την ψυχική κατάσταση του ατόμου και την πιθανότητα επίθεσης. Η πρόκληση είναι στον τομέα της ανάπτυξης του ψυχολογικού μοντέλου και ειδικότερα, στο πώς να καταστεί ένα τέτοιο μοντέλο ακριβές και ρεαλιστικό. Οι συγγραφείς θεωρούν ότι οι ανωτέρω παράγοντες αλληλεπιδρούν. Το μεν ψυχολογικό προφίλ παρέχει τα ουσιώδη στοιχεία για την Δομική Ανίχνευση απειλών, αφαιρώντας ένα μεγάλο τμήμα που θεωρείται ότι δεν σχετίζεται με βάση την ψυχολογική σημασιολογία και επομένως βελτιώνει την επεκτασιμότητα. Ταυτόχρονα, το ψυχολογικό προφίλ μειώνει το ποσοστό ψευδούς ανάλυσης του SA, καθιστώντας την ανίχνευση απειλών πιο εύχρηστη και εφαρμόσιμη.

Τέλος, έτερη μελέτη (Young, Goldberg, Memory, Sartain, Senator 2013) των Young, Goldberg, Memory, Sartain και Senator παρουσιάζει τα πρώτα αποτελέσματα από μια εκτενή σειρά πειραμάτων για την ανίχνευση εσωτερικών απειλών, που προκύπτουν από τη χρήση μιας πραγματικής εταιρικής βάσης δεδομένων. Σε αυτήν, η έρευνα επικεντρώνεται: (1) στην επιλογή κατάλληλων αλγορίθμων για χρήση στην ανίχνευση εσωτερικών απειλών, (2) στον εντοπισμό χαρακτηριστικών που είναι ενδεικτικά της δραστηριότητας που είναι γνωστό ότι συσχετίζεται με την απειλή των εμπιστευτικών πληροφοριών και (3) στην μοντελοποίηση - κατηγοριοποίηση

περιπτώσεων σεναρίων απειλής εμπιστευτικών πληροφοριών. Παρουσιάζεται επίσης οπτικοποίηση των δεδομένων, για τον προσδιορισμό των ανωμαλιών μεταξύ διαφορετικών τύπων δεδομένων, όπως, εισαγωγές στη βάση, τους χρήστες βάσης δεδομένων και τις χρονικές περιόδους χρήσης. Τα αποτελέσματα των πειραμάτων χρονικού διαστήματος δύο (2) μηνών δείχνουν ότι αυτές οι μέθοδοι είναι ελπιδοφόρες, με αρκετά πειράματα να παρέχουν βαθμολογίες καμπύλης κοντά στο 1, 0. Οι μελετητές σημειώνουν ότι η παρούσα έρευνα διαφέρει από την προηγούμενη, καθ' όσον σε αυτή ενσωματώνεται γνώση σε πολλαπλά σημεία εκκίνησης για ανάλυση με τη μορφή εκ των προτέρων δεικτών απειλής, ανίχνευσης κακόβουλων συμπεριφορών σε πολλά μοντέλα και τύπους δεδομένων και υψηλό επίπεδο ανίχνευσης προτύπου που οφείλεται σε γνωστά ή ύποπτα σενάρια απειλών. Διαπιστώνεται ότι αυτός ο συνδυασμός σημασιολογικής και δομικής ανάλυση σε πολλαπλές κλίμακες, δηλ. από χαμηλής στάθμης ανταλλαγής δεδομένων σε υψηλού επιπέδου μοντέλα συμπεριφοράς, ανιχνεύει περιπτώσεις ρεαλιστικών σεναρίων πληροφορικής - που αποτελούνται από πολύπλοκους συνδυασμούς δραστηριοτήτων - με μεγάλη ακρίβεια. Έχοντας διαπιστώσει την αποτελεσματικότητα των τριών (3) μεθόδων για τον εντοπισμό των σημείων εκκίνησης της ανάλυσης των δεδομένων, οι συγγραφείς θεωρούν καλό την διερεύνηση τρόπων για τον συνδυασμό των αποτελεσμάτων, ως συμπληρωματική εργασία, ώστε να βοηθήσει στην περαιτέρω μείωση του σφάλματος στα αποτελέσματα.

1.3 Σύντομος σχολιασμός μελετών

Παρουσιάστηκαν ανωτέρω ορισμένες μελέτες, που αφορούν στο ζήτημα της ανίχνευσης εσωτερικών απειλών. Σε κάθε μία εξ αυτών περιέχεται η μέθοδος ανίχνευσης εσωτερικών απειλών, που χρησιμοποίησαν οι μελετητές, τα πλεονεκτήματα και μειονεκτήματά της, καθώς και τα εξαγόμενα συμπεράσματα της. Οι μελέτες αυτές κατέδειξαν ότι οι μέθοδοι για την ανίχνευση εσωτερικών απειλών βασίζονται σε πληθώρα τεχνικών, που ποικίλλουν. Σε κάθε μελέτη η εξαγωγή συμπερασμάτων είναι αυτή, που συμβάλλει στην συνέχιση της έρευνας ενός ορισμένου και συγκεκριμένου πεδίου. Θεμελιώδη μέθοδο για την ανίχνευση εσωτερικών απειλών συνιστά η χρησιμοποίηση και ανάπτυξη της Τεχνητής Νοημοσύνης. Η τελευταία παρέχει σημαντικά πλεονεκτήματα στους ερευνητές, οι οποίοι επιδιώκουν συνεχώς να ανακαλύπτουν νέους τρόπους, που θα τους οδηγήσουν σε ακόμη πιο έγκυρα και ασφαλή συμπεράσματα. Η παρούσα μεταπτυχιακή διατριβή επιδιώκει να προσθέσει ένα ακόμη στοιχείο

στον ήδη υπάρχοντα όγκο πληροφοριών για τους τρόπους αντιμετώπισης αυτού του ζητήματος με την χρησιμοποίηση της Τεχνητής Νοημοσύνης για την εξαγωγή των συμπερασμάτων της.

Το κύριο μειονέκτημα των μεθόδων που παρουσιάστηκαν, έγκειται στον μικρό όγκο των δεδομένων προς μελέτη. Για την παρούσα μεταπτυχιακή διατριβή χρησιμοποιήθηκαν δεδομένα μεγέθους 90 GB και πλέον. Περαιτέρω, στις έρευνες που διεξήχθησαν χρησιμοποιήθηκαν αλγόριθμοι, των οποίων τόσο η υλοποίηση, όσο και η εξαγωγή συμπερασμάτων από την εφαρμογή τους, αποτελεί χρονοβόρα διαδικασία λόγω της πραγματοποίησης πολλαπλών δοκιμών.

1.4 Συμβολή της παρούσας μεταπτυχιακής διατριβής

Στην παρούσα μεταπτυχιακή διατριβή, χρησιμοποιήθηκε και υλοποιήθηκε ο αλγόριθμος εκπαίδευσης CNN μέσω του προγράμματος Tensorflow της Google, ο οποίος εκπαιδεύτηκε ώστε να αναγνωρίζει πιθανές απειλές από τις εικόνες, που παρήχθησαν από τα δεδομένα του διαθέσιμου dataset. Η χρησιμοποίηση αυτής της μεθόδου επιλέχθηκε εν προκειμένω, για την αμεσότερη εξόρυξη των δεδομένων (data mining). Στην παρούσα μέθοδο δεν οριοθετήθηκαν οι ρόλοι και οι συμπεριφορές του χρήστη, σύμφωνα με συγκεκριμένους ρόλους και χαρακτηριστικά ή ακόμα και συγκεκριμένων ψυχολογικών προφίλ και κατ' αυτόν τον τρόπο αποφεύχθηκε ο κίνδυνος εξαγωγής λανθασμένων συμπερασμάτων εξαιτίας των μικρών αποκλίσεων από αυτά. Αντίθετα, παρουσιάστηκε ένας πιο άμεσος τρόπος, αυτός της οπτικοποίησης (visualization) όλων των δεδομένων, που σχετίζονται με τους υπό εξέταση χρήστες. Από την εξέταση των εικόνων αυτών με τη βοήθεια της Τεχνητής Νοημοσύνης, προέκυψαν τα τελικά συμπεράσματα για τη συμπεριφορά αυτών και συγκεκριμένα απαντήθηκε το θεμελιώδες ερώτημα, εάν η συμπεριφορά εκάστου χρήστη χαρακτηρίζεται ως «κακόβουλη» και ως «απειλή» ή όχι για το Πληροφοριακό Σύστημα.

Η μέθοδος της έρευνας, που διεξήχθη για την ανίχνευση τυχόν εσωτερικών απειλών στο διαθέσιμο dataset, ήταν η δημιουργία εικόνων από τα δεδομένα, για έναν συγκεκριμένο αριθμό χρηστών. Οι εικόνες, απεικόνιζαν τη δραστηριότητα και τη συμπεριφορά του κάθε χρήστη, όπως αυτή προέκυψε από τις ενέργειες, που ο τελευταίος είχε πραγματοποιήσει στο Πληροφοριακό Σύστημα. Οι εικόνες, που παρήχθησαν, χρησιμοποιήθηκαν για την εκπαίδευση του αλγορίθμου με τη βοήθεια της Μηχανικής Μάθησης (Machine Learning). Σκοπός ήταν να αναγνωρίσει ποιές από αυτές τις συμπεριφορές των χρηστών τις οποίες είχαμε εισαγάγει στο σύστημα συνιστούν

κακόβουλες συμπεριφορές και άρα και επικίνδυνες για το εκάστοτε Πληροφοριακό Σύστημα. Τα βήματα, που ακολουθήσαμε για την ολοκλήρωση της διαδικασίας και την εξαγωγή των συμπερασμάτων μας ήταν τα ακόλουθα:

Κατανομή των δεδομένων και δημιουργία αρχείων, τα οποία βασίστηκαν στα δεδομένα του υπό εξέταση χρήστη.

Εισαγωγή των αρχείων των δεδομένων, που δημιουργήσαμε στη βιβλιοθήκη D3.js, επιλογή κατάλληλου σχεδίου δημιουργίας εικόνων, εξετάζοντας τα διαθέσιμα σχέδια (patterns) της βιβλιοθήκης της εφαρμογής και δημιουργία των εικόνων του υπό εξέταση χρήστη, οι οποίες περιελάμβαναν τις δραστηριότητές του κατά τη διάρκεια κάθε ημέρας.

Δημιουργία των εικόνων.

Εκπαίδευση του αλγορίθμου με το πρόγραμμα Tensorflow και κατόπιν εξέταση της συμπεριφοράς των χρηστών, την οποία χαρακτηρίσαμε είτε ως «φυσιολογική», είτε ως «κακόβουλη».

Εξαγωγή συμπερασμάτων.

Κεφάλαιο 2
Τεχνητή Νοημοσύνη

2.1 Εισαγωγή

Η Τεχνητή Νοημοσύνη παρουσιάζει εντυπωσιακή εξέλιξη τα τελευταία χρόνια και - παρόλο που αποτελεί συχνά θέμα υπερβολών ή ακόμα και σεναρίων καταστροφολογίας - αρκεί να εστιάσουμε στην πραγματική τεχνολογική πρόοδο για να συνειδητοποιήσουμε τις συναρπαστικές δυνατότητές της (Κρασαδάκης 2018). Χαρακτηριστικό παράδειγμα εφαρμογών, που στηρίζεται σε τεχνολογίες τεχνητής νοημοσύνης είναι τα Τεχνητά Νευρωνικά Δίκτυα. Τα συστήματα αυτά βασίζονται στην προσπάθεια μίμησης των ιδιοτήτων των βιολογικών νευρωνικών δικτύων και ειδικότερα, στην προσπάθεια των ανθρώπων να κατανοήσουν κάποιες από τις στοιχειωδέστερες λειτουργίες και δυνατότητες του βιολογικού εγκεφάλου, όπως είναι η κωδικοποίηση των αριθμών, των λέξεων, των οντοτήτων, των εννοιών (Σχίζας, Νεοκλέους 2017).

2.2 Σύντομη Ιστορική Αναδρομή

«Λέγοντας λογισμός, εννοώ υπολογισμός», διακήρυξε ο Άγγλος φιλόσοφος Χόμπς (Thomas Hobbes, 1588 – 1679), εισάγοντας προφητικά την Τεχνητή Νοημοσύνη γύρω στο 1650 (Haugeland 2011:38). Ωστόσο, το ερώτημα περί του εάν οι υπολογιστές μπορούν να διαθέτουν νοημοσύνη – ή, γενικότερα, εάν οι μηχανές μπορούν να σκέφτονται – τέθηκε για πρώτη φορά στα τέλη της δεκαετίας του 1940.

Μία από τις πρώτες και σημαντικότερες συνεισφορές στο πεδίο της μηχανικής νοημοσύνης είναι αυτή του Βρετανού μαθηματικού Alan Turing, ο οποίος θεωρείται ένας από τους πατέρες της τεχνητής νοημοσύνης. Η εργασία του με τίτλο «Υπολογιστικές μηχανές και νοημοσύνη» δημοσιεύτηκε πριν από μισό και πλέον αιώνα (Computing machinery and intelligence, Turing, 1950). Πλην όμως, άντεξε στη δοκιμασία του χρόνου και η προσέγγιση του Turing παραμένει καθολικά εφαρμόσιμη έως σήμερα. Ο Turing δεν επιχείρησε να ορίσει τις έννοιες «μηχανή» και «νοημοσύνη». Απέφυγε να εμπλακεί στο επίπεδο της σημασιολογίας, επινοώντας ένα παιχνίδι – το περίφημο παιχνίδι της μίμησης – (imitation game), γνωστό και ως δοκιμασία Turing (Turing test). Αντί να ρωτάμε «μπορούν οι μηχανές να σκέφτονται;», πρότεινε ο Turing, θα έπρεπε να ρωτάμε «μπορούν οι μηχανές να περάσουν επιτυχώς ένα τεστ συμπεριφοράς που θα αποδεικνύει την εκδήλωση της νοημοσύνης;». Προέβλεψε μάλιστα ότι έως το έτος 2000 θα ήταν δυνατό να προγραμματιστεί ένας υπολογιστής με τρόπο ώστε, μετά από μια πεντάλεπτη

συνομιλία με άνθρωπο, να έχει 30% πιθανότητες να εξαπατήσει τον «ανακριτή» του, κάνοντάς τον να πιστέψει ότι συνδιαλέγεται με άνθρωπο και όχι με μηχανή.

Η εργασία, που άνοιξε τον δρόμο για την καθιέρωση της Τεχνητής Νοημοσύνης ως επιστημονικού πεδίου παρουσιάστηκε το 1943, από τους Warren McCulloch και Walter Pitts. Ο McCulloch, με σπουδές φιλοσοφίας και ιατρικής στο Πανεπιστήμιο Columbia, διετέλεσε επικεφαλής του εργαστηρίου Basic Research Laboratory στο τμήμα Ψυχιατρικής του Πανεπιστημίου του Illinois. Η έρευνά του πάνω στο κεντρικό νευρικό σύστημα οδήγησε στην πρώτη σημαντική συνεισφορά στο πεδίο της Τεχνητής Νοημοσύνης: ένα μοντέλο των νευρώνων του ανθρώπινου εγκεφάλου. Από τη συνεργασία του McCulloch με τον Walter Pitts, έναν νεαρό μαθηματικό, προέκυψε ένα μοντέλο τεχνητών νευρωνικών δικτύων, το οποίο αντιμετώπιζε τους νευρώνες ως μηχανισμούς δύο (2) καταστάσεων (on/off) – δηλαδή κάθε νευρώνας μπορούσε να βρίσκεται είτε σε κατάσταση ενεργοποίησης, είτε σε κατάσταση απενεργοποίησης (McCulloch and Pitts, 1943). Στα πλαίσια αυτής της εργασίας, οι McCulloch και Pitts έδειξαν ότι το μοντέλο νευρωνικού δικτύου, που πρότειναν ήταν ουσιαστικά ισοδύναμο με τη μηχανή Turing. Επιπλέον απέδειξαν ότι οποιαδήποτε υπολογίσιμη συνάρτηση θα ήταν δυνατό να υπολογιστεί μέσω ενός δικτύου συνδεδεμένων νευρώνων καθώς και ότι ακόμα και απλές δομές τέτοιων δικτύων θα μπορούσαν να διαθέτουν ικανότητα μάθησης.

Τρίτος εκ των ιδρυτών της Τεχνητής Νοημοσύνης ήταν ο επιφανής μαθηματικός John von Neumann. Γεννημένος στην Ουγγαρία υπήρξε συνάδελφος και φίλος του Alan Turing. Ο Neumann διετέλεσε Σύμβουλος στο έργο ανάπτυξης του υπολογιστή ENIAC (Electronic Numerical Integrator and Calculator) στο Πανεπιστήμιο της Pennsylvania και βοήθησε στον σχεδιασμό του EDVAC, του πρώτου υπολογιστή, που διέθετε μνήμη για την αποθήκευση προγραμμάτων. Αναγνωρίζοντας την αξία του μοντέλου νευρωνικού δικτύου των McCulloch και Pitts, ο von Neumann ενθάρρυνε και υποστήριξε ενεργά τους Marvin Minsky και Dean Edmonds όταν, σπουδαστές ακόμα στο τμήμα μαθηματικών του Princeton, κατασκεύασαν τον πρώτο βασισμένο σε νευρωνικό δίκτυο υπολογιστή, το 1951 (Negnevitsky 2018:2 επ).

2.3 Τεχνητή νοημοσύνη: Έννοια και λειτουργία της

Η αναπαράσταση της γνώσης και η προσαρμογή των σημασιολογικών περιγραφών αποτελεί πεδίο έρευνας τα τελευταία χρόνια και πολλά συστήματα έχουν προταθεί για το σκοπό αυτό. Οι δομικές διαφορές υπολογιστή και εγκεφάλου δεν εμποδίζουν την προσπάθεια να υποκατασταθούν κάποιες νοητικές λειτουργίες του εγκέφαλου από υπολογιστικό σύστημα, όπως έχει ήδη γίνει για τους αριθμητικούς υπολογισμούς. Αυτό σημαίνει ότι υπάρχει η δυνατότητα αποδοτικότερης εργασίας με την χρήση ενός δυνατού εργαλείου, που απαλλάσσει τον ανθρώπινο εγκέφαλο από χρονοβόρες νοητικές λειτουργίες επιτυγχάνοντας έτσι μεγαλύτερη ταχύτητα λογισμού. Στα πλαίσια αυτής της λογικής ξεκίνησε η πρόοδος του κλάδου της επιστήμης υπολογιστών, ο οποίος ασχολείται με τη σχεδίαση και την υλοποίηση υπολογιστικών συστημάτων, δηλαδή συστημάτων που επιδεικνύουν χαρακτηριστικά που σχετίζονται με τη νοημοσύνη στην ανθρώπινη συμπεριφορά και ονομάστηκε Τεχνητή Νοημοσύνη. Έτσι, μέσα σε μια περίοδο τεχνολογικής ανάπτυξης άρχισε να υλοποιείται η ιδέα να κατασκευαστούν μηχανές, οι οποίες θα υιοθετούσαν ανθρώπινες συμπεριφορές και θα λειτουργούσαν με λογική σκέψη (Πλέρου 2016:128 επ).

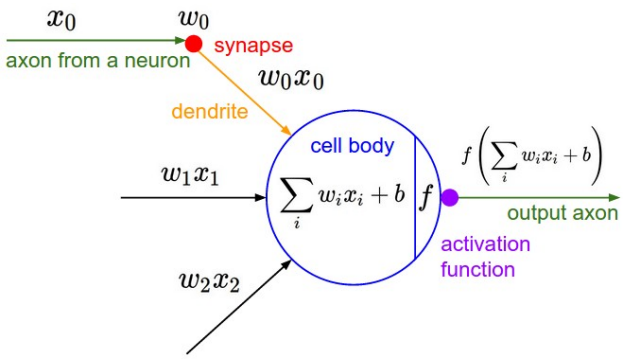
Τα πρώτα υπολογιστικά συστήματα δημιουργήθηκαν για την επεξεργασία αριθμών. Ο αριθμός είναι ένα είδος συμβόλου. Υπάρχουν πολλά άλλα είδη συμβόλων, όπως λέξεις, σχήματα, κτλ. Με το να γενικεύσουμε την «επεξεργασία αριθμών» σε «επεξεργασία συμβόλων», καταλήγουμε σε μια πολύ πιο γενική και πολύ πιο χρήσιμη θεώρηση των υπολογιστικών συστημάτων ως επεξεργαστών συμβόλων. Αυτή η θεώρηση έχει μεγάλη σημασία για την Τεχνητή Νοημοσύνη, διότι η επεξεργασία γνώσης αποτελεί αναπόσπαστο μέρος των συστημάτων Τεχνητής Νοημοσύνης. Η επεξεργασία της γνώσης μέσα στα πλαίσια υπολογιστικών συστημάτων, συνεπάγεται την αναπαράστασή της με τυπικό, συμβολικό, τρόπο. Τα προβλήματα με τα οποία καταπιάνονται συστήματα Τεχνητής Νοημοσύνης είναι συνήθως δύσκολα, τα οποία είναι αδύνατο να επιλυθούν με εξαντλητική εξέταση όλων των πιθανών (μερικών) λύσεων. Για αυτό το λόγο ένα ευφυές σύστημα πρέπει να έχει την ικανότητα να πλοηγείται, με αποτελεσματικότητα και αποδοτικότητα, σε ένα πολύ μεγάλο χώρο αναζήτησης, επιλέγοντας σε κάθε στάδιο την πιο «υποσχόμενη» διαδρομή, η οποία στην πλειονότητα των περιπτώσεων, θα το οδηγήσει, εάν όχι σε βέλτιστη λύση, τουλάχιστον σε μία αρκετά ικανοποιητική λύση. Οι μηχανισμοί διείσδυσης ή εστίασης σε ένα μεγάλο χώρο αναζήτησης (λύσεων) ονομάζονται ευρετικά. Η αναζήτηση λύσεων και η καθοδήγηση μέσω ευρετικών αποτελεί μία πολύ κεντρική έννοια στο πεδίο της Τεχνητής Νοημοσύνης. Γενικά, η επεξεργασία γνώσης μέσω συμβολικών αναπαραστάσεων και η καθοδήγηση αναζήτησης λύσεων μέσω ευρετικών είναι τα

χαρακτηριστικά, που διακρίνουν τις μεθόδους Τεχνητής Νοημοσύνης από τις υπόλοιπες υπολογιστικές μεθόδους (Κεραυνού, 2000: 16).

2.4 Νευρωνικά Δίκτυα: Ορισμός και χαρακτηριστικά

Ένα Νευρωνικό Δίκτυο μπορεί να οριστεί ως ένα μοντέλο συλλογισμών βασισμένο στη λειτουργία του ανθρώπινου εγκεφάλου. Ο εγκέφαλος αποτελείται από ένα πυκνά διασυνδεδεμένο σύνολο νευρικών κυττάρων – αποκαλούνται νευρώνες (neurons) και αποτελούν τις βασικές μονάδες επεξεργασίας πληροφορίας που διαθέτει. Ο ανθρώπινος εγκέφαλος περιλαμβάνει σχεδόν δέκα (10) δισεκατομμύρια νευρώνες και εξήντα (60) τρισεκατομμύρια συνδέσεων μεταξύ τους, οι οποίες αποκαλούνται συνάψεις (synapses) (Shepherd and Koch 1990). Χρησιμοποιώντας πολλαπλούς νευρώνες ταυτόχρονα, ο εγκέφαλος μπορεί να εκτελεί τις λειτουργίες του πολύ ταχύτερα απ' ότι οι γρηγορότεροι υπολογιστές που υπάρχουν σήμερα (Negnevitsky 2018:166).

Ένας νευρώνας είναι μια μονάδα επεξεργασίας πληροφορίας. Τα τρία (3) βασικά στοιχεία αυτού του μοντέλου είναι: α) ένα σύνολο από συνάψεις ή συνδετικούς κρίκους, β) ένας αθροιστής και γ) μια συνάρτηση ενεργοποίησης. Κάθε νευρώνας έχει πολλές εισόδους αλλά μόνο μία έξοδο, η οποία αποτελεί είσοδο για άλλους νευρώνες. Οι συνδέσεις διαφέρουν ως προς τη σημαντικότητά τους, που προσδιορίζεται από το συντελεστή βάρους (σύναψη). Η επεξεργασία κάθε νευρώνα καθορίζεται από τη συνάρτηση μεταφοράς, η οποία καθορίζει την κάθε έξοδο σε σχέση με τις εισόδους και τους συντελεστές βάρους. Κάθε νευρώνας, παριστάνεται από ένα σύνολο γραμμικών συναπτικών συνδέσεων, ένα εξωτερικά εφαρμοζόμενο κατώφλι και μια μη-γραμμική σύνδεση ενεργοποίησης. Το κατώφλι παριστάνεται από συναπτικές συνδέσεις με σήμα εισόδου τιμής -1. Οι συναπτικές συνδέσεις ενός νευρώνα ζυγίζουν τα αντίστοιχα σήματα εισόδου. Το άθροισμα των βαρών των σημάτων εισόδου καθορίζει το συνολικό εσωτερικό επίπεδο ενεργοποίησης του νευρώνα που ζητείται. Η σύνδεση ενεργοποίησης συνθλίβει (περιορίζει) το εσωτερικό επίπεδο ενεργοποίησης, για την παραγωγή της εξόδου που παριστάνει την κατάσταση του νευρώνα (http://www.icsd.aegean.gr/lecturers/kavallieratou/PattRec_files/4other_class.pdf).



Εικόνα 2-1: Απεικόνιση νευρώνα ενός Τεχνητού Νευρωνικού Δικτύου

Παρόλο, που ένα σημερινό Τεχνητό Νευρωνικό Δίκτυο μοιάζει με τον ανθρώπινο εγκέφαλο περίπου όσο ένα χάρτινο αεροπλανάκι με ένα υπερηχητικό αεροσκάφος, δεν παύει να αποτελεί ένα μεγάλο βήμα προς τα εμπρός. Τα Τεχνητά Νευρωνικά Δίκτυα έχουν τη δυνατότητα «μάθησης» - δηλαδή, χρησιμοποιούν την εμπειρία για να βελτιώνουν την απόδοσή τους. Αφού εκτεθούν σε επαρκή αριθμό δειγμάτων, τα Τεχνητά Νευρωνικά Δίκτυα μπορούν να γενικεύουν βάσει της αποκτηθείσας γνώσης έτσι ώστε να είναι σε θέση να την εφαρμόζουν και σε περιπτώσεις, που δεν έχουν συναντήσει ακόμη. Μπορούν να αναγνωρίζουν χειρόγραφους χαρακτήρες, να εντοπίζουν συγκεκριμένες λέξεις στην ανθρώπινη ομιλία, ή ακόμα και να ανιχνεύουν την ύπαρξη εκρηκτικών σε αεροδρόμια. Επιπλέον, τα τεχνητά νευρωνικά δίκτυα μπορούν να εντοπίζουν την ύπαρξη συγκεκριμένων σχηματισμών, μοτίβων – ή, γενικότερα, προτύπων (patterns), όπως είναι ευρέως γνωστά -, τα οποία οι άνθρωποι ακόμα και οι ειδήμονες αδυνατούν να αναγνωρίσουν. Για παράδειγμα, η Τράπεζα Chase Manhattan Bank χρησιμοποίησε ένα νευρωνικό δίκτυο για να εξετάσει έναν όγκο πληροφοριών σχετιζόμενων με τη χρήση κλεμμένων πιστωτικών καρτών και ανακάλυψε ότι η πιο «ύποπτη» κατηγορία αγορών με τέτοιες κάρτες ήταν τα γυναικεία υποδήματα και ειδικότερα αυτά στο εύρος τιμών μεταξύ 40 και 80 δολαρίων (Negnevitsky 2018:166 επ.).

Όλες οι μέθοδοι μάθησης μπορούν να καταταχτούν σε δύο (2) κατηγορίες: τη μάθηση με επίβλεψη (supervised learning) και τη μάθηση χωρίς επίβλεψη (unsupervised learning). Η μάθηση με επίβλεψη είναι μια διαδικασία, η οποία συνδυάζει έναν εξωτερικό εκπαιδευτή και τη συνολική ή γενικευμένη πληροφορία. Κάποιες από τις μεθόδους, οι οποίες συγκαταλέγονται σε αυτή την κατηγορία είναι η μάθηση με διόρθωση σφάλματος και η στοχαστική μάθηση. Παραδείγματα τα οποία αντιπροσωπεύουν τη μάθηση με επίβλεψη συμπεριλαμβάνουν αποφάσεις για το πότε θα πρέπει να σταματήσει η διαδικασία εκπαίδευσης, αποφάσεις αναφορικά με τη συχνότητα παρουσίασης στο δίκτυο, τα πρότυπα εκπαίδευσης και την

παρουσίαση προόδου του δικτύου. Η μάθηση με επίβλεψη χωρίζεται σε δύο (2) ακόμα κατηγορίες: στη δομική (structural) και στην προσωρινή (temporal) εκμάθηση. Οι αλγόριθμοι, οι οποίοι βρίσκονται στην πρώτη κατηγορία, χρησιμοποιούνται για την εύρεση της βέλτιστης σχέσης μεταξύ εισόδων και εξόδων για κάθε ξεχωριστό ζευγάρι προτύπων. Παραδείγματα της δομικής εκμάθησης αποτελούν η αναγνώριση και η κατηγοριοποίηση προτύπων, ενώ παραδείγματα της προσωρινής εκμάθησης η πρόβλεψη και ο έλεγχος. Από την άλλη, οι αλγόριθμοι της μάθησης χωρίς επίβλεψη αναφέρονται ως αυτό-οργανώμενοι (self-organized) και είναι διαδικασίες, οι οποίες δεν απαιτούν να είναι παρών ένας «εξωτερικός» δάσκαλος ή επιβλέπων. Βασίζονται, μάλιστα, μόνο σε τοπική πληροφορία, καθ' όλη τη διάρκεια της εκπαίδευσης του Τεχνητού Νευρωνικού Δικτύου. Οι συγκεκριμένοι αλγόριθμοι οργανώνουν τα δεδομένα και ανακαλύπτουν τις σημαντικές συλλογικές ιδιότητες. Για παράδειγμα, αλγόριθμοι εκπαίδευσης χωρίς επίβλεψη είναι ο αλγόριθμος Hebbian, ο διαφορικός αλγόριθμος Hebbian και ο Min-Max αλγόριθμος. Κατά κύριο λόγο οι περισσότερες διαδικασίες εκπαίδευσης είναι off line. Όταν χρησιμοποιείται όλο το δείγμα προτύπων για την τροποποίηση των τιμών των βαρών, πριν την τελική χρήση του δικτύου ως εφαρμογή, τότε ονομάζεται off line εκπαίδευση. Οι αλγόριθμοι εκπαίδευσης off line έχουν την απαίτηση να βρίσκονται στην εκπαίδευση του δικτύου παρόντα όλα τα πρότυπα. Το γεγονός αυτό αποκλείει την πιθανότητα εισαγωγής νέων πληροφοριών μέσω νέων προτύπων. Βέβαια, υπάρχουν και Τεχνητά Νευρωνικά Δίκτυα, τα οποία δεν αποκλείουν την εισαγωγή νέας πληροφορίας, μετά την τελική τους μοντελοποίηση. Αν παρουσιαστεί ανάγκη εισαγωγής νέου προτύπου στο δίκτυο, μπορεί να γίνει απευθείας χωρίς τον κίνδυνο να χαθεί κανένα μέρος της αρχικής πληροφορίας. Το πλεονέκτημα των δικτύων, που χρησιμοποιούν off line διαδικασίες εκπαίδευσης επικεντρώνεται κυρίως στη δυνατότητα να δίνουν καλύτερες λύσεις σε δύσκολα προβλήματα (https://el.wikipedia.org/wiki/%CE%9D%CE%B5%CF%85%CF%81%CF%89%CE%BD%CE%B9%CE%BA%CF%8C_%CE%B4%CE%AF%CE%BA%CF%84%CF%85%CE%BF).

2.5 Εφαρμογές των Νευρωνικών Δικτύων

Τα Νευρωνικά Δίκτυα εφαρμόζονται σε ένα ευρύ φάσμα τομέων της επιστήμης και της τεχνολογίας μεταξύ των οποίων:

- Ρομποτική: Ρομπότ – μεταφορείς, που γνωρίζουν πού βρίσκονται και πού πηγαίνουν μπορούν να μετακινούν υλικά και να μεταφέρουν περισσότερο βάρος με μεγαλύτερη ασφάλεια σχεδόν σε κάθε εργοστάσιο. Αποθήκες χρησιμοποιούν συχνά ευφυή συστήματα για να αποθηκεύουν υλικά με τη μεγαλύτερη ασφάλεια και αποτελεσματικότητα και για να τα ξαναβρίσκουν σε χρόνο σημαντικά μικρότερο από όσο θα χρειαζόταν ένας άνθρωπος. Συγκροτήματα γραφείων κάποιες φορές χρησιμοποιούν ρομπότ για τη μεταφορά εγγράφων και αλληλογραφίας. Τα ρομπότ-μεταφορείς δεν χρησιμοποιούνται βέβαια μόνο εντός κτιρίων. Μπορούν να χρησιμοποιηθούν σε δρόμους και σιδηροδρόμους για τη μεταφορά εμπορευμάτων και ανθρώπων.

- Ιατρική : Κάποια νοσοκομεία χρησιμοποιούν ρομπότ για να μεταφέρουν φάρμακα και ιατρικά έγγραφα. Εκτός αυτών, ρομπότ – χειρουργοί έχουν κάνει εγχειρήσεις σε εγκεφάλους, καρδιές και μηριαίες αρθρώσεις. Οι Επιστήμονες στο Imperial College του Λονδίνου έχουν δημιουργήσει ένα ρομπότ που μπορεί να κάνει λήψη αίματος πιο αξιόπιστα απ' ότι ο άνθρωπος (Graham, 2004: 9 επ.).

- Χρηματοοικονομικά

- Φυσική

- Γεωλογία

- Βιολογία

- Γλωσσολογία

- Εκπαίδευση

- Υπολογιστές

2.6 Επίλογος

Όπως έγινε αντιληπτό, η Τεχνητή Νοημοσύνη συνιστά, επί της ουσίας, ένα πολυεπιστημονικό πεδίο, το οποίο αντλεί στοιχεία από πολλούς και διαφορετικούς τομείς (Φιλοσοφία, Μαθηματικά, Γνωστική Ψυχολογία, Μηχανική, κτλ.), και οι εφαρμογές του αφορούν πολλούς και διαφορετικούς τομείς (Ιατρική, Εκπαίδευση, Γλωσσολογία, Γεωλογία, Βιολογία, Αστρονομία, κτλ.) (Κεραυνού 2000:31). Στις μέρες μας ειδικά, τα συστήματα Τεχνητής Νοημοσύνης, τα βρίσκουμε παντού γύρω μας και ας μην το φανταζόμαστε ή το υποψιαζόμαστε: οδηγούν αυτοκίνητα, αποφασίζουν σχετικά με τις αιτήσεις υποθηκών ή δανείων, βοηθούν στη μετάφραση κειμένων, στην αναγνώριση προσώπων, στα κοινωνικά δίκτυα, στον εντοπισμό της θέσης, δημιουργούν έργα τέχνης, παίζουν παιχνίδια κλπ. (http://physics4u.gr/blog/2018/01/18/%CF%84%CE%B9-%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9-%CE%B7-%CF%84%CE%B5%CF%87%CE%BD%CE%B7%CF%84%CE%AE-%CE%BD%CE%BF%CE%B7%CE%BC%CE%BF%CF%83%CF%8D%CE%BD%CE%B7-%CE%BA%CE%B1%CE%B9-%CF%80%CF%89%CF%82-%CE%B8%CE%B1/).

Ειδικά, όσον αφορά στα Τεχνητά Νευρωνικά Δίκτυα, διαδραματίζουν επιπρόσθετα εξαιρετικά σημαντικά ρόλο στην πρόληψη και στην αντιμετώπιση απειλών και κινδύνων των πληροφοριακών συστημάτων Εταιρειών ή/και Οργανισμών, τα οποία συνιστούν θεμελιώδες τμήμα της εύρυθμης λειτουργίας τους. Αντιλαμβάνεται λοιπόν κανείς ότι η διασφάλιση της ομαλής και ασφαλούς λειτουργίας τους καθίσταται επιτακτική ανάγκη.

Κεφάλαιο 3

Απειλές στην ασφάλεια πληροφοριακών συστημάτων

3.1 Εισαγωγή

Η σημερινή εποχή χαρακτηρίζεται από την ιδιαίτερα μεγάλη ανάπτυξη και την γενικευμένη χρήση της τεχνολογίας της πληροφορικής και των επικοινωνιών. Οι υπολογιστές χρησιμοποιούνται σήμερα σε όλες σχεδόν τις ανθρώπινες δραστηριότητες και σε κάθε είδους εργασίες, εμπορικές, επιστημονικές κλπ. Παράλληλα όμως αυξάνονται οι κίνδυνοι και τα κρούσματα από ηθελημένες ή τυχαίες καταστροφές, αλλοιώσεις ή μη εξουσιοδοτημένη χρήση των δεδομένων και γενικότερα των υπολογιστικών πόρων. Οι συνέπειες από πιθανές καταστροφές, αλλοιώσεις ή κακή χρήση των δεδομένων μπορούν να σημαίνουν όχι μόνο σημαντικές ζημίες και κόστη, αλλά και κινδύνους για την προστασία των ατομικών δικαιωμάτων των πολιτών (Πάγκαλος, Μαυρίδης 2002:16).

3.2 Ασφάλεια τεχνολογίας, πληροφορίας και επικοινωνιών

Η έννοια της ασφάλειας ενός πληροφοριακού συστήματος σχετίζεται με την ικανότητα ενός Οργανισμού να προστατεύει τις πληροφορίες του από τυχόν αλλοιώσεις και καταστροφές καθώς και από μη εξουσιοδοτημένη χρήση των πόρων του. Σχετίζεται επίσης με την ικανότητά του να παρέχει ορθές και αξιόπιστες πληροφορίες, οι οποίες είναι διαθέσιμες στους εξουσιοδοτημένους χρήστες κάθε φορά που τις αναζητούν. Η ικανότητα αυτή στηρίζεται στη λήψη μέτρων, τα οποία διασφαλίζουν την ακεραιότητα και την εμπιστευτικότητα των δεδομένων, καθώς και την αδιάλειπτη λειτουργία του υπολογιστικού συστήματος ((Πάγκαλος, Μαυρίδης 2002:17).

Κατά τη διάρκεια των αιώνων, δημιουργήθηκε ένα περίπλοκο σύνολο πρωτοκόλλων και μηχανισμών για την αντιμετώπιση ζητημάτων ασφάλειας της πληροφορίας, όταν οι πληροφορίες μεταφέρονται με φυσικά έγγραφα. Για παράδειγμα, το απόρρητο της αλληλογραφίας διασφαλίζεται από σφραγισμένους φακέλους που παραδίδονται από μια πιστοποιημένη Υπηρεσία Αλληλογραφίας. Ο τρόπος καταγραφής των πληροφοριών δεν έχει μεταβληθεί δραματικά από τότε. Ενώ οι πληροφορίες τυπικά αποθηκεύονταν και μεταδίδονταν σε χαρτί, ένα μεγάλο μέρος τους βρίσκεται τώρα σε μαγνητικά μέσα και μεταδίδεται μέσω τηλεπικοινωνιακών συστημάτων, μερικά εκ των οποίων είναι ασύρματα. Ωστόσο, αυτό που έχει

αλλάξει δραματικά είναι η ικανότητα αντιγραφής και εναλλαγής πληροφοριών. Κάποιο πρόσωπο μπορεί να δημιουργήσει χιλιάδες πανομοιότυπα αντίγραφα μιας πληροφορίας, που είναι αποθηκευμένη ηλεκτρονικά, που το καθένα να μην μπορεί να διακριθεί από το πρωτότυπο. Αυτό, που χρειάζεται λοιπόν η κοινωνία, όπου οι πληροφορίες κατά το πλείστον αποθηκεύονται και μεταδίδονται κυρίως σε ηλεκτρονική μορφή, είναι ένα μέσο για την διασφάλιση της ασφάλειας των πληροφοριών, η οποία είναι ανεξάρτητη από το φυσικό μέσο που την καταγράφει ή τη μεταδίδει ώστε οι στόχοι της ασφάλειας των πληροφοριών να βασίζονται αποκλειστικά στην ψηφιακή πληροφορία και μόνο (Menezes, Oorschot, Vanstone 1997: 2 επ.). Κατά συνέπεια, οι στόχοι της ασφάλειας των πληροφοριών δεν μπορούν να επιτευχθούν μόνο μέσω μαθηματικών αλγορίθμων και πρωτοκόλλων, αλλά απαιτούν διαδικαστικές τεχνικές και τήρηση νόμων για την επίτευξη του επιθυμητού αποτελέσματος.

Η Ασφάλεια Τεχνολογίας Πληροφορίας και Επικοινωνιών (ICT Security) περιλαμβάνει την ασφάλεια:

1. Των υπολογιστικών συστημάτων και εφαρμογών, δηλαδή την προστασία από μη εξουσιοδοτημένες ενέργειες, όπως αλλαγή δικαιωμάτων πρόσβασης, κακόβουλη εκτέλεση εντολών, τροποποίηση της διάρθρωσης του συστήματος, κακόβουλη ή λανθασμένη χρήση, διακοπή λειτουργίας, καθώς και τη φυσική προστασία των υπολογιστικών συστημάτων.

2. Των δικτύων και των υποδομών, δηλαδή την προστασία από μη εξουσιοδοτημένη λογική πρόσβαση σε ένα δίκτυο, παράκαμψη ή τροποποίηση των κανόνων δρομολόγησης στο δίκτυο, παρακολούθηση του μέσου επικοινωνίας, διακοπή της επικοινωνίας, φυσική προστασία των υποδομών επικοινωνίας κτλ. και

3. Των πληροφοριών, δηλαδή την προστασία των δεδομένων ως προς την εμπιστευτικότητα, την ακεραιότητα και τη διαθεσιμότητά τους (Μάγκος 2007: 4 επ.).

3.3 Απειλές στην ασφάλεια

Ως απειλή (threat) νοείται μια δυνητική αιτία πρόκλησης περιστατικού παραβίασης της ασφάλειας, η οποία μπορεί να προκαλέσει ζημία στο σύστημα ή στον Οργανισμό. Οι απειλές διακρίνονται στις κάτωθι κατηγορίες:

- φυσικές απειλές (π.χ φωτιά, πλημμύρα, σεισμός),

- απειλές τεχνητής φύσης (π.χ διακοπή ηλεκτροδότησης, αστοχία λογισμικού συστήματος και δικτύου, αστοχία λογισμικού εφαρμογών, βλάβη εξυπηρετητή, βλάβη συσκευής δικτύου) και

- ανθρώπινες απειλές, οι οποίες διακρίνονται επιμέρους σε σκόπιμες (π.χ πλαστοπροσωπία από εσωτερικούς/εξωτερικούς χρήστες ή παρόχους υπηρεσιών, μη εξουσιοδοτημένη χρήση εφαρμογής, μη εξουσιοδοτημένη τροποποίηση δεδομένων, φιλτράρισμα επικοινωνιών ή παρεμβολές στις επικοινωνίες, κλοπή υλικού, ηθελημένη πρόκληση βλάβης – βανδαλισμός) και σε τυχαίες (π.χ εσφαλμένη διαγραφή αρχείων, λανθασμένη δρομολόγηση, σφάλμα συντήρησης υλικού ή λογισμικού, εισαγωγή κακόβουλου λογισμικού) απειλές (Κάτσικας 2014: 44).

Ο Stallings (Stallings 2007:703 επ.) διακρίνει τις επιθέσεις στην ασφάλεια σε δύο (2) μεγάλες κατηγορίες:

- τις παθητικές απειλές, οι οποίες προσπαθούν να αποκτήσουν ή να χρησιμοποιήσουν πληροφορίες από το σύστημα χωρίς να επηρεάζουν τους πόρους του συστήματος και

- τις ενεργητικές απειλές, οι οποίες προσπαθούν να τροποποιήσουν τους πόρους του συστήματος ή να επηρεάσουν τη λειτουργία τους.

Ειδικότερα, οι παθητικές επιθέσεις ανήκουν στην κατηγορία της υποκλοπής ή της παρακολούθησης της μετάδοσης των δεδομένων. Ο στόχος του επιτιθέμενου είναι η απόκτηση των μεταδιδόμενων πληροφοριών. Δύο (2) τύποι παθητικών επιθέσεων είναι η απόκτηση του περιεχομένου των μηνυμάτων και η ανάλυση του ρυθμού ανταλλαγής τους στο δίκτυο. Οι παθητικές επιθέσεις είναι πολύ δύσκολο να ανιχνευθούν επειδή δεν περιλαμβάνουν καμία αλλοίωση των δεδομένων. Τυπικά, η ανταλλαγή των δεδομένων πραγματοποιείται με προφανή

τρόπο και ούτε ο αποστολέας, ούτε ο παραλήπτης γνωρίζει ότι ένα τρίτο πρόσωπο έχει διαβάσει τα μηνύματα ή έχει παρατηρήσει την ανταλλαγή τους. Ωστόσο, είναι εφικτό να αποφευχθεί η πραγματοποίηση αυτών των επιθέσεων μέσω της κρυπτογράφησης. Για το λόγο αυτό, όσον αφορά στην συγκεκριμένη κατηγορία απειλών, το σημαντικότερο είναι η πρόληψή τους παρά η ανίχνευσή τους.

Από την άλλη, οι ενεργητικές επιθέσεις περιλαμβάνουν τροποποίηση της ροής δεδομένων ή δημιουργία ψευδούς ροής ανταλλαγής δεδομένων και μπορούν να υποδιαιρεθούν σε τέσσερις (4) κατηγορίες: «μεταμφίεση», «επανάληψη», «τροποποίηση των μηνυμάτων» και «άρνηση υπηρεσιών». Οι ενεργητικές επιθέσεις παρουσιάζουν τα αντίθετα χαρακτηριστικά από αυτά των παθητικών επιθέσεων. Οι ενεργητικές επιθέσεις είναι αρκετά δύσκολο να αποτραπούν εντελώς, διότι αυτό θα απαιτούσε και φυσική προστασία όλων των εγκαταστάσεων επικοινωνίας. Αντ' αυτού, στόχος εν προκειμένω είναι η ανίχνευσή τους και η αποκατάσταση της ζημίας, που προκλήθηκε από αυτές.

3.3.1 Κακόβουλο λογισμικό (malicious software)

Με τον όρο «κακόβουλο λογισμικό» νοείται το λογισμικό εκείνο, το οποίο σκόπιμα συμπεριλαμβάνεται ή εισάγεται σε ένα σύστημα και έχει στόχο να καταστρέψει ή να αλλοιώσει τα δεδομένα ή να απενεργοποιήσει ένα δίκτυο ή ένα υπολογιστικό σύστημα (Simpson, Backman, Corley 2013:77). Πιθανότατα, οι πιο εξελιγμένοι τύποι απειλών των υπολογιστικών συστημάτων προέρχονται ίσως από προγράμματα που εκμεταλλεύονται τις ευπάθειές τους. Οι απειλές αυτές αναφέρονται ως «κακόβουλο λογισμικό» (Stallings 2011:785).

Το κακόβουλο λογισμικό διακρίνεται σε κατηγορίες ανάλογα με τον τρόπο αναπαραγωγής του και την αυτονομία του από άλλα προγράμματα - ξενιστές και σε επιμέρους είδη ανάλογα με τον τρόπο δράσης του. Σύμφωνα με τον Stallings, η κατηγοριοποίησή του χρησιμοποιεί ως κριτήρια την αυτονομία και την αναπαραγωγή του λογισμικού. Ειδικότερα, με την έννοια «αυτονομία» χαρακτηρίζεται η δυνατότητα του κακόβουλου λογισμικού να λειτουργήσει χωρίς να χρειάζεται να προσκολληθεί σε ένα λογισμικό – ξενιστή (host), ενώ με την έννοια «αναπαραγωγή» νοείται η δυνατότητά του να αναπαράγεται από μόνο του, όταν οι συνθήκες το επιτρέπουν. Χρησιμοποιώντας αυτές τις δύο (2) ιδιότητες ως κριτήρια κατηγοριοποίησης του κακόβουλου λογισμικού προκύπτει η κατηγοριοποίησή του σε αυτό, που χρειάζεται ξενιστή, όπως για

παράδειγμα οι κερκόπορτες (backdoors), οι λογικές βόμβες (logic bombs), οι δούρειοι ίπποι (trojan horses) – που σημειωτέον δεν δημιουργεί αντίγραφα- και οι ιοί (viruses) και σε αυτό που δεν χρειάζεται ξενιστή, όπως για παράδειγμα τα βακτήρια και οι αναπαραγωγοί. Περαιτέρω, με βάση τον τρόπο δράσης του το κακόβουλο λογισμικό (malware) διακρίνεται σε ιομορφικό λογισμικό και σε μη ιομορφικό λογισμικό (Ηλιάδης 2004: 235 επ.).

3.3.2 Κοινωνική μηχανική (social engineering)

Οι ορισμοί για την έννοια της Κοινωνικής Μηχανικής, είναι πολλοί και ποικίλλουν. Ενδεικτικά αναφέρονται οι εξής: Στην Wikipedia (2010) η κοινωνική μηχανική ορίζεται ως η πράξη της χειραγώγησης ατόμων με σκοπό την απόσπαση πληροφοριών. Σύμφωνα με τον ανωτέρω ορισμό αν και είναι παρόμοια με το «τέχνασμα» ή την απλή «απάτη», ο όρος είναι κυρίως συνδεδεμένος με την εξαπάτηση ατόμων με σκοπό τη συλλογή πληροφοριών, την απάτη ή την πρόσβαση σε ένα υπολογιστικό σύστημα. Στις περισσότερες περιπτώσεις αυτός, που την εφαρμόζει δεν έρχεται ποτέ πρόσωπο με πρόσωπο με το άτομο που εξαπατά. Επ' αυτού του ζητήματος, ο Davis επισημαίνει ότι αν και η κοινωνική μηχανική έχει οριστεί με πολλούς τρόπους, ο καλύτερος ορισμός είναι ο εξής: *«ένας εχθρός, που χειρίζεται ή χρησιμοποιεί ψυχολογικά τεχνάσματα για να κερδίσει την εμπιστοσύνη ενός εξουσιοδοτημένου εργαζόμενου σε ένα δίκτυο, εκμεταλλευόμενος τη φυσική τάση του ανθρώπου να εμπιστεύεται και να βοηθάει τους άλλους»* (Trim, Upton 2013:45 επ.). Περαιτέρω, ο Harley ορίζει την κοινωνική μηχανική ως *«ψυχολογική χειραγώγηση, εξειδικευμένη ή μη, ενός προσώπου ή ενός συνόλου προσώπων για την επίτευξη επιθυμητής επίδρασης στη συμπεριφορά τους»*. Τέλος, σύμφωνα με τον John Palumbo, η κοινωνική μηχανική είναι η εκ μέρους ενός επιτιθέμενου χρήση ψυχολογικών μεθόδων σε νόμιμους χρήστες ενός πληροφοριακού συστήματος προκειμένου να αποκτήσει εμπιστευτικές πληροφορίες (ονόματα χρηστών και κωδικούς πρόσβασης) και κατ' επέκταση πρόσβαση στο σύστημα. Πολλοί μάλιστα, έχουν χαρακτηρίσει την κοινωνική μηχανική ως μια διαδικασία εκμετάλλευσης δύο (2) ανθρώπινων αδυναμιών κατά την πραγματοποίηση μιας επίθεσης: Η πρώτη είναι ότι κανένας δεν επιθυμεί να θεωρηθεί αμαθής και η δεύτερη είναι η εκμετάλλευση της ανθρώπινης εμπιστοσύνης. Αυτές είναι δύο (2) αδυναμίες που καθιστούν την κοινωνική μηχανική ανθεκτική στην καταπολέμησή της, επειδή κανείς δεν θέλει να παραδεχτεί ότι ηττήθηκε από αυτήν. Για το λόγο αυτό η κοινωνική μηχανική αποτελεί απειλή στην ασφάλεια

του συστήματος (Kizza 2009:100). Οι επιτιθέμενοι χρησιμοποιούν πολλές μεθόδους υλοποίησης, συμπεριλαμβανομένων των ακόλουθων:

- τηλέφωνο: Η συγκεκριμένη μέθοδος θεωρείται η πιο κλασική μέθοδος και συνίσταται στο ότι οι κακόβουλοι χρήστες καλούν τηλεφωνικά ένα πρόσωπο, το οποίο έχουν στοχοποιήσει για να εκμεταλλευτούν και συνομιλούν μαζί του προκειμένου να εκμαιεύσουν σταδιακά πληροφορίες από αυτό, προσποιούμενοι έναν νόμιμο χρήστη του πληροφοριακού συστήματος ή ότι είναι μέλη του Οργανισμού.

- απευθείας σύνδεση (online): Με την εν λόγω μέθοδο οι επιτιθέμενοι συλλέγουν κρίσιμες πληροφορίες από απρόσεκτους χρήστες, που βρίσκονται συνδεδεμένοι στο διαδίκτυο. Είναι γνωστό ότι η εξάρτηση και η υπερβολική χρήση του διαδικτύου έχει ως αποτέλεσμα οι χρήστες να δημιουργούν αρκετούς λογαριασμούς στο διαδίκτυο. Στις μέρες μας, ένας μέσος χρήστης έχει κατά μέσο όρο περίπου τέσσερις (4) με πέντε (5) λογαριασμούς, εκ των οποίων ένας για οικιακή χρήση, ένας για επαγγελματική και επιπλέον ένας (1) ή δύο (2) για κοινωνικά δίκτυα ή επαγγελματικούς οργανισμούς. Ωστόσο, με την χρησιμοποίηση της πρακτικής αυτής, ο χρήστης είναι βέβαιο ότι θα ξεχάσει ορισμένους κωδικούς πρόσβασης, ειδικά αυτούς, τους λογαριασμούς των οποίων χρησιμοποιεί λιγότερο συχνά. Από την άλλη, πολλοί χρήστες χρησιμοποιούν λανθασμένα, τον ίδιο κωδικό πρόσβασης σε πολλούς λογαριασμούς τους ώστε να μην αντιμετωπίζουν το ανωτέρω πρόβλημα. Οι επιτιθέμενοι το γνωρίζουν αυτό και στοχεύουν στους χρήστες αυτούς, τους οποίους προσεγγίζουν με έξυπνες μεθόδους - δολώματα, όπως για παράδειγμα λέγοντάς τους ότι κέρδισαν σε λαχειοφόρες αγορές ή ότι επελέγησαν σε κληρώσεις ή ότι έχουν κερδίσει συγκεκριμένο αριθμό βραβείων σε διαγωνισμούς, καλώντας τους ταυτόχρονα- προκειμένου να λάβουν το βραβείο τους - να συμπληρώσουν μια ηλεκτρονική φόρμα, η οποία συνήθως βρίσκεται στο διαδίκτυο. Με την συμπλήρωσή της ο επιτιθέμενος αποκτά τον κωδικό πρόσβασης του χρήστη.

- dumpster diving: Η μέθοδος αυτή συνιστά μια τεχνική συλλογής πληροφοριών, γνωστή και ως «trashing» και περιλαμβάνει την κλοπή πληροφοριών από προσωπικούς και εταιρικούς κάδους απορριμμάτων. Με τη χρησιμοποίηση αυτής της μεθόδου ένας επιτιθέμενος μπορεί να ανακτήσει από κάδους και δοχεία απορριμμάτων ατομικούς αριθμούς κοινωνικής ασφάλισης, τραπεζικούς λογαριασμούς, προσωπικά αρχεία με ευαίσθητα προσωπικά δεδομένα καθώς και έναν πλήρη κατάλογο προσωπικών και

επαγγελματικών πληροφοριών, οι οποίες παρέχουν στους επιτιθέμενους τα ακριβή στοιχεία, που χρειάζονται για να εκμεταλλευτούν το δίκτυο του Οργανισμού.

- αυτοπροσώπως: Η μέθοδος αυτή είναι η παλαιότερη από τις τεχνικές κλοπής πληροφοριών, η οποία προηγείται των υπολογιστών. Σύμφωνα με αυτήν, ο επιτιθέμενος είναι ένα πρόσωπο που έχει φυσική παρουσία σε έναν Οργανισμό και περιστασιακά ελέγχει πίνακες ανακοινώσεων, ψάχνει απορρίμματα από κάδους, που βρίσκονται σε τουαλέτες ή σε διαδρόμους της Εταιρείας και γευματίζει και συνομιλεί με τους υπαλλήλους αυτής. Σε μεγάλες Εταιρείες, αυτό μπορεί να επιτευχθεί μόνο σε ορισμένες περιπτώσεις πριν αναπτυχθούν σχέσεις εμπιστοσύνης. Από τέτοιες φιλικές σχέσεις, μπορεί ένα πρόσωπο να αποκαλύψει εμπιστευτικές πληροφορίες χωρίς καν να το συνειδητοποιήσει.

- κακόβουλη αλληλογραφία (snail mail): Η μέθοδος αυτή υλοποιείται με διάφορους τρόπους και δεν περιορίζεται μόνο στην κοινωνική μηχανική αλλά έχει χρησιμοποιηθεί και για την διάπραξη και άλλων αδικημάτων. Η μέθοδος αυτή λαμβάνει χώρα με δύο (2) τρόπους: ο επιτιθέμενος επιλέγει το θύμα του και αντικαθιστά την διεύθυνση αλληλογραφίας του με μία νέα, που είναι η δική του. Με αυτό τον τρόπο παρέχεται στον επιτιθέμενο η δυνατότητα να παρακολουθεί την αλληλογραφία του χρήστη. Από την παρακολουθούμενη αλληλογραφία ο επιτιθέμενος μπορεί να συγκεντρώσει πολλές πληροφορίες, οι οποίες ενδεχομένως περιλαμβάνουν αριθμούς τραπεζικών λογαριασμών και πιστωτικών καρτών του θύματος και κωδικούς ελέγχου πρόσβασης. Άλλο παράδειγμα, που χρησιμοποιείται για την μέθοδο αυτή είναι η χρήση ερωτηματολογίου, το οποίο ο επιτιθέμενος ρίχνει σκόπιμα στην αλληλογραφία του θύματος, δελεάζοντάς τον με την προσφορά χρημάτων προκειμένου να συμπληρώσει «κάποιες απλές» ερωτήσεις και να του τις αποστείλει. Ωστόσο, στην πραγματικότητα, οι ερωτήσεις, ζητούν από το ανυποψίαστο θύμα πολλά περισσότερα στοιχεία και δεν συνιστούν απλές πληροφορίες.

- Η πλαστοπροσωπία είναι επίσης ένα παλιό τέχνασμα, που έχει χρησιμοποιηθεί σε ανυποψίαστα θύματα από εγκληματίες προκειμένου να αποκτήσουν μια σειρά αγαθών. Στις μέρες μας τα αγαθά είναι οι πληροφορίες. Η μέθοδος αυτή συνίσταται στο ότι ο επιτιθέμενος προσποιείται ότι είναι το θύμα με αποτέλεσμα να αποκτήσει επαφές με τα κατάλληλα πρόσωπα ώστε να αποσπάσει τις πληροφορίες, που χρειάζεται. Σε μεγάλους Οργανισμούς μάλιστα οι οποίοι έχουν εκατοντάδες ή χιλιάδες εργαζομένους

διάσπαρτους σε όλο τον πλανήτη, είναι πολύ εύκολο κάποιος να παραστήσει κάποιο υψηλά ιστάμενο πρόσωπο, όπως για παράδειγμα έναν Αντιπρόεδρο ή έναν Υπεύθυνο μιας Επιχείρησης. Με δεδομένο μάλιστα ότι οι περισσότεροι υπάλληλοι θέλουν να δείχνουν καλοί στους Προϊσταμένους τους, μπορεί να καταλήξουν ασυνείδητα στην παροχή εμπιστευτικών πληροφοριών σε ακατάλληλα πρόσωπα. (Kizza, 2009: 100 επ.).

3.3.3 Εξωτερικοί εισβολείς (outsiders)

Ο όρος «hacker» έχει αλλάξει νόημα όλα αυτά τα χρόνια καθώς η τεχνολογία εξελίσσεται. Στις μέρες μας ο όρος έχει δύο (2) αντίθετες σημασίες: Ο ένας ορισμός αναφέρεται σε ένα πρόσωπο το οποίο απολαμβάνει να ανακαλύπτει σε βάθος τον τομέα των υπολογιστών καθώς και τον τρόπο που θα εκμεταλλευτεί τις δυνατότητές τους σε αντίθεση με τους περισσότερους χρήστες που περιορίζονται μόνο στην εκμάθηση των βασικών τους λειτουργιών. Ο αντίθετος ορισμός αναφέρεται σε έναν χρήστη, ο οποίος προσπαθεί να αποκτήσει χρήσιμες πληροφορίες από τους άλλους με κακόβουλες ενέργειες. Πριν από την απόκτηση της τρέχουσας υποτιμητικής σημασίας, ο όρος hacking χρησιμοποιούνταν για να περιγράψει ένα πρόσωπο το οποίο ειδικεύονταν στην εγγραφή και τροποποίηση προγραμμάτων ηλεκτρονικών υπολογιστών. Οι hacker θεωρούνταν άνθρωποι, που γνώριζαν πολύ καλά την πληροφορική. Θεωρούνταν εμπειρογνώμονες υπολογιστών που θα μπορούσαν να κάνουν τον υπολογιστή να πραγματοποιήσει κάθε επιθυμητή εντολή μέσω του προγραμματισμού. Σήμερα, ωστόσο, ο όρος αναφέρεται στην διαδικασία απόκτησης μη εξουσιοδοτημένης πρόσβασης σε ένα υπολογιστικό σύστημα για διάφορους σκοπούς, συμπεριλαμβανομένης της κλοπής και τροποποίησης δεδομένων. Υπάρχουν διάφορες υποκατηγορίες hackers, που βασίζονται στην διαφορετική τους δράση. Οι κατηγορίες αυτές είναι οι ακόλουθες:

- cracker: είναι το πρόσωπο που παραβιάζει την ασφάλεια ενός συστήματος. Τα άτομα αυτά ανήκουν στην κατηγορία των πιο σκληροπυρηνικών hackers, που χαρακτηρίζονται περισσότερο ως επαγγελματίες που παραβιάζουν την ασφάλεια με στόχο την κλοπή. Ο όρος δημιουργήθηκε στα μέσα της δεκαετίας του '80 από μία κατηγορία hackers, η οποία ήθελε να διαφοροποιηθεί από τα άτομα με εγκληματικά κίνητρα, τα οποία είχαν μοναδικό σκοπό την παραβίαση των συστημάτων ασφαλείας. Η κατηγορία αυτή υποστήριζε ότι οι δημοσιογράφοι παραποιούσαν τον όρο «hacker». Ανησυχούσαν ότι τα μέσα μαζικής ενημέρωσης δεν μπόρεσαν να διακρίνουν μεταξύ αυτών που είχαν και

αυτών που δεν είχαν εγκληματικά κίνητρα. Η διάκριση αυτή ωστόσο απέτυχε. Έτσι, οι δύο (2) όροι hack και crack χρησιμοποιούνται συχνά χωρίς διαφοροποίηση. Παρά το γεγονός ότι ο κόσμος εξακολουθεί να μην βλέπει τη διαφορά μεταξύ των hackers και crackers, οι πρώτοι εξ αυτών εξακολουθούν να υποστηρίζουν ότι υπάρχει μεγάλη διαφορά ανάμεσα σε αυτά που κάνουν οι ίδιοι και στις ενέργειες της δεύτερης κατηγορίας.

- hacktivism: Ο όρος αυτός είναι ένας συνδυασμός μεταξύ του όρου hacking και του όρου «ακτιβισμός» (activism). Τα πρόσωπα που ανήκουν σε αυτή την κατηγορία είναι συνειδητοποιημένοι hackers που υποστηρίζουν ότι οι ενέργειές τους, τις οποίες πραγματοποιούν με τη βοήθεια των υπολογιστικών συστημάτων, έχουν κάποια αιτία και ελπίζουν ότι με τον τρόπο αυτό θα αναδείξουν τις πράξεις που θεωρούν θεσμικά και πολιτικά ανήθικες. Σε αυτήν την κατηγορία επίσης υπάγονται και πράξεις πολιτικής ανυπακοής που χρησιμοποιούν το διαδίκτυο. Οι μέθοδοι, που χρησιμοποιούνται μεταξύ των οποίων το αυτοματοποιημένο μαζικής αποστολής μήνυμα (automated e-mail bomb), παραποίηση ιστοσελίδας (web defacing), αλλάζουν με την πάροδο του χρόνου και την εξέλιξη της τεχνολογίας (Kizza 2009:113 επ.).

3.3.4 Εσωτερικές απειλές (inside threats)

Ως «εσωτερική απειλή» νοείται το πρόσωπο, που πιθανώς έχει προνομιακή πρόσβαση σε ταξινομημένα, ευαίσθητα ή περιουσιακά δεδομένα και χρησιμοποιεί αυτό το πλεονέκτημα για να αφαιρέσει πληροφορίες από έναν Οργανισμό και να τις μεταφέρει σε μη εξουσιοδοτημένους εξωτερικούς χρήστες. Στην έννοια αυτή περιλαμβάνονται, αφ' ενός μεν οι χρήστες της επιχείρησης που παρακάμπτουν τις διαδικασίες ελέγχου για την πρόσβαση σε διαβαθμισμένα δεδομένα ή/και πληροφορίες, αφ' ετέρου δε οι χρήστες, που αποκτούν πρόσβαση σε λογαριασμούς χρηστών με περισσότερα δικαιώματα σε σχέση με τα δικαιώματα, που ήδη έχουν. Στοιχεία από έρευνες από πολλούς αξιόπιστους Οργανισμούς δείχνουν σταθερά ότι η μεγαλύτερη απειλή για την ασφάλεια σε κάθε επιχείρηση είναι οι ίδιοι οι υπάλληλοί της. Το 1997, η Εταιρεία «Ernst & Young» πήρε συνέντευξη από τέσσερις χιλιάδες διακόσιους είκοσι έξι (4.226) Διαχειριστές Πληροφοριακών Συστημάτων και επαγγελματίες από όλο τον κόσμο για την ασφάλεια των δικτύων τους. Από τις απαντήσεις που έλαβε, το 75% των Διαχειριστών πίστευε

ότι οι εξουσιοδοτημένοι χρήστες και οι εργαζόμενοι συνιστούν απειλή για την ασφάλεια των συστημάτων τους. Το 42% των ερωτηθέντων της Εταιρείας ανέφεραν ότι είχαν βιώσει εξωτερικές κακόβουλες επιθέσεις κατά το παρελθόν, ενώ το 43% ανέφερε κακόβουλες ενέργειες από τους ίδιους τους υπαλλήλους της Εταιρείας (Kizza 2009:79 επ.).

Είναι γνωστό ότι οι δυσαρεστημένοι εσωτερικοί χρήστες αποτελούν μια σημαντική πηγή εγκλημάτων, που σχετίζονται με τη χρήση ηλεκτρονικών υπολογιστών επειδή δεν απαιτούνται πολλές και ειδικευμένες γνώσεις αναφορικά με το πληροφοριακό σύστημα του Οργανισμού. Στις περισσότερες περιπτώσεις, οι εσωτερικοί χρήστες χρησιμοποιούν το Πληροφοριακό Σύστημα της Εταιρείας καθημερινά. Το γεγονός αυτό τους επιτρέπει να αποκτήσουν απεριόριστη πρόσβαση στα υπολογιστικά συστήματα πάνω στα οποία εργάζονται, προκαλώντας βλάβη τόσο σε αυτά, όσο και στα δεδομένα. Σύμφωνα με την έκθεση του Ινστιτούτου Ασφάλειας Υπολογιστών (CSI)/FBI, το έτος 1999 το 55% των ερωτηθέντων ανέφεραν κακόβουλη δραστηριότητα από εσωτερικούς χρήστες.

Σύμφωνα με τον Jack Strauss, Πρόεδρο και Διευθύνοντα Σύμβουλο της SafeCorp, μιας συμβουλευτικής Εταιρείας για την ασφάλεια των πληροφοριών στην εργασία, που έχει έδρα στο Dayton του Οχάιο, οι υπάλληλοι μιας Εταιρείας, είτε σκόπιμα, είτε άθελά τους αποτελούν την μεγαλύτερη απειλή για την ασφάλεια των πληροφοριών στις σύγχρονες επιχειρήσεις. Ο Strauss πιστεύει ότι συνιστά λάθος των υπευθύνων ασφαλείας να παραμελούν να κλειδώνουν την πίσω πόρτα στις κτιριακές της εγκαταστάσεις, να παραμελούν να κρυπτογραφούν ευαίσθητα δεδομένα στους φορητούς υπολογιστές τους ή να μην ανακαλούν προνόμια πρόσβασης όταν οι εργαζόμενοι αποχωρούν από την επιχείρηση (Kizza 2009:109).

3.4 Τρόποι αντιμετώπισης των απειλών

Οι περισσότεροι Οργανισμοί και επιχειρήσεις σήμερα εξαρτώνται από την πληροφοριακή τους υποδομή και τους πόρους των πληροφοριακών συστημάτων που διαθέτουν, όχι μόνο για να λειτουργήσουν, αλλά και για να αναπτυχθούν και να διευρύνουν τις δραστηριότητές τους. Η απρόσκοπτη λειτουργία των πληροφοριακών συστημάτων είναι επομένως βασική απαίτηση και επιδίωξη των Οργανισμών, που αξιοποιούν τις τεχνολογίες της πληροφορικής και των τηλεπικοινωνιών. Το ζήτημα της ασφάλειας των πληροφοριακών συστημάτων, τόσο λόγω της σημασίας του, όσο και γιατί αποτελεί σύνθετο πρόβλημα, απαιτεί μια συστηματική και

ολοκληρωμένη αντιμετώπιση. Η χρήση κατά περίπτωση τεχνολογικών μέτρων ασφάλειας, ακόμα και όταν αυτά είναι τα βέλτιστα διαθέσιμα, δεν επαρκεί, διότι κρίσιμα αγαθά του πληροφοριακού συστήματος, όπως είναι οι πληροφορίες, μπορεί να βρίσκονται διάσπαρτα μέσα στον οργανισμό. Το επίπεδο της ασφάλειας ενός πληροφοριακού συστήματος καθορίζεται από την ασφάλεια του ασθενέστερου σημείου του (weak – link phenomenon), κατά συνέπεια η εφαρμογή αποσπασματικών τεχνολογικών μέτρων ασφάλειας δεν είναι επαρκής, όταν τα μέτρα αυτά δεν εντάσσονται σε μια συνολική στρατηγική και δεν συνδυάζονται σε μια ενιαία και ολιστική αντιμετώπιση της επικινδυνότητας του πληροφοριακού συστήματος.

Η ολοκληρωμένη αντιμετώπιση του ζητήματος της ασφάλειας υλοποιείται με διαδικασίες, που εντάσσονται στη διαχείριση της ασφάλειας των πληροφοριακών συστημάτων εντός Οργανισμού, οι οποίες περιλαμβάνουν:

- τον προσδιορισμό των κινδύνων, που αντιμετωπίζουν τα πληροφοριακά συστήματα και των αναγκαίων μέτρων για την προστασία τους από τους κινδύνους αυτούς,

- τον καθορισμό μιας πολιτικής ασφαλείας για τα πληροφοριακά συστήματα και τον προσδιορισμό των διαθέσιμων πόρων για την εφαρμογή της πολιτικής ασφάλειας,

- τον καθορισμό των ρόλων, των αρμοδιοτήτων και την απόδοση υπευθυνοτήτων για τα ζητήματα της ασφάλειας των πληροφοριακών συστημάτων,

- την ενημέρωση και ευαισθητοποίηση των χρηστών σε ζητήματα ασφάλειας και την εκπαίδευση και κατάρτισή τους στη χρήση και εφαρμογή των μέτρων ασφαλείας,

- τον καθορισμό σχεδίων ανάνηψης και συνέχειας από την πραγματοποίηση περιστατικών ασφάλειας και

- την αξιολόγηση όλων των διαδικασιών διαχείρισης της ασφάλειας των πληροφοριακών συστημάτων.

Η πολιτική ασφάλειας αποτελεί μια από τις βασικότερες πρακτικές για την διαχείριση της ασφάλειας και η ανάπτυξη και εφαρμογή της θεωρείται απαραίτητη για τους οργανισμούς, που διαθέτουν πληροφοριακά συστήματα (Καρύδα 2004:378 επ.).

Η επίτευξη της ασφάλειας των πληροφοριών σε μια ηλεκτρονική κοινωνία απαιτεί μια ευρεία ποικιλία τεχνικών και νομικών δεξιοτήτων. Δεν υπάρχει, ωστόσο, εγγύηση ότι όλοι οι στόχοι ασφάλειας πληροφοριών που κρίνονται αναγκαίοι μπορούν να ικανοποιηθούν επαρκώς (Menezes, Oorschot, Vanstone 1997:4).

Κεφάλαιο 4

Πειραματικός Σχεδιασμός και αποτελέσματα πειράματος

4.1 Μέθοδος υλοποίησης

Η μέθοδος υλοποίησης, που χρησιμοποιήθηκε για την ανίχνευση τυχόν εσωτερικών απειλών στο διαθέσιμο dataset, συνίσταται στη δημιουργία εικόνων (οπτικοποίηση) από τα δεδομένα, για έναν συγκεκριμένο αριθμό χρηστών. Οι εικόνες, απεικονίζουν τη δραστηριότητα και τη συμπεριφορά του χρήστη, όπως αυτή προκύπτει από τις ενέργειες, που έχει πραγματοποιήσει στο Πληροφοριακό Σύστημα για το συγκεκριμένο χρονικό διάστημα, που έχει οριστεί. Οι εικόνες που προέκυψαν, χρησιμοποιήθηκαν για την εκπαίδευση του αλγορίθμου CNN μέσω του προγράμματος TensorFlow της Google ώστε με τη βοήθεια της Τεχνητής Νοημοσύνης να αναγνωρίζει ποιές από τις συμπεριφορές των χρηστών είναι κακόβουλες και επομένως και επικίνδυνες για το εκάστοτε Πληροφοριακό Σύστημα. Ακολουθήθηκαν τρία (3) στάδια προκειμένου να ολοκληρωθεί η διαδικασία και τελικά να εξαχθούν συμπεράσματα, τα οποία είναι τα ακόλουθα:

1. Εξαγωγή και κατανομή των δεδομένων του υπάρχοντος dataset ανά χρήστη, που εξετάστηκε και δημιουργία αρχείων, τα οποία βασίστηκαν στα δεδομένα του.

2.- Εισαγωγή των αρχείων των δεδομένων, που δημιουργήθηκαν, στη βιβλιοθήκη d3.js της Java, επιλογή κατάλληλου σχεδίου δημιουργίας εικόνων με την εξέταση των διαθέσιμων σχεδίων (patterns) της βιβλιοθήκης της εφαρμογής και δημιουργία των εικόνων (οπτικοποίηση) ανά χρήστη, οι οποίες περιελάμβαναν την δραστηριότητά του σε μηνιαία και εβδομαδιαία βάση.

- Δημιουργία των εικόνων.

3.-Εκπαίδευση του αλγορίθμου CNN με το πρόγραμμα Tensorflow της Google και δοκιμή αυτού.

- Εξαγωγή συμπερασμάτων.

4.2 Dataset

Το σύνολο δεδομένων (dataset), που χρησιμοποιήθηκε, λήφθηκε από το CERT, το οποίο αποτελεί τμήμα του Ινστιτούτου Τεχνολογίας Λογισμικού (SEI). Ο συγκεκριμένος Οργανισμός μελετά και επιλύει προβλήματα, που μπορεί να προκαλέσουν εκτεταμένες αρνητικές συνέπειες στον τομέα της ασφάλειας στον κυβερνοχώρο, ευπάθειες σε προϊόντα λογισμικού, καθώς και μακροπρόθεσμες αλλαγές στα δικτυωμένα συστήματα. Η δράση του περιλαμβάνει επίσης και την ανάπτυξη μεθόδων για τη βελτίωση της ασφάλειας στον κυβερνοχώρο. Σε συνεργασία με προμηθευτές λογισμικού, συμβάλλει στην επίλυση των τρωτών σημείων του λογισμικού. Αναπτύσσει εργαλεία, προϊόντα και μεθόδους για να βοηθήσει εταιρείες και Οργανισμούς να διεξάγουν εγκληματολογικές εξετάσεις και έρευνες, όπως επίσης και να αναλύουν ευπάθειες και να παρακολουθούν δίκτυα μεγάλης κλίμακας. Περαιτέρω, στο έργο του περιλαμβάνεται η συλλογή δεδομένων και η εξόρυξη (data mining), στατιστική και ανάλυση τάσεων, η ασφάλεια υπολογιστών και δικτύων, η διαχείριση περιστατικών ασφαλείας και εμπιστευτικών πληροφοριών, η διασφάλιση λογισμικού και πολλά άλλα. Τα αποτελέσματα της εν λόγω διατριβής ελπίζουμε να συμβάλουν στη βελτίωση της πρακτικής της ασφάλειας στον κυβερνοχώρο.

Το έργο του CERT είναι προσβάσιμο από όλους τους ενδιαφερόμενους, μέσω της ιστοσελίδας του (https://www.sei.cmu.edu/about/divisions/cert/index.cfm). Οι συνεργασίες του περιλαμβάνουν κυβερνητικούς Οργανισμούς υψηλού επιπέδου, συμπεριλαμβανομένου του Υπουργείου Εθνικής Άμυνας των ΗΠΑ και του Υπουργείου Εσωτερικής Ασφάλειας (DHS) καθώς και του Ομοσπονδιακού Γραφείου Ερευνών (FBI).

Για την εκπόνηση της παρούσας μεταπτυχιακής διατριβής, χρησιμοποιήθηκε συγκεκριμένο dataset, του οποίου έγινε λήψη (download) από το site του CERT. Το αρχείο περιελάμβανε αρχεία καταγραφής Πληροφοριακών Συστημάτων (log files), τύπου csv, που κατέγραφαν δραστηριότητα, η οποία κάλυπτε χρονική περίοδο δεκαοκτώ (18) μηνών, αρξαμένη την 01.01.2010 και λήξασα στις 31.05.2011. Μέσα από τα αρχεία αυτά και μετά από ανάλυση και επεξεργασία, επιχειρήθηκε να παρουσιαστεί μια εικόνα του Πληροφοριακού Συστήματος και να αναλυθούν συμπεριφορές χρηστών, οι οποίες χαρακτηρίζονται ως κακόβουλες. Η έκδοση, που ελήφθη είναι η 6.2r.

Το σύνολο των δεδομένων απαρτίζεται από επτά (7) μέρη – αρχεία και συγκεκριμένα:

• Ένα αρχείο με τις καταχωρήσεις σύνδεσης/αποτύπωσης κάθε χρήστη (logon.csv).

• Ένα αρχείο με τα αρχεία, που χρησιμοποιήθηκαν από τους χρήστες (decoy_file.csv).

• Ένα αρχείο με τα έγγραφα, που χρησιμοποιήθηκαν από τους χρήστες (file.csv).

• Ένα αρχείο με τα μηνύματα ηλεκτρονικού ταχυδρομείου, που απεστάλησαν και λήφθηκαν από τους χρήστες (email.csv).

• Ένα αρχείο με τα αιτήματα http, που απεστάλησαν από τους χρήστες (http.csv).

• Ένα αρχείο με τις συσκευές, που χρησιμοποιήθηκαν από τους χρήστες (device.csv).

• Ένα αρχείο με το όνομα εκάστου χρήστη και την θέση του στον Οργανισμό (psychometric.csv).

Επιπλέον, το CERT παρείχε ένα σύνολο αρχείων, στα οποία περιγράφονταν πέντε (5) περιπτώσεις, που αποτελούσαν σενάρια απειλών. Για κάθε σενάριο, παρέχονταν ένα (1) αρχείο απαντήσεων, που επέτρεπε τον εντοπισμό ύποπτων συμπεριφορών μεταξύ όλων των δραστηριοτήτων.

Οι απαντήσεις αυτές χρησιμοποιήθηκαν για να ελεγχθεί μετά την διεξαγωγή του πειράματος, εάν τα αποτελέσματα, που έχουν εξαχθεί με βάση την οπτικοποίηση επαληθεύονται.

Κάθε αρχείο περιείχε συγκεκριμένου τύπου περιεχόμενο, ανάλογο με την εκάστοτε ενέργεια, που εκτελέστηκε. Τα αρχεία καταγραφής (log files) περιλαμβάνουν τις κάτωθι πληροφορίες:

logon.csv:

id,date,user,pc,activity

Παράδειγμα:

{F3X8-Y2GT43DR-4906OHBL},01/02/2010 02:19:18,DNS1758,PC-0414,Logon

files.csv:

id,date,user,pc,filename,activity,to_removable_media,from_removable_media,content

Παράδειγμα:

{U4B5-M9PO56AC-1319NFVD},01/02/201007:27:13,SDH2394,PC5849,

R:\22B5gX4\R6SQWEDC.jpg,File Open,False,True,FF-D8

device.csv:

id,date,user,pc,file_tree,activity

Παράδειγμα:

{C7F1-G7LE60RU-2483DAXS},01/02/2010 07:22:42,JKS2444,PC-6961,

R:\JKS2444,Connect

email.csv:

id,date,user,pc,to,cc,bcc,from,activity,size,attachments,content

Παράδειγμα:

{S6L0-N6ZJ21ZX-9274DSEG},04/12/2011 11:03:27,SJF3798,PC-6691,

Aiko_Manning@msn.com,,,Farmer_Sylvester@earthlink.net,Send,38877,,"Allen...”

http.csv:

id,date,user,pc,url,activity,content

Παράδειγμα:

{E4X5-V5JL82ZL-0022EUOG},05/31/2011 16:56:52,SAM3862,PC-4718,

http://myspace.com/Electron/gev/Pnehpntr816120798.php,WWW Visit,"The …”

decoy.csv:

decoy_filename,pc

Παράδειγμα: C:\LJE2413\795JW126.jpg,PC-0302

psychometric.csv:

employee_name,user_id,O,C,E,A,N

Παράδειγμα: Nicholas Fletcher Pruitt,NFP2441,34,39,38,36,21

Δεδομένου ότι τα αρχεία δεν περιείχαν ενδείξεις ούτε για τον εντοπισμό του «κακόβουλου» χρήστη, ούτε για τον προσδιορισμό των απειλών, αρχικά ήταν απαραίτητο να λάβει χώρα μια προεργασία ώστε να καθοριστούν οι κανόνες και τα κριτήρια αναζήτησης των απειλών. Στην πραγματικότητα, κάθε Εταιρεία ή Οργανισμός, έχει διαφορετικούς κανόνες ανάλογα με τις δραστηριότητές της/του και την Πολιτική Ασφάλειας, που υιοθετεί και ακολουθεί. Ως αποτέλεσμα, είναι αδύνατο να θεσπιστούν καθολικοί κανόνες.

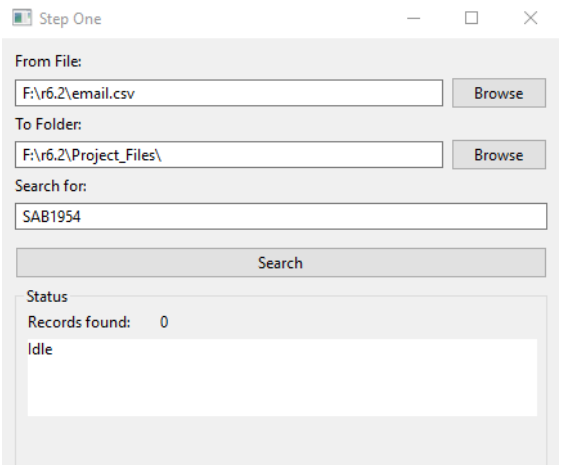
4.3 Πρώτο στάδιο πειράματος

Στο πρώτο μέρος του πειράματος δημιουργήθηκε μια εφαρμογή σε γλώσσα Java μέσω της οποίας τέθηκαν οι κανόνες αναζήτησης της δραστηριότητας των υπό εξέταση δεκαπέντε (15) χρηστών, οι οποίοι έφεραν τις εξής ονομασίες :

- ACM2278 (Salesman)
- CMP2946 (Salesman)
- PLJ1771 (I.T Admin)
- MBG3183 (Electrical Engineer)
- SAB1954 (Mechanical Engineer)
- ABK0481 (Software Quality Engineer)
- CCB3055 (Administrator)
- KCB1975 (Electrical Engineer)

- ACG0312 (Production Line worker)
- CJM0584 (Electrical Engineer)
- SKB2635 (Computer Scientist)
- CDE1846 (Electrical Engineer)
- LAN2608 (Electrical Engineer)
- POD0750 (Electrical Engineer)
- PIM3569 (Electrical Engineer)

Ειδικότερα, η εφαρμογή περιελάμβανε τέσσερα (4) βήματα, που συνδέονταν μεταξύ τους. Στο πρώτο βήμα (Step One), το πρόγραμμα ζητούσε την εισαγωγή του αρχείου στο οποίο θα γίνονταν η αναζήτηση του υπό εξέταση χρήστη (file.csv, email.csv, http.csv) το όνομα του χρήστη και ο φάκελος που θα έβαζε το αποτέλεσμα, όπως απεικονίζεται κατωτέρω.

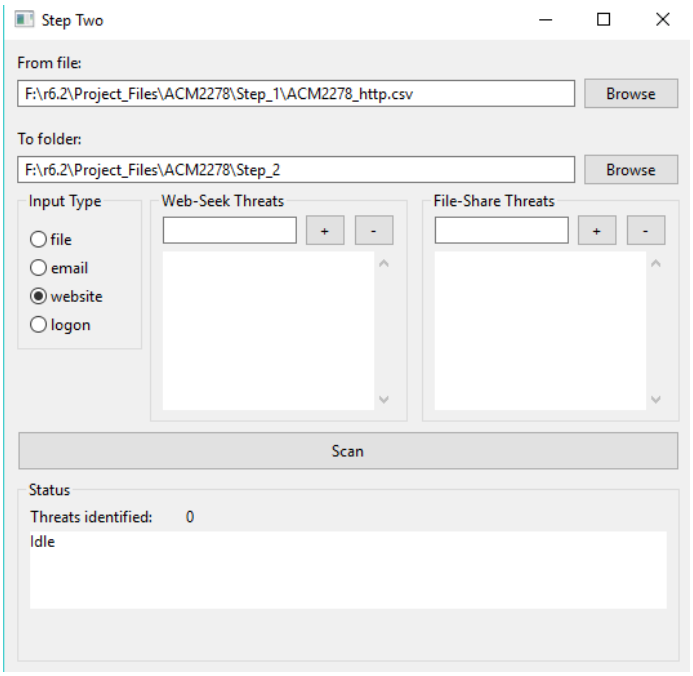


Εικόνα 4-1: User Interface πρώτου βήματος προγράμματος

Το πρόγραμμα έκανε αναζήτηση σε κάθε γραμμή του αρχείου, που εξετάστηκε για το όνομα του χρήστη, που είχαμε εισάγει. Σε περίπτωση ανεύρεσης του ονόματος του χρήστη αντέγραφε τη γραμμή στην οποία τον βρήκε και την εισήγαγε σε ένα νέο αρχείο. Το αρχείο αυτό λοιπόν περιελάμβανε όλες τις γραμμές τις οποίες είχε κάνει αναζήτηση και είχε βρει το πρόγραμμα. Στο τέλος δημιουργήθηκαν τρία (3) αρχεία (file.csv, email.csv, http.csv) για κάθε χρήστη, τα οποία περιελάμβαναν τη συνολική δραστηριότητά του για όλη τη χρονική περίοδο.

π.χ ACM2278_http.csv, ACM2278_email.csv, ACM2278_file.csv

Στο δεύτερο βήμα (Step Two), στο πρόγραμμα εισήχθησαν τα αποτελέσματα του πρώτου βήματος και καθορίστηκαν οι κανόνες για την αναζήτηση των απειλών και των κακόβουλων συμπεριφορών σε κάθε ένα από αυτά.



Εικόνα 4-2: User Interface δεύτερου βήματος προγράμματος

Στην κατηγορία website επιλέξαμε όρους, που συνδέονταν με κοινωνικά δίκτυα, με site αναζήτησης εργασίας, site κακόβουλου λογισμικού και site διαμοιρασμού αρχείων. Συγκεκριμένα εισάγαμε τους κάτωθι δεκαοκτώ (18) όρους, τους οποίους εν συνεχεία αναζήτησε το πρόγραμμα:

Myspace,

Facebook,

Twitter,

Wikileaks,

Monster,

LinkedIn,

Jobhuntersbible,

Simplyhired,

Indeed,

keylogger,

Dropbox,

Mediafire,

4Shared,

Skydrive,

iCloud,

Rapidshare,

Depositfiles,

Zippyshare.

Στην κατηγορία email επιλέξαμε όρους, που συνδέονταν με κοινωνικά δίκτυα, με site αναζήτησης εργασίας και site κακόβουλου λογισμικού. Συγκεκριμένα εισάγαμε τους κάτωθι δεκατέσσερις (14) όρους, τους οποίους εν συνεχεία αναζήτησε το πρόγραμμα: Facebook,

Twitter,

Wikileaks,

Dropbox,

Keylogger,

Experience,

Employees,

Employer,

Candidate,

Training,

Position,

Career,

Expert,

Requirements.

Επιπρόσθετα στην κατηγορία αυτή το πρόγραμμα έκανε αναζήτηση για επισυναπτόμενα αρχεία, τα οποία και χώρισε σε τρεις (3) μεγάλες κατηγορίες:

αρχεία με μέγεθος 50Kb – 100 Kb,

αρχεία με μέγεθος 100Kb - 200Kb και

αρχεία με μέγεθος μεγαλύτερο των 200Kb.

Τέλος, στην κατηγορία file, επιλέξαμε όρους, που συνδέονταν με αρχεία κακόβουλου λογισμικού (Keylogger), αρχεία που ενδεχομένως διέρρευσαν στο διαδίκτυο και αρχεία που ενδεχομένως διαμοιράστηκαν μέσω διαδικτύου. Συγκεκριμένα εισάγαμε τους κάτωθι δέκα (10) όρους, τους οποίους εν συνεχεία αναζήτησε το πρόγραμμα: Keylogger,

Wikileaks,

Dropbox,

Mediafire,

4Shared,

Skydrive,

iCloud,

Rapidshare,

Depositfiles,

Zippyshare.

Αποτέλεσμα ήταν η δημιουργία τριών (3) αρχείων για κάθε χρήστη (file.csv, email.csv, http.csv), τα οποία είχαν την δομή:

date (ημερομηνία),

user (όνομα χρήστη που αναζητήθηκε,

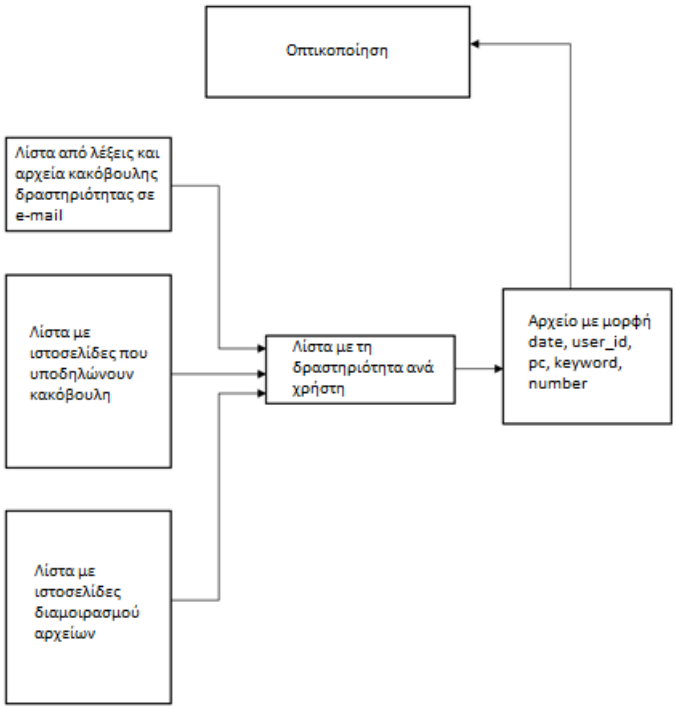
pc (υπολογιστής, από τον οποίο πραγματοποιήθηκε η ενέργεια),

keyword (τύπος της απειλής),

number (ένας αριθμός που αντιστοιχεί στην εκάστοτε απειλή) και περιείχαν τις αντίστοιχες ως ανωτέρω απειλές αρχείων.

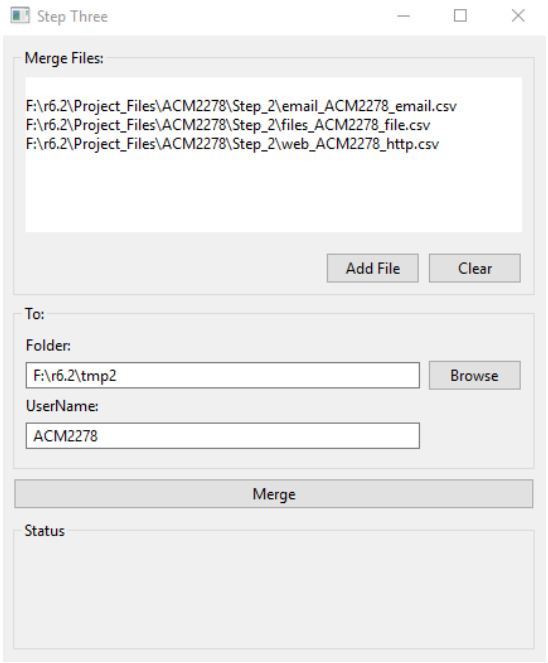
π.χ. web_ACM2278_http.csv, email_ACM2278_email.csv, files_ACM2278_file.csv

Ο αριθμός που προστέθηκε στο τέλος κάθε γραμμής του ανωτέρω αρχείου όριζε και ένα συγκεκριμένο τύπο παράνομης δραστηριότητας. Αντιστοιχούσε δε σε ένα μοναδικό χρώμα, το οποίο θα απεικονίζονταν, στο επόμενο στάδιο της οπτικοποίησης, στις εικόνες, που θα παράγονταν με την βιβλιοθήκη D3.js στην απεικόνιση της δραστηριότητας των χρηστών.



Εικόνα 4-3: Block διάγραμμα της διαδικασίας του πρώτου σταδίου

Στο τρίτο βήμα (Step Three) στο πρόγραμμα εισάγαμε τα τρία (3) αρχεία ανά χρήστη, που προέκυψαν από το δεύτερο βήμα, όπως εμφαίνεται στην εικόνα, που ακολουθεί:



Εικόνα 4-4: User Interface τρίτου βήματος προγράμματος

Σκοπός ήταν η συνένωσή τους (merge) σε ένα μοναδικό αρχείο, το οποίο περιλάμβανε όλη τη δραστηριότητα του εκάστοτε χρήστη για την εξεταζόμενη χρονική περίοδο και το οποίο είχε την δομή:

date (ημερομηνία),

user (όνομα χρήστη που αναζητήθηκε),

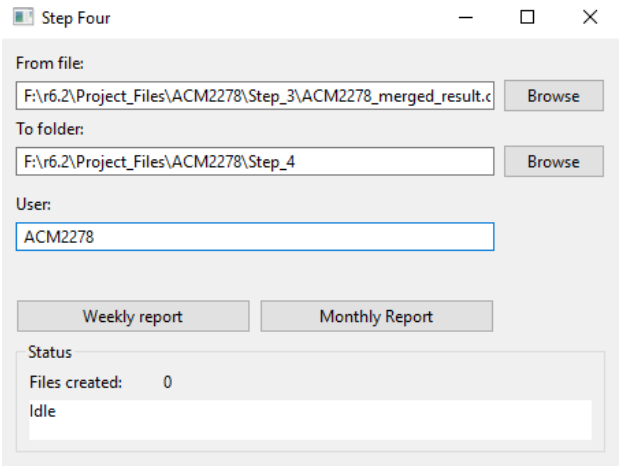
pc (υπολογιστής, από τον οποίο πραγματοποιήθηκε η ενέργεια),

keyword (τύπος της απειλής),

number (ένας αριθμός που αντιστοιχεί στην εκάστοτε απειλή).

π.χ. ACM2278_merged_result.csv

Στο τέταρτο βήμα (Step Four) έγινε εισαγωγή στο πρόγραμμα του ανωτέρω αρχείου που παράχθηκε στο ως άνω στάδιο προκειμένου να διαχωριστεί η εβδομαδιαία και μηνιαία δραστηριότητα ανά χρήστη, όπως απεικονίζεται κατωτέρω:



Εικόνα 4-5: User Interface τέταρτου βήματος προγράμματος

Τα αρχεία, που παρήχθησαν περιελάμβαναν την δραστηριότητα του χρήστη, τόσο ανά εβδομάδα, όσο και ανά μήνα για την εξεταζόμενη χρονική περίοδο με χρονολογική σειρά. Τέλος σημειώνεται ότι και τα αρχεία αυτά είχαν τη δομή:

date (ημερομηνία),

user (όνομα χρήστη που αναζητήθηκε),

pc (υπολογιστής, από τον οποίο πραγματοποιήθηκε η ενέργεια),

keyword (τύπος της απειλής),

number (ένας αριθμός που αντιστοιχεί στην εκάστοτε απειλή).

π.χ. sorted_2010_01.csv,

π.χ. sorted_w_2010_01_02.csv

4.4 Δεύτερο στάδιο πειράματος

Στο δεύτερο στάδιο του πειράματος τα τελικά αρχεία του προηγούμενου σταδίου εισήχθησαν στη βιβλιοθήκη D3.js της Java, για να οπτικοποιηθούν και να παραχθούν εικόνες, οι οποίες θα απεικόνιζαν με τη μορφή συγκεκριμένου γραφήματος την δραστηριότητα του κάθε χρήστη. Όπως αναφέρθηκε ανωτέρω, ο αριθμός που προστέθηκε στο τέλος κάθε γραμμής των αρχείων, που δημιουργήθηκαν στο προηγούμενο στάδιο όριζε ένα συγκεκριμένο τύπο παράνομης δραστηριότητας και συγκεκριμένα:

- 0: μπλε για site αναζήτησης εργασίας, κοινωνικών δικτύων, κ.α.
- 1: κόκκινο για site διαμοιρασμού αρχείων (file-sharing websites).
- 3: ροζ για απειλές ηλεκτρονικής αλληλογραφίας.
- 4: πράσινο για απειλές σε αρχεία.

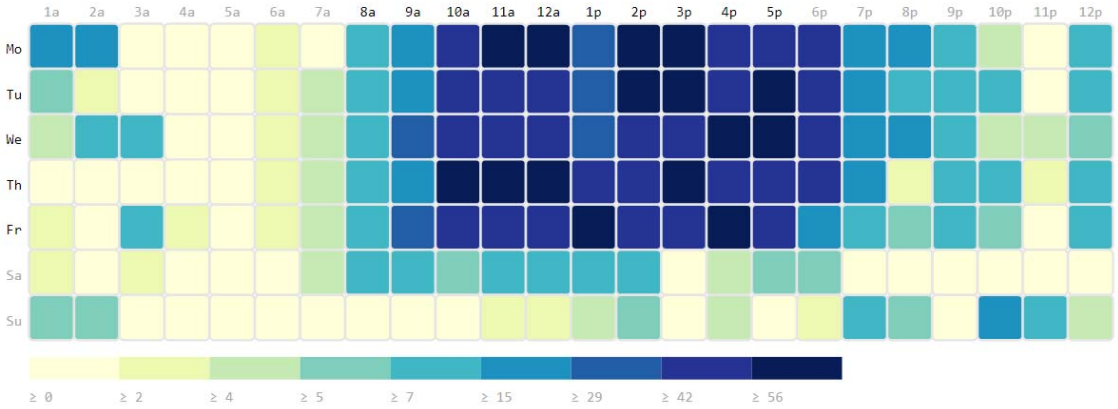
- 6: κίτρινο για επισυναπτόμενα αρχεία με μέγεθος 50Kb – 100 Kb.
- 7: πορτοκαλί για επισυναπτόμενα αρχεία με μέγεθος 100Kb - 200Kb.
- 8: καφέ για επισυναπτόμενα αρχεία με μέγεθος μεγαλύτερο των 200Kb.

Σχετικά με το πρόγραμμα, που χρησιμοποιήθηκε στο στάδιο αυτό για την οπτικοποίηση (visualization), το D3.js (https://d3js.org/) είναι μια βιβλιοθήκη JavaScript, που χρησιμοποιείται για τον χειρισμό εγγράφων και βασίζεται σε δεδομένα. Το D3 βοηθά στην απεικόνιση των δεδομένων χρησιμοποιώντας HTML, SVG και CSS. Η βιβλιοθήκη D3, βασίζεται σε σύγχρονα web – based πρότυπα, λειτουργεί μέσω αναβαθμισμένων προγραμμάτων περιήγησης (web browsers), όπως Firefox, Chrome, Safari, Opera, IE9+, συνδυάζοντας ισχυρές μεθόδους και στοιχεία απεικόνισης, που βασίζονται στο χειρισμό δεδομένων τύπου DOM. Η D3.js, ενσωματώνεται σε μια ιστοσελίδα HTML και χρησιμοποιεί ενσωματωμένες λειτουργίες JavaScript για την επιλογή στοιχείων, τη δημιουργία αντικειμένων SVG, το στυλ τους ή την προσθήκη μεταβάσεων, δυναμικών εφέ ή εργαλείων. Αυτά τα αντικείμενα μπορούν επίσης να είναι ευρέως διατυπωμένα χρησιμοποιώντας το CSS box model. Μεγάλα σύνολα δεδομένων μπορούν εύκολα να συνδεθούν με αντικείμενα SVG χρησιμοποιώντας απλές λειτουργίες D3.js για τη δημιουργία πλουσίων γραφημάτων κειμένου/γραφικών και διαγραμμάτων. Τα δεδομένα μπορούν να είναι σε διάφορες μορφές, με πιο συνηθισμένη την JSON, τιμές διαχωρισμένες με κόμμα (CSV) ή geoJSON, αλλά εάν απαιτείται, λειτουργίες JavaScript μπορούν να γραφούν για να διαβάσουν άλλες μορφές δεδομένων. Με ελάχιστη επιβάρυνση στους πόρους ενός υπολογιστικού συστήματος το D3 είναι εξαιρετικά γρήγορο, υποστηρίζοντας μεγάλα σύνολα δεδομένων και δυναμικές συμπεριφορές για αλληλεπίδραση μέσω στατικών αλλά και κινούμενων εικόνων.

Στην ενότητα examples της ιστοσελίδας https://d3js.org/, παρουσιάζονται παραδείγματα των διαθέσιμων μορφών των απεικονίσεων, που μπορούν να αναπαρασταθούν μέσω του D3.js, τα οποία είναι πολυάριθμα και μπορούν να ικανοποιήσουν τις ανάγκες και του πιο απαιτητικού χρήστη.

Στην παρούσα μεταπτυχιακή διατριβή, επιλέχθηκε η απεικόνιση τύπου **Day/Hour Heatmap**, επειδή ήταν η μοναδική από τους προτεινόμενους τύπους απεικονίσεων, που περιελάμβανε όλες τις ημέρες της εβδομάδας και όλες τις ώρες της ημέρας. Αυτός ο τύπος ήταν ο πλέον κατάλληλος

για να απεικονιστούν δεδομένα εβδομάδων και μηνών. Η μορφή της απεικόνισης εμφαίνεται στην παρακάτω εικόνα:



Day/Hour Heatmap

Εικόνα 4-6: Τύπος της απεικόνισης που επιλέχθηκε

Σκοπός του προγράμματος ήταν η απεικόνιση – οπτικοποίηση (visualization) των αρχείων, που δημιουργήθηκαν στο προηγούμενο στάδιο και στη συνέχεια ανάλογα με τη μορφή της κάθε παράστασης ο χαρακτηρισμός της δραστηριότητας, που απεικονίστηκε είτε ως κακόβουλη, είτε ως φυσιολογική. Ακολουθεί η περιγραφή του προγράμματος, που υλοποιήθηκε:

Αρχικά, κατασκευάστηκε το περιβάλλον σχεδίασης και ορίστηκαν οι διαστάσεις που απαιτούνταν για την σωστή σχεδίαση της εικόνας. Προς διευκόλυνσή μας, η σχεδίαση έγινε με τέτοιο τρόπο ώστε να περιλαμβάνει ολόκληρη την εικόνα για να μην χρειάζεται να γίνει scroll down με το ποντίκι για να την εμφανίσει. Περαιτέρω, ήταν απαραίτητο να λάβει χώρα σωστή ρύθμιση, με βάση την ανάλυση της οθόνης του υπολογιστή, ώστε η εικόνα να είναι ευδιάκριτη:

```
var margin = { top: 20, right: 0, bottom: 0, left: 30 },
width = 1400 - margin.left - margin.right,
height = 610 - margin.top - margin.bottom,
gridSize = Math.floor(width / 69),
legendElementWidth = gridSize*2,
```

Στη συνέχεια, σχεδιάστηκε το περιβάλλον μέσα στο οποίο πραγματοποιήθηκε η απεικόνιση. Αυτό περιελάμβανε δύο (2) άξονες, εκ των οποίων ο πρώτος (κάθετος) εμφανίζει τις ημέρες του

μήνα και ο δεύτερος (οριζόντιος) τις ώρες της ημέρας. Οι ημέρες, οι ώρες καθώς και τα χρώματα, που χρησιμοποιήθηκαν δηλώθηκαν στην αρχή του κώδικα του προγράμματος.

```
colors = ["#0d00b0","#ff0000","#888888","#ff0080","#34b600","#888888","#f0ff00","#ffa500","#4f3300","#888888"],
days = ["1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31"],
times = ["1a", "2a", "3a", "4a", "5a", "6a", "7a", "8a", "9a", "10a", "11a", "12a", "1p", "2p", "3p", "4p", "5p", "6p", "7p", "8p", "9p", "10p", "11p", "12p"];
```

Παράλληλα, καθορίστηκαν οι διαστάσεις της περιοχής σχεδίασης, σύμφωνα με το μέγεθος του grid, που είχε οριστεί. Το αποτέλεσμα ήταν η δημιουργία ενός χώρου, που αποτελούνταν από νοητά τετράγωνα, εκ των οποίων το καθένα αντιστοιχούσε σε κάθε ημέρα και ώρα. Κάθε φορά, που ενεργοποιούνταν ένα τετράγωνο, χρωματίζονταν με το χρώμα, που είχε οριστεί ανάλογα με τον αριθμό, που περιγράφηκε, και προέκυπτε το αποτέλεσμα.

```
var svg = d3.select("#chart").append("svg")
.attr("width", width + margin.left + margin.right)
.attr("height", height + margin.top + margin.bottom)
.append("g")
.attr("transform", "translate(" + margin.left + "," + margin.top + ")");

var dayLabels = svg.selectAll(".dayLabel")
.data(days)
.enter().append("text")
.text(function (d) { return d; })
.attr("x", 0)
.attr("y", function (d, i) { return i * gridSize; })
.style("text-anchor", "end")
.attr("transform", "translate(-6," + gridSize / 1.5 + ")")
.attr("class", function (d, i) { return ((i >= 0 && i <= 4) ? "dayLabel mono axis axis-workweek" : "dayLabel mono axis"); });

var timeLabels = svg.selectAll(".timeLabel")
.data(times)
.enter().append("text")
.text(function(d) { return d; })
.attr("x", function(d, i) { return i * gridSize; })
.attr("y", 0)
```

```
.style("text-anchor", "middle")
.attr("transform", "translate(" + gridSize / 2 + ", -6)")
.attr("class", function(d, i) { return ((i >= 7 && i <= 16) ? "timeLabel mono axis axis-worktime" : "timeLabel
mono axis"); });
```

Το επόμενο βήμα ήταν η εισαγωγή των δεδομένων των αρχείων, που προέκυψαν από το προηγούμενο βήμα. Για να υλοποιηθεί το βήμα αυτό, χρησιμοποιήθηκε η συνάρτηση d3.csv της βιβλιοθήκης του D3.js. Η συνάρτηση δέχτηκε ως είσοδο τα αρχεία csv με τη δραστηριότητα όλων των χρηστών και πραγματοποίησε την οπτικοποίηση (visualization), με βάση τον αριθμό του τελευταίου πεδίου του εκάστοτε csv αρχείου, το οποίο αντιστοιχούσε σε συγκεκριμένο χρώμα και άρα σε συγκεκριμένη δραστηριότητα.

```
var color = function(csvFile) {
d3.csv(dataset[j],
function(d) {
return {
date: +d.date,
user: +d.user,
pc: +d.pc,
day: +d.date.split(" ")[0].split("/")[1],
hour: +d.date.split("/")[2].split(" ")[1].split(":")[0],
keyword: +d.keyword,
number: +d.number
};
```

Στο τελευταίο τμήμα του προγράμματος, πραγματοποιήθηκε η λήψη του στοιχείου SVG, που δημιουργήθηκε, η μετατροπή του σε εικόνα τύπου png και η λήψη του στο σκληρό δίσκο του υπολογιστή.

```
var html = d3.select("svg")
.attr("version", 1.1)
.attr("xmlns", "http://www.w3.org/2000/svg")
.node().parentNode.innerHTML;

var image = new Image();
image.src = '' + window.btoa(html);
```

```
image.onload = function() {
var canvas = document.createElement("canvas");
canvas.width = image.width;
canvas.height = image.height;
var context = canvas.getContext("2d");
context.drawImage(image, 0, 0);

var a = document.createElement("a");
a.download = "image.png";
a.href = canvas.toDataURL("image/png");
document.body.appendChild(a);
a.click();
}
```

Οι εικόνες, που παρήχθησαν χρησιμοποιήθηκαν στο τελευταίο στάδιο του πειράματος για την εκπαίδευση του αλγορίθμου Τεχνητής Νοημοσύνης, μέσω του προγράμματος TensorFlow, για την αυτόματη κατηγοριοποίηση (classification) της δραστηριότητας σε φυσιολογική ή κακόβουλη. Κατωτέρω παρουσιάζεται ολόκληρος ο κώδικας του προγράμματος, που χρησιμοποιήθηκε:

```
<!DOCTYPE html>
<meta charset="utf-8">
<html>
<head>
<style>
rect.bordered {
stroke: #E6E6E6;
stroke-width:2px;
}

text.mono {
font-size: 9pt;
font-family: Consolas, courier;
fill: #000;
}

text.axis-workweek {
fill: #000;
}

text.axis-worktime {
```

```
fill: #000;
}
</style>
<script src="http://d3js.org/d3.v3.js"></script>
</head>
<body>
<div id="chart"></div>
<div id="dataset-picker">
</div>
<script type="text/javascript">
var margin = { top: 20, right: 0, bottom: 0, left: 30 },
width = 1400 - margin.left - margin.right,
height = 610 - margin.top - margin.bottom,
gridSize = Math.floor(width / 69),
legendElementWidth = gridSize*2,
buckets = 10,
colors = ["#0d00b0","#ff0000","#888888","#ff0080","#34b600","#888888","#f0ff00","#ffa500","#4f3300","#888888"],
days = ["1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31"],
times = ["1a", "2a", "3a", "4a", "5a", "6a", "7a", "8a", "9a", "10a", "11a", "12a", "1p", "2p", "3p", "4p", "5p", "6p", "7p", "8p", "9p", "10p", "11p", "12p"];
datasets = ["*.csv"];

var svg = d3.select("#chart").append("svg")
.attr("width", width + margin.left + margin.right)
.attr("height", height + margin.top + margin.bottom)
.append("g")
.attr("transform", "translate(" + margin.left + "," + margin.top + ")");

var dayLabels = svg.selectAll(".dayLabel")
.data(days)
.enter().append("text")
.text(function (d) { return d; })
.attr("x", 0)
.attr("y", function (d, i) { return i * gridSize; })
.style("text-anchor", "end")
.attr("transform", "translate(-6," + gridSize / 1.5 + ")")
.attr("class", function (d, i) { return ((i >= 0 && i <= 4) ? "dayLabel mono axis axis-workweek" : "dayLabel mono axis"); });
```

```
var timeLabels = svg.selectAll(".timeLabel")
.data(times)
.enter().append("text")
.text(function(d) { return d; })
.attr("x", function(d, i) { return i * gridSize; })
.attr("y", 0)
.style("text-anchor", "middle")
.attr("transform", "translate(" + gridSize / 2 + ", -6)")
.attr("class", function(d, i) { return ((i >= 7 && i <= 16) ? "timeLabel mono axis axis-worktime" : "timeLabel
mono axis"); });

var color = function(csvFile) {
d3.csv(datasets[j],
function(d) {
return {
date: +d.date,
user: +d.user,
pc: +d.pc,
day: +d.date.split(" ")[0].split("/")[1],
hour: +d.date.split("/")[2].split(" ")[1].split(":")[0],
keyword: +d.keyword,
number: +d.number
};
},
function(error, data) {
var colorScale = d3.scale.quantile()
.domain([0, buckets -1])
.range(colors);

var cards = svg.selectAll(".hour")
.data(data, function(d) {return d.day+':'+d.hour;});

cards.append("title");

cards.enter().append("rect")
.attr("x", function(d) { return (d.hour - 1) * gridSize; })
.attr("y", function(d) { return (d.day - 1) * gridSize; })
.attr("rx", 4)
.attr("ry", 4)
.attr("class", "hour bordered")
.attr("width", gridSize)
```

```
.attr("height", gridSize)
.style("fill", colors[0]);
cards.transition().duration(1000)
.style("fill", function(d) { return colorScale(d.number); });

cards.select("title").text(function(d) { return d.date+" : "+d.keyword; });

cards.exit().remove();

});
};

color(datasets[0]);

var html = d3.select("svg")
.attr("version", 1.1)
.attr("xmlns", "http://www.w3.org/2000/svg")
.node().parentNode.innerHTML;

var image = new Image();
image.src = '' + window.btoa(html);

image.onload = function() {
var canvas = document.createElement("canvas");
canvas.width = image.width;
canvas.height = image.height;
var context = canvas.getContext("2d");
context.drawImage(image, 0, 0);

var a = document.createElement("a");
a.download = "image.png";
a.href = canvas.toDataURL("image/png");
document.body.appendChild(a);
a.click();
}

</script>
</body>
</html>
```

Παρήχθησαν συνολικά χίλιες εκατόν ενενήντα εννέα (1199) εικόνες. Συγκεκριμένα δε για κάθε χρήστη παρήχθησαν οι εξής εικόνες :

- ACM2278: πενήντα (50) εικόνες, εκ των οποίων εννέα (9) απεικόνιζαν τη μηνιαία δραστηριότητά του και σαράντα μία (41) την εβδομαδιαία.

- CMP2946: ογδόντα εννέα (89) εικόνες, εκ των οποίων δεκαπέντε (15) απεικόνιζαν τη μηνιαία δραστηριότητά του και εβδομήντα τέσσερις (74) την εβδομαδιαία.

- PLJ1771: σαράντα έξι (46) εικόνες, εκ των οποίων οκτώ (8) απεικόνιζαν τη μηνιαία δραστηριότητά του και τριάντα οκτώ (38) την εβδομαδιαία.

- MBG3183: ενενήντα εννέα (99) εικόνες, εκ των οποίων δεκαεπτά (17) απεικόνιζαν τη μηνιαία δραστηριότητά του και ογδόντα δύο (82) την εβδομαδιαία.

- SAB1954: εκατόν μία (101) εικόνες, εκ των οποίων δεκαεπτά (17) απεικόνιζαν τη μηνιαία δραστηριότητά του και ογδόντα τέσσερις (84) την εβδομαδιαία.

- ABK0481: έντεκα (11) εικόνες, εκ των οποίων τρεις (3) απεικόνιζαν τη μηνιαία δραστηριότητά του και οκτώ (8) την εβδομαδιαία.

- CCB3055: ενενήντα εννέα (99) εικόνες, εκ των οποίων δεκαεπτά (17) απεικόνιζαν τη μηνιαία δραστηριότητά του και ογδόντα δύο (82) την εβδομαδιαία.

- KCB1975: ενενήντα οκτώ (98) εικόνες, εκ των οποίων δεκαεπτά (17) απεικόνιζαν τη μηνιαία δραστηριότητά του και ογδόντα μία (81) την εβδομαδιαία.

- ACG0312: ενενήντα επτά (97) εικόνες, εκ των οποίων δεκαεπτά (17) απεικόνιζαν τη μηνιαία δραστηριότητά του και ογδόντα (80) την εβδομαδιαία.

- CJM0584: δεκατρείς (13) εικόνες, εκ των οποίων δύο (2) απεικόνιζαν τη μηνιαία δραστηριότητά του και έντεκα (11) την εβδομαδιαία.

- SKB2635: ενενήντα εννέα (99) εικόνες, εκ των οποίων δεκαεπτά (17) απεικόνιζαν τη μηνιαία δραστηριότητά του και ογδόντα δύο (82) την εβδομαδιαία.

- CDE1846: εκατό (100) εικόνες, εκ των οποίων δεκαεπτά (17) απεικόνιζαν τη μηνιαία δραστηριότητά του και ογδόντα τρεις (83) την εβδομαδιαία.

- LAN2608: ενενήντα εννέα (99) εικόνες, εκ των οποίων δεκαεπτά (17) απεικόνιζαν τη μηνιαία δραστηριότητά του και ογδόντα δύο (82) την εβδομαδιαία.

- POD0750: ενενήντα εννέα (99) εικόνες, εκ των οποίων δεκαεπτά (17) απεικόνιζαν τη μηνιαία δραστηριότητά του και ογδόντα δύο (82) την εβδομαδιαία.

- PIM3569: ενενήντα εννέα (99) εικόνες, εκ των οποίων δεκαεπτά (17) απεικόνιζαν τη μηνιαία δραστηριότητά του και ογδόντα δύο (82) την εβδομαδιαία.

Στη συνέχεια, επιλέξαμε ποιές εξ αυτών συνιστούσαν κακόβουλη δραστηριότητα και ποιές αντίθετα συνιστούσαν φυσιολογική δραστηριότητα. Η επιλογή έγινε με συγκεκριμένα κριτήρια, ενδεικτικά αναφερομένων της πυκνότητας της δραστηριότητας σε συγκεκριμένα χρονικά διαστήματα, των χρωμάτων της καθώς και της ώρας, που διαπιστώθηκε σε περίπτωση εκτός εργασίμου ωραρίου.

Τελικώς, από το σύνολο των ανωτέρω εικόνων αξιολογήθηκε ότι επτακόσιες εξήντα εννέα (769) εξ αυτών περιείχαν κακόβουλη δραστηριότητα και τετρακόσιες τριάντα (430) φυσιολογική δραστηριότητα. Ειδικότερα, η δραστηριότητα, ανά χρήστη, ήταν η εξής:

- ACM2278: σαράντα οκτώ (48) εικόνες περιείχαν κακόβουλη δραστηριότητα και δύο (2) φυσιολογική δραστηριότητα.

- CMP2946: ογδόντα εννέα (89) εικόνες περιείχαν κακόβουλη δραστηριότητα και καμία (0) φυσιολογική δραστηριότητα.

- PLJ1771: τριάντα μία (31) εικόνες περιείχαν κακόβουλη δραστηριότητα και δεκαπέντε (15) φυσιολογική δραστηριότητα.

- MBG3183: ενενήντα επτά (97) εικόνες περιείχαν κακόβουλη δραστηριότητα και δύο (2) φυσιολογική δραστηριότητα.

- SAB1954: τριάντα έξι (36) εικόνες περιείχαν κακόβουλη δραστηριότητα και εξήντα πέντε (65) φυσιολογική δραστηριότητα.

- ABK0481: καμία (0) εικόνα δεν περιείχε κακόβουλη δραστηριότητα και έντεκα (11) περιέχουν φυσιολογική δραστηριότητα.

- CCB3055: ενενήντα τέσσερις (94) εικόνες περιείχαν κακόβουλη δραστηριότητα και πέντε (5) φυσιολογική δραστηριότητα.

- KCB1975: πενήντα δύο (52) εικόνες περιείχαν κακόβουλη δραστηριότητα και σαράντα έξι (46) φυσιολογική δραστηριότητα.

- ACG0312: καμία (0) εικόνα δεν περιείχε κακόβουλη δραστηριότητα και ενενήντα επτά (97) περιέχουν φυσιολογική δραστηριότητα.

- CJM0584: έξι (6) εικόνες περιείχαν κακόβουλη δραστηριότητα και επτά (7) φυσιολογική δραστηριότητα.

- SKB2635: σαράντα εννέα (49) εικόνες περιείχαν κακόβουλη δραστηριότητα και πενήντα (50) φυσιολογική δραστηριότητα.

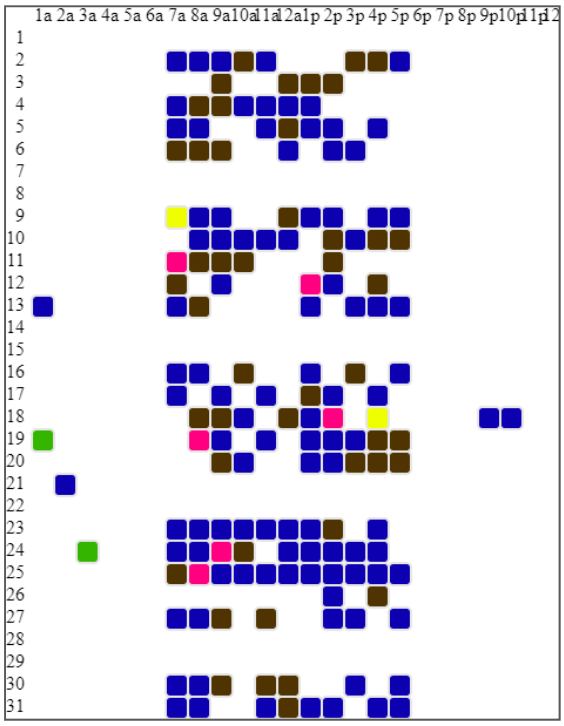
- CDE1846: ογδόντα επτά (87) εικόνες περιείχαν κακόβουλη δραστηριότητα και δεκατρείς (13) φυσιολογική δραστηριότητα.

- LAN2608: σαράντα επτά (47) εικόνες περιείχαν κακόβουλη δραστηριότητα και πενήντα δύο (52) φυσιολογική δραστηριότητα.

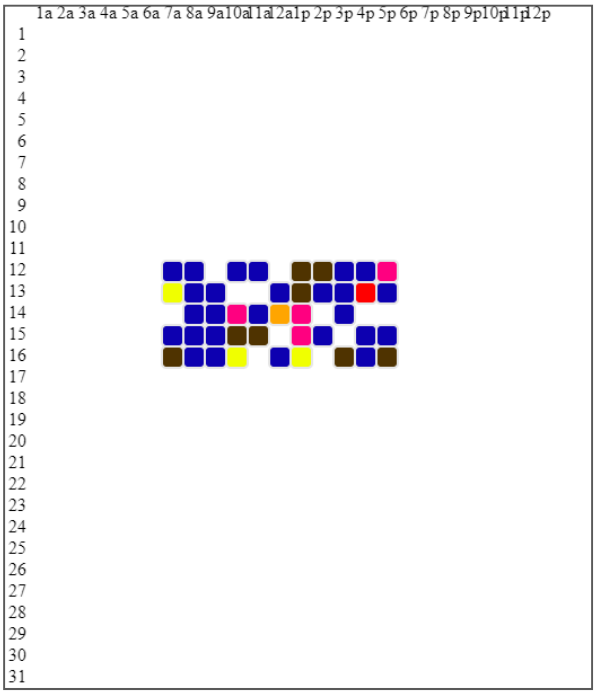
- POD0750: εβδομήντα πέντε (75) εικόνες περιείχαν κακόβουλη δραστηριότητα και είκοσι τέσσερις (24) φυσιολογική δραστηριότητα.

- PIM3569: πενήντα οκτώ (58) εικόνες περιείχαν κακόβουλη δραστηριότητα και σαράντα μία (41) φυσιολογική δραστηριότητα.

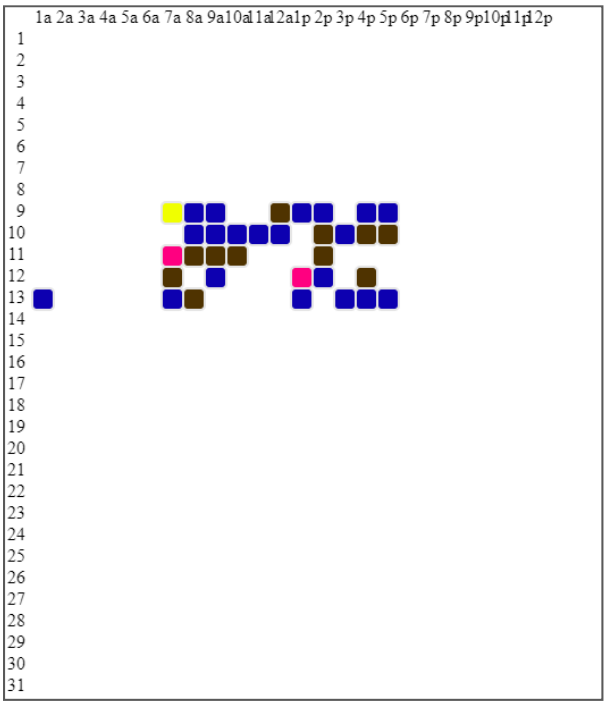
Ενδεικτικά παρατίθενται κατωτέρω έξι (6) εκ των εικόνων, που παρήχθησαν, εκ των οποίων οι τρεις (3) πρώτες κρίθηκαν ότι περιείχαν κακόβουλη δραστηριότητα και οι τρεις (3) επόμενες φυσιολογική δραστηριότητα:



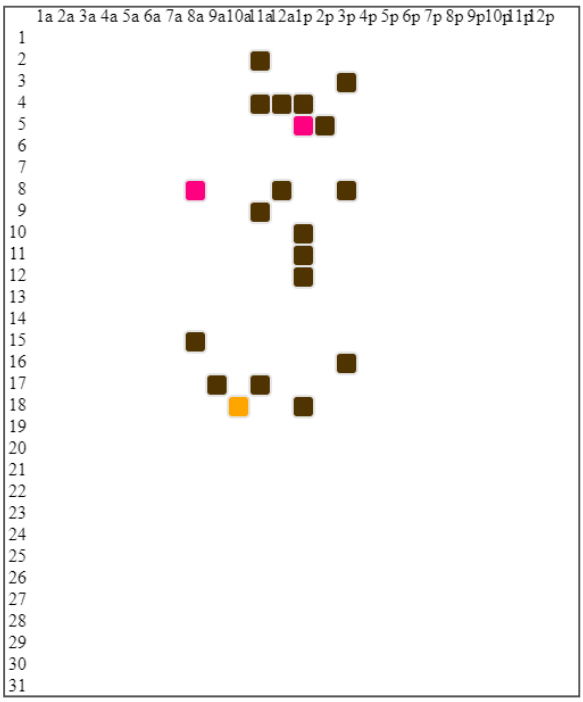
Εικόνα 4-7: Μηνιαία απεικόνιση κακόβουλης δραστηριότητας χρήστη



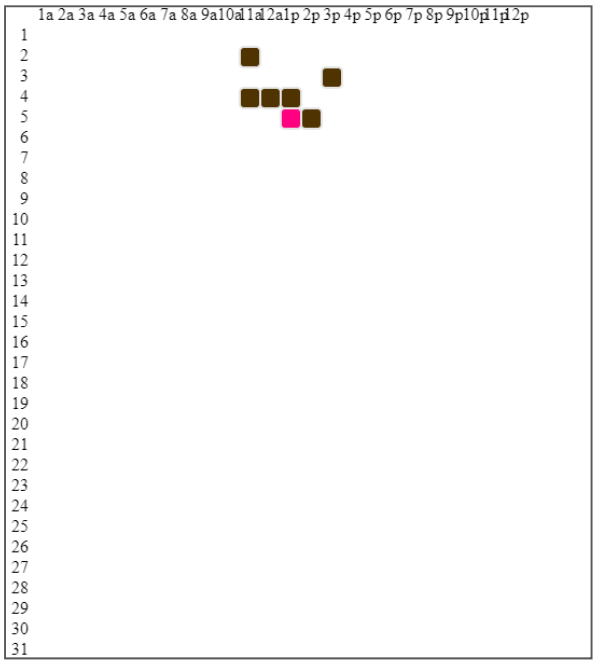
Εικόνα 4-8: Εβδομαδιαία απεικόνιση κακόβουλης δραστηριότητας χρήστη



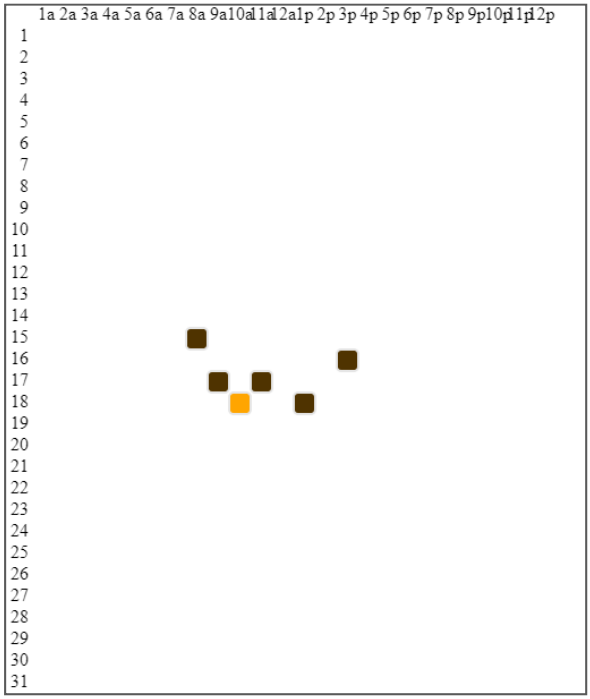
Εικόνα 4-9: Εβδομαδιαία απεικόνιση κακόβουλης δραστηριότητας χρήστη



Εικόνα 4-10: Μηνιαία απεικόνιση φυσιολογικής δραστηριότητας χρήστη



Εικόνα 4-11: Εβδομαδιαία απεικόνιση φυσιολογικής δραστηριότητας χρήστη



Εικόνα 4-12: Εβδομαδιαία απεικόνιση φυσιολογικής δραστηριότητας χρήστη

4.5 Τρίτο στάδιο πειράματος

Στο τρίτο στάδιο του πειράματος χρησιμοποιήθηκε η βιβλιοθήκη Tensorflow της Google. Το TensorFlow™ (https://www.tensorflow.org/) είναι μια βιβλιοθήκη λογισμικού ανοιχτού κώδικα για αριθμητικούς υπολογισμούς, η οποία χρησιμοποιεί γραφήματα ροής δεδομένων. Οι κόμβοι στο γράφημα αντιπροσωπεύουν μαθηματικές λειτουργίες, ενώ οι άκρες των γραφημάτων αντιπροσωπεύουν τις πολυδιάστατες συστοιχίες δεδομένων (tensors), που επικοινωνούν μεταξύ τους. Η ευέλικτη αρχιτεκτονική δίνει τη δυνατότητα στο χρήστη να αναπτύξει μοντέλα υπολογισμών σε μία ή περισσότερες CPU ή GPU σε υπολογιστή, διακομιστή (server) ή κινητή συσκευή με ένα μόνο API. Το TensorFlow αναπτύχθηκε αρχικά από ερευνητές και μηχανικούς της Google, μέσω του ερευνητικού οργανισμού Machine Intelligence της Google για σκοπούς διερεύνησης της Τεχνητής Νοημοσύνης και της έρευνας σε βαθιά Νευρωνικά Δίκτυα. Το σύστημα βέβαια μπορεί να εφαρμοστεί σε ευρύ φάσμα και άλλων τομέων εκτός των Νευρωνικών Δικτύων.

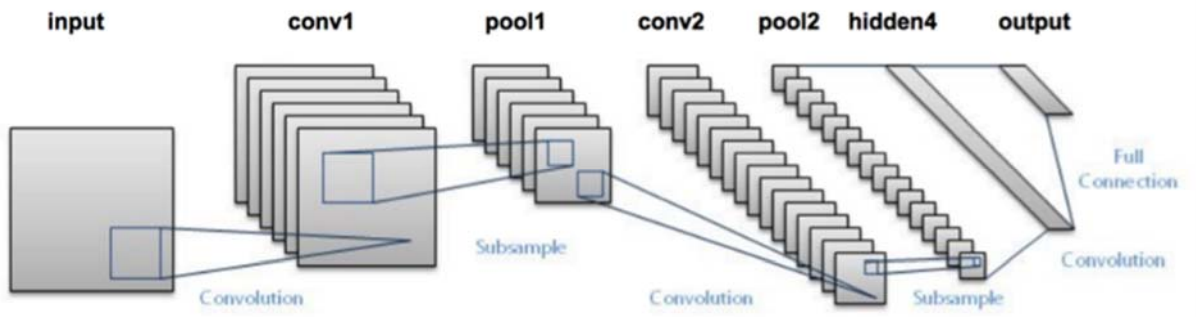
Στην παρούσα μεταπτυχιακή διατριβή, χρησιμοποιήθηκε το Tensorflow για την υλοποίηση ενός CNN και την κατηγοριοποίηση των εικόνων (image classification), που προέκυψαν από το προηγούμενο στάδιο. Σκοπός ήταν η υλοποίηση ενός συστήματος έξι (6) επιπέδων (layers), το οποίο θα αναγνώριζε εάν η εικόνα, που θα «διαβάζε» ως είσοδο (input), χαρακτηρίζονταν ως φυσιολογική ή ως κακόβουλη.

Στον τομέα των νευρωνικών δικτύων, τα CNN αποτελούν μια κλάση βαθιών τεχνητών νευρωνικών δικτύων με ευθεία τροφοδοσία προς τα εμπρός και έχει εφαρμοστεί επιτυχώς στην ανάλυση οπτικών εικόνων.

Τα CNN χρησιμοποιούν μια παραλλαγή πολλαπλών στρώσεων (perceptrons) που έχουν σχεδιαστεί για να απαιτούν ελάχιστη επεξεργασία. Είναι επίσης γνωστά ως αμετάβλητα τεχνητά νευρωνικά δίκτυα, βασισμένα στην αρχιτεκτονική των κοινών βαρών (weights).

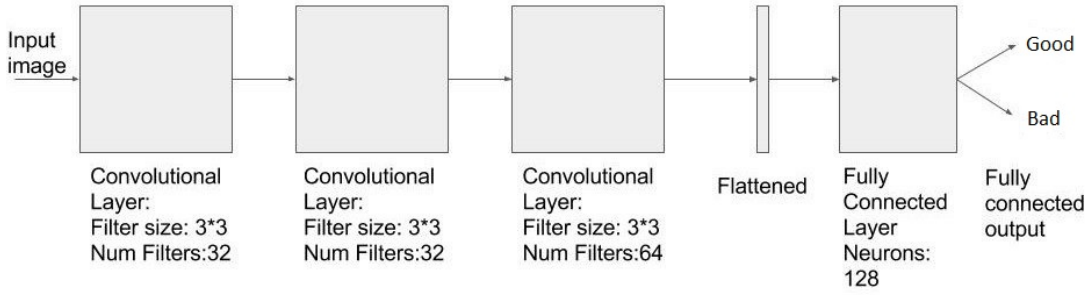
Τα CNN χρησιμοποιούν σχετικά μικρή προεπεξεργασία σε σύγκριση με άλλους αλγόριθμους ταξινόμησης εικόνων. Αυτό σημαίνει ότι το δίκτυο μαθαίνει με ευκολία τα φίλτρα που κατασκευάστηκαν με παραδοσιακούς αλγόριθμους. Αυτή η ανεξαρτησία από τις προηγούμενες γνώσεις είναι ένα σημαντικό πλεονέκτημα των CNN δικτύων.

Έχουν εφαρμογές στην αναγνώριση εικόνων και βίντεο, συστήματα συστημένων και επεξεργασία φυσικής γλώσσας.



Εικόνα 4-13: Σχηματική διάταξη ενός CNN δικτύου

Το σύστημα, που υλοποιήθηκε εμφαίνεται στο παρακάτω σχήμα:



Εικόνα 4-14: Σχηματική διάταξη του αλγορίθμου CNN που υλοποιήθηκε

Όπως παρατηρείται, αποτελείται από ένα συνελικτικό επίπεδο (convolutional layer), το οποίο δέχεται σαν είσοδο την εικόνα, που θέλουμε να κατηγοριοποιήσουμε. Η εικόνα αυτή είναι η οπτικοποίηση (visualization) της δραστηριότητας ενός φυσιολογικού ή ενός κακόβουλου χρήστη. Στη συνέχεια ακολουθούν δύο (2) ακόμα όμοια στρώματα, τα οποία οδηγούνται σε ένα στρώμα κανονικοποίησης της εικόνας (flattened). Τέλος, υπάρχουν δύο (2) ακόμη συνελικτικά επίπεδα, το δεύτερο από τα οποία αποτελείται από δύο (2) εξόδους, οι οποίες αντιπροσωπεύουν την πιθανότητα η δραστηριότητα του χρήστη να χαρακτηρίζεται ως φυσιολογική ή ως κακόβουλη.

4.5.1 Βιβλιοθήκη

Οι βιβλιοθήκες και οι συναρτήσεις, που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος ήταν οι εξής:

- OpenCV: Βιβλιοθήκη για την ανάγνωση και επεξεργασία εικόνων.

- Συνάρτηση Shape: Χρησιμοποιείται για τη ρύθμιση του μεγέθους των συστοιχιών των δεδομένων (tensors).

- Συνάρτηση Softmax: Μετατρέπει το διάνυσμα K διαστάσεων «X», που περιέχει πραγματικές τιμές, στο ίδιο διαμορφωμένο διάνυσμα πραγματικών τιμών στην περιοχή του (0, 1), του οποίου το άθροισμα είναι 1. Εφαρμόστηκε η συνάρτηση Softmax στην έξοδο του συνελικτικού νευρωνικού στρώματος, για να μετατρέψει την έξοδο σε πιθανότητα για κάθε μια κατηγορία.

$$o(x)_j = \frac{e^{x_i}}{\sum_{n=1}^{N} e^{x_n}} \; for \; j = 1 \ldots . N$$

Εικόνα 4-15: Μαθηματικός τύπος της συνάρτησης softmax

4.5.2 Εικόνες – Είσοδος συστήματος

Στο προηγούμενο στάδιο παρήχθησαν χίλιες εκατόν ενενήντα εννέα (1199) εικόνες, εκ των οποίων – όπως αναφέρθηκε – τετρακόσιες τριάντα (430) αξιολογήθηκε ότι περιέχουν φυσιολογική δραστηριότητα. Για το λόγο αυτό έγινε επιλογή αντίστοιχου αριθμού εικόνων με κακόβουλη δραστηριότητα. Τελικώς, για το τρίτο στάδιο του πειράματος χρησιμοποιήθηκαν συνολικά οκτακόσιες εξήντα (860) εικόνες, από τις οποίες οι οκτακόσιες σαράντα (840) αποτέλεσαν τις εικόνες για την εκπαίδευση του αλγορίθμου (Training Data), ως εξής:

- Δεδομένα εκπαίδευσης (Training Data): Χρησιμοποιήθηκε το 80% των εικόνων για εκπαίδευση.

- Δεδομένα επικύρωσης (Validation Data): Για την επικύρωση χρησιμοποιήθηκε το 20% των εικόνων της εκπαίδευσης.

Τέλος, οι υπόλοιπες είκοσι (20) εικόνες αποτέλεσαν τις εικόνες για την δοκιμή του αλγορίθμου (Testing Data).

- Δεδομένα δοκιμών (Testing Data): Αποτελούν ανεξάρτητα δεδομένα (εικόνες) για δοκιμές του συστήματος και εξαγωγή των συμπερασμάτων των προβλέψεών του.

Ο στόχος της εκπαίδευσης του δικτύου, ήταν η εκμάθηση των κατάλληλων τιμών των βαρών για όλους τους νευρώνες του, οι οποίοι εργάζονταν για να κάνουν ταξινόμηση μεταξύ μιας φυσιολογικής και μιας κακόβουλης δραστηριότητας. Η αρχική τιμή αυτών των βαρών μπορεί να είναι οποιαδήποτε αλλά λειτουργεί καλύτερα εάν ληφθούν κανονικές κατανομές (με μέση μηδενική και μικρή διακύμανση). Ο κώδικας, που χρησιμοποιήθηκε για το σκοπό αυτό φαίνεται παρακάτω:

```
def create_weights(shape):
return tf.Variable(tf.truncated_normal(shape, stddev=0.05), name='weights')
def create_biases(size):
return tf.Variable(tf.constant(0.05, shape=[size]), name='biases')
```

4.5.3 Ορισμός των επιπέδων του νευρωνικού δικτύου

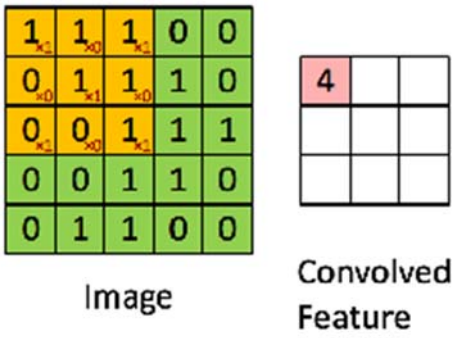
Το βασικότερο τμήμα του προγράμματος, ήταν ο ορισμός των επιπέδων (layers) του νευρωνικού δικτύου, που υλοποιήθηκε, τα οποία ήταν τα κάτωθι:

- Συνελικτικό επίπεδο (Convolutional Layer): Δημιουργείται με τη βοήθεια της συνάρτησης **tf.nn.conv2d**. Ο πρωταρχικός σκοπός της λειτουργίας αυτής είναι να εξάγει τα κύρια χαρακτηριστικά από την εικόνα εισόδου. Διατηρεί τη σχέση μεταξύ των εικονοστοιχείων (pixels), με την εκμάθηση των χαρακτηριστικών της εικόνας, χρησιμοποιώντας μικρά τετράγωνα (blocks) δεδομένων εισόδου.
  Οι είσοδοι, που δέχεται είναι οι εξής :

  - Input: η έξοδος (ενεργοποίηση) από το προηγούμενο επίπεδο. Τυπικά, στο πρώτο στρώμα περιελίξεων, οι εικόνες που περνούν, έχουν μέγεθος που ισούται με: πλάτος Χ ύψος Χ αριθμό καναλιών (num_channels).

- Φίλτρο: μεταβλητές με δυνατότητα κατάρτισης, που καθορίζουν το φίλτρο. Εκκινεί με μια τυχαία κανονική κατανομή και εκπαιδεύεται για αυτά τα βάρη. Είναι σημαντικό να σημειωθεί ότι τα φίλτρα λειτουργούν ως ανιχνευτές χαρακτηριστικών από την αρχική εικόνα εισόδου. Στην πράξη, ένα CNN μαθαίνει τις τιμές αυτών των φίλτρων μόνο του, κατά τη διάρκεια της εκπαιδευτικής διαδικασίας (αν και θα πρέπει ακόμα να καθορίσουμε παραμέτρους όπως ο αριθμός των φίλτρων, το μέγεθός τους, η αρχιτεκτονική του δικτύου κλπ. πριν από τη διαδικασία εκπαίδευσης). Όσο μεγαλύτερος είναι ο αριθμός των φίλτρων που διαθέτουμε, τόσο περισσότερες λειτουργίες και χαρακτηριστικά της εικόνας εξάγονται άρα επέρχεται και πιο ακριβής αναγνώριση σε δεδομένες εικόνες και σχέδια που τροφοδοτούν το δίκτυο (https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/).

- Βήματα (Strides): Καθορίζουν πόσο μετακινείται το φίλτρο όταν συμβαίνουν οι συνελίξεις (convolutions). Στο σχήμα που φαίνεται παρακάτω, το φίλτρο απεικονίζεται με κίτρινο χρώμα και περνά τμηματικά πάνω από την εικόνα ολισθαίνοντας με ένα συγκεκριμένο αριθμό βημάτων, φιλτράροντάς την.



Εικόνα 4-16: Απεικόνιση διαδικασίας της συνέλιξης

Το αποτέλεσμα είναι να παραχθεί μία νέα εικόνα που θα είναι ουσιαστικά το προϊόν της συνελικτικής διαδικασίας.

- Padding ή zero - padding: Είναι η συμπλήρωση της εικόνας εισόδου με μηδενικά pixels, προκειμένου το φίλτρο να μπορέσει να εφαρμοστεί επακριβώς επάνω σε αυτή. Η τιμή SAME

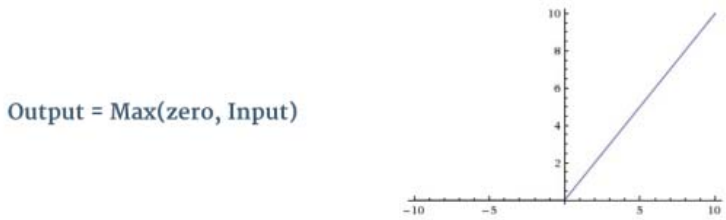
σημαίνει ότι θα πρέπει να πληκτρολογηθεί η είσοδος έτσι ώστε οι έξοδοι x, y να είναι ίδιες με εκείνες της εισόδου.

Μετά την συνέλιξη, προστέθηκαν οι πολώσεις (biases) αυτού του νευρώνα, οι οποίες ήταν επίσης μαθησιακές και εκπαιδευτικές για το σύστημα. Και πάλι η εκπαίδευση ξεκίνησε με τυχαία κανονική κατανομή και το σύστημα διδάχτηκε τις τιμές κατά τη διάρκειά της. Η συνάρτηση, που χρησιμοποιήθηκε ήταν η ακόλουθη: **nn.max_pool**.

```
tf.nn.max_pool(value=layer,
 ksize=[1, 2, 2, 1],
 strides=[1, 2, 2, 1],
 padding='SAME')
```

- Εφαρμόστηκε η συνάρτηση ενεργοποίησης RELU, η οποία έλαβε ως είσοδο, την έξοδο της nn.max.pool.

Η ReLU είναι μια μη γραμμική λειτουργία. Η παραγωγή του δίνεται από:



Εικόνα 4-17: Γραφική απεικόνιση της συνάρτησης ReLU

Η λειτουργία της, βασίζεται σε στοιχεία (εφαρμόζεται ανά εικονοστοιχείο) και αντικαθιστά όλες τις αρνητικές τιμές τους στον πίνακα των χαρακτηριστικών της εικόνας, με μηδέν. Ο σκοπός της είναι να εισάγει μη γραμμικότητα στο ConvNet, αφού τα περισσότερα από τα δεδομένα που θέλουμε να μάθει το ConvNet είναι μη γραμμικά.

Η τελική μορφή του συνελικτικού επιπέδου, που δημιουργήθηκε ήταν η εξής:

```
def create_convolutional_layer(input,
 num_input_channels,
 conv_filter_size,
 num_filters, name='conv'):
with tf.name_scope(name)
weights = create_weights(shape=[conv_filter_size, conv_filter_size, num_input_channels,    num_filters])
```

```
biases = create_biases(num_filters)

layer = tf.nn.conv2d(input=input,
 filter=weights,
 strides=[1, 1, 1, 1],
 padding='SAME')

layer += biases

layer = tf.nn.max_pool(value=layer,
ksize=[1, 2, 2, 1],
strides=[1, 2, 2, 1],
padding='SAME')
layer = tf.nn.relu(layer)

return layer
```

- Επίπεδο Συμπίεσης (Flattening Layer): Η έξοδος ενός συνελικτικού στρώματος (convolutional layer), είναι ένας πολυδιάστατος τανυστής (tensor), ο οποίος πρέπει να μετατραπεί σε μονοδιάστατο. Αυτό λαμβάνει χώρα στο επίπεδο συμπίεσης. Χρησιμοποιήθηκε απλώς η λειτουργία αναδιαμόρφωσης (reshape) για να δημιουργήσουμε έναν ενιαίο τρισδιάστατο τανυστή, όπως ορίζεται παρακάτω:

```
def create_flatten_layer(layer):
layer_shape = layer.get_shape()
num_features = layer_shape[1:4].num_elements()
layer = tf.reshape(layer, [-1, num_features])

return layer
```

- Πλήρως συνδεδεμένο επίπεδο (Fully Connected Layer): Ορίζουμε μια συνάρτηση για να δημιουργηθεί ένα πλήρως συνδεδεμένο επίπεδο. Όπως και σε κάθε άλλο επίπεδο, δηλώθηκαν τα βάρη και οι πολώσεις (biases) ως τυχαίες κανονικές κατανομές. Στη συνέχεια, λήφθησαν όλες οι είσοδοι και εφαρμόστηκε ο θεμελιώδης τύπος του νευρωνικού δικτύου, δηλ. z = wx + b.

```
def create_fc_layer(input,
 num_inputs,
 num_outputs,
```

```
 use_relu=True, name='fc'):
with tf.name_scope(name):
        weights = create_weights(shape=[num_inputs, num_outputs])
        biases = create_biases(num_outputs)

layer = tf.matmul(input, weights) + biases
```

- Placeholders και είσοδοι: Οι placeholders νοούνται ως μεταβλητές, που αποθήκευσαν προς επεξεργασία τις εικόνες εκπαίδευσης που εισήχθησαν. Όλες οι εικόνες εισόδου αναγνώστηκαν στο αρχείο dataset.py και έχουν μέγεθος έως 128 x 128 x 3. Το σύμβολο εισαγωγής x δημιουργήθηκε με τη μορφή [None, 128, 128, 3]. Η πρώτη διάσταση (None), σημαίνει ότι είναι δυνατό να εισαχθεί οποιοσδήποτε αριθμός εικόνων. Αναλόγως, δημιουργήθηκε μία μεταβλητή κράτησης θέσης y_true για την αποθήκευση των προβλέψεων. Για κάθε εικόνα, ορίστηκαν δύο (2) έξοδοι που αντιπροσωπεύουν τις πιθανότητες της κάθε κατηγορίας (good_bad χρήστης).

```
x = tf.placeholder(tf.float32, shape=[None, img_size,img_size,num_channels], name='x')

y_true = tf.placeholder(tf.float32, shape=[None, num_classes], name='y_true')
y_true_cls = tf.argmax(y_true, dimension=1)
```

- Σχεδίαση δικτύου (Network Design): Χρησιμοποιήθηκαν οι συναρτήσεις που ορίστηκαν παραπάνω, για τη δημιουργία των διαφόρων επιπέδων (layers) του συστήματος, που σχεδιάστηκε:

```
layer_conv1 = create_convolutional_layer(input=x,
 num_input_channels=num_channels,
 conv_filter_size=filter_size_conv1,
 num_filters=num_filters_conv1)

layer_conv2 = create_convolutional_layer(input=layer_conv1,
 num_input_channels=num_filters_conv1,
 conv_filter_size=filter_size_conv2,
 num_filters=num_filters_conv2)

layer_conv3= create_convolutional_layer(input=layer_conv2,
  num_input_channels=num_filters_conv2,
 conv_filter_size=filter_size_conv3,
 num_filters=num_filters_conv3)
```

```
layer_flat = create_flatten_layer(layer_conv3)

layer_fc1 = create_fc_layer(input=layer_flat,
 num_inputs=layer_flat.get_shape()[1:4].num_elements(),
 num_outputs=fc_layer_size,
 use_relu=True)
 layer_fc2 = create_fc_layer(input=layer_fc1,
 num_inputs=fc_layer_size,
 num_outputs=num_classes,
 use_relu=False)
```

- Προβλέψεις: Η μεταβλητή y_pred, που δημιουργήθηκε, αντιστοιχεί στην πιθανότητα, η εικόνα που εισάγεται να συνιστά φυσιολογική ή κακόβουλη δραστηριότητα. Η κατηγορία με το μεγαλύτερο ποσοστό ως τιμή πρόβλεψης, είναι και η πιο πιθανή να ισχύει.

  Στη συνέχεια, ορίστηκε ως παράμετρος του προγράμματος το κόστος (cost), ώστε να φτάσουμε στη βέλτιστη τιμή των βαρών. Ως κόστος ορίζεται το ποσοστό απόκλισης της πραγματικής από την επιθυμητή τιμή. Χρησιμοποιήθηκε μια μεταβλητή, η οποία υπολογίστηκε χρησιμοποιώντας τη συνάρτηση του Tensorflow: softmax_cross_entropy_with_logits. Η τελευταία έλαβε την έξοδο του τελευταίου πλήρως συνδεδεμένου επιπέδου και των ετικετών του (y_values) για τον υπολογισμό της τιμής cross_entropy, της οποίας ο μέσος όρος έδωσε το κόστος.

```
with tf.name_scope("cross_ent"):
        cross_entropy = tf.nn.softmax_cross_entropy_with_logits(logits=layer_fc2,
        labels=y_true)
        cost = tf.reduce_mean(cross_entropy)
```

- Βελτιστοποίηση (Optimization): Η βελτιστοποίηση βοηθά στην όσο το δυνατό μεγαλύτερη μείωση του κόστους. Για να επιτευχθεί αυτό, χρησιμοποιήθηκε η βιβλιοθήκη του Tensorflow, AdamOptimizer, η οποία ρύθμισε και βελτιστοποίησε και τις τιμές των βαρών (weights). Η ελαχιστοποίηση του κόστους έλαβε χώρα με ρυθμό εκμάθησης της τάξης του 0,0001. Οι εικόνες εκπαίδευσης εισήχθησαν ανά παρτίδα των 10 (batch_size) σε κάθε επανάληψη. Το κόστος υπολογίστηκε με τη βοήθεια κάποιων εικόνων, που προήλθαν από τις εικόνες εκπαίδευσης και ονομάστηκαν εικόνες

επικύρωσης (validation images). Αυτό έγινε για να αποφευχθούν τυχόν καθυστερήσεις στην εκτέλεση της ρουτίνας εκπαίδευσης του αλγορίθμου.

```
def train(num_iteration):
global total_iterations

for i in range(total_iterations,
 total_iterations + num_iteration):

x_batch, y_true_batch, _, cls_batch = data.train.next_batch(batch_size)
x_valid_batch, y_valid_batch, _, valid_cls_batch = data.valid.next_batch(batch_size)

feed_dict_tr = {x: x_batch,
 y_true: y_true_batch}
feed_dict_val = {x: x_valid_batch,
y_true: y_valid_batch}

session.run(optimizer, feed_dict=feed_dict_tr)

if i % int(data.train.num_examples/batch_size) == 0:
val_loss = session.run(cost, feed_dict=feed_dict_val)
epoch = int(i / int(data.train.num_examples/batch_size))

show_progress(epoch, feed_dict_tr, feed_dict_val, val_loss)
saver.save(session, 'good-bad-model')

total_iterations += num_iteration
```

- Πρόβλεψη (Prediction): Μετά την ολοκλήρωση της διαδικασίας εκπαίδευσης του αλγορίθμου, εκτελέστηκε το πρόγραμμα πρόβλεψης. Συγκεκριμένα, εκτελέστηκε το αρχείο predict.py και έγινε εισαγωγή της εικόνας, που επιθυμούσαμε να ελεγχθεί. Κατωτέρω, παρουσιάζεται ο κώδικας, που χρησιμοποιήθηκε:

```
dir_path = os.path.dirname(os.path.realpath(__file__))
image_path=sys.argv[1]
filename = dir_path +'/' +image_path
image_size=128
num_channels=3
images = []
image = cv2.imread(filename)
```

```
image = cv2.resize(image, (image_size, image_size),0,0, cv2.INTER_LINEAR)
images.append(image)
images = np.array(images, dtype=np.uint8)
images = images.astype('float32')
images = np.multiply(images, 1.0/255.0)
x_batch = images.reshape(1, image_size,image_size,num_channels)

sess = tf.Session()
saver = tf.train.import_meta_graph('good-bad-model.meta')
saver.restore(sess, tf.train.latest_checkpoint('./'))
graph = tf.get_default_graph()

y_pred = graph.get_tensor_by_name("y_pred:0")

x= graph.get_tensor_by_name("x:0")
y_true = graph.get_tensor_by_name("y_true:0")
y_test_images = np.zeros((1, 2))

feed_dict_testing = {x: x_batch, y_true: y_test_images}
result=sess.run(y_pred, feed_dict=feed_dict_testing)
print(result)
```

Εν προκειμένω η εκπαίδευση του αλγορίθμου ολοκληρώθηκε και καταγράφηκαν τα αποτελέσματα: Training Accuracy: 100.0%, Validation Accuracy: 90.6%, Validation Loss: 0.582, όπως εμφαίνεται κατωτέρω:

```
bill@bill-VirtualBox:~/Επιφάνεια εργασίας/InsiderThreats$ python3 train.py
Going to read training images
Now going to read good files (Index: 0)
Now going to read bad files (Index: 1)
Complete reading input data. Will Now print a snippet of it
Number of files in Training-set:        672
Number of files in Validation-set:      168
Training Epoch 1 --- Training Accuracy:  50.0%, Validation Accuracy:  62.5%,  Validation Loss: 0.686
Training Epoch 2 --- Training Accuracy:  50.0%, Validation Accuracy:  43.8%,  Validation Loss: 0.708
Training Epoch 3 --- Training Accuracy:  50.0%, Validation Accuracy:  53.1%,  Validation Loss: 0.676
Training Epoch 4 --- Training Accuracy:  78.1%, Validation Accuracy:  62.5%,  Validation Loss: 0.654
Training Epoch 5 --- Training Accuracy:  81.2%, Validation Accuracy:  68.8%,  Validation Loss: 0.618
Training Epoch 6 --- Training Accuracy:  78.1%, Validation Accuracy:  71.9%,  Validation Loss: 0.623
Training Epoch 7 --- Training Accuracy:  87.5%, Validation Accuracy:  90.6%,  Validation Loss: 0.381
Training Epoch 8 --- Training Accuracy:  84.4%, Validation Accuracy:  78.1%,  Validation Loss: 0.528
Training Epoch 9 --- Training Accuracy:  81.2%, Validation Accuracy:  75.0%,  Validation Loss: 0.528
Training Epoch 10 --- Training Accuracy:  84.4%, Validation Accuracy:  90.6%,  Validation Loss: 0.384
Training Epoch 11 --- Training Accuracy:  87.5%, Validation Accuracy:  78.1%,  Validation Loss: 0.557
Training Epoch 12 --- Training Accuracy:  84.4%, Validation Accuracy:  87.5%,  Validation Loss: 0.269
Training Epoch 13 --- Training Accuracy:  84.4%, Validation Accuracy:  81.2%,  Validation Loss: 0.394
Training Epoch 14 --- Training Accuracy:  87.5%, Validation Accuracy:  81.2%,  Validation Loss: 0.350
Training Epoch 15 --- Training Accuracy:  90.6%, Validation Accuracy:  90.6%,  Validation Loss: 0.263
Training Epoch 16 --- Training Accuracy:  90.6%, Validation Accuracy:  84.4%,  Validation Loss: 0.324
Training Epoch 17 --- Training Accuracy:  96.9%, Validation Accuracy:  93.8%,  Validation Loss: 0.190
Training Epoch 18 --- Training Accuracy:  96.9%, Validation Accuracy:  90.6%,  Validation Loss: 0.245
Training Epoch 19 --- Training Accuracy: 100.0%, Validation Accuracy:  84.4%,  Validation Loss: 0.270
Training Epoch 20 --- Training Accuracy: 100.0%, Validation Accuracy:  90.6%,  Validation Loss: 0.174
Training Epoch 21 --- Training Accuracy: 100.0%, Validation Accuracy:  93.8%,  Validation Loss: 0.199
Training Epoch 22 --- Training Accuracy: 100.0%, Validation Accuracy:  96.9%,  Validation Loss: 0.154
Training Epoch 23 --- Training Accuracy: 100.0%, Validation Accuracy:  93.8%,  Validation Loss: 0.183
Training Epoch 24 --- Training Accuracy: 100.0%, Validation Accuracy:  87.5%,  Validation Loss: 0.240
Training Epoch 25 --- Training Accuracy: 100.0%, Validation Accuracy:  93.8%,  Validation Loss: 0.122
Training Epoch 26 --- Training Accuracy: 100.0%, Validation Accuracy:  93.8%,  Validation Loss: 0.160
Training Epoch 27 --- Training Accuracy: 100.0%, Validation Accuracy:  93.8%,  Validation Loss: 0.150
Training Epoch 28 --- Training Accuracy: 100.0%, Validation Accuracy:  93.8%,  Validation Loss: 0.166
```

Εικόνα 4-18: Βήματα εκπαίδευσης του αλγορίθμου

```
Training Epoch 108 --- Training Accuracy: 100.0%, Validation Accuracy:  93.8%,  Validation Loss: 0.163
Training Epoch 109 --- Training Accuracy: 100.0%, Validation Accuracy:  87.5%,  Validation Loss: 0.435
Training Epoch 110 --- Training Accuracy: 100.0%, Validation Accuracy:  96.9%,  Validation Loss: 0.073
Training Epoch 111 --- Training Accuracy: 100.0%, Validation Accuracy:  96.9%,  Validation Loss: 0.253
Training Epoch 112 --- Training Accuracy: 100.0%, Validation Accuracy:  90.6%,  Validation Loss: 0.173
Training Epoch 113 --- Training Accuracy: 100.0%, Validation Accuracy:  93.8%,  Validation Loss: 0.176
Training Epoch 114 --- Training Accuracy: 100.0%, Validation Accuracy:  87.5%,  Validation Loss: 0.443
Training Epoch 115 --- Training Accuracy: 100.0%, Validation Accuracy:  96.9%,  Validation Loss: 0.084
Training Epoch 116 --- Training Accuracy: 100.0%, Validation Accuracy:  96.9%,  Validation Loss: 0.256
Training Epoch 117 --- Training Accuracy: 100.0%, Validation Accuracy:  87.5%,  Validation Loss: 0.172
Training Epoch 118 --- Training Accuracy: 100.0%, Validation Accuracy:  93.8%,  Validation Loss: 0.198
Training Epoch 119 --- Training Accuracy: 100.0%, Validation Accuracy:  84.4%,  Validation Loss: 0.455
Training Epoch 120 --- Training Accuracy: 100.0%, Validation Accuracy:  96.9%,  Validation Loss: 0.097
Training Epoch 121 --- Training Accuracy: 100.0%, Validation Accuracy:  96.9%,  Validation Loss: 0.264
Training Epoch 122 --- Training Accuracy: 100.0%, Validation Accuracy:  87.5%,  Validation Loss: 0.174
Training Epoch 123 --- Training Accuracy: 100.0%, Validation Accuracy:  93.8%,  Validation Loss: 0.233
Training Epoch 124 --- Training Accuracy: 100.0%, Validation Accuracy:  84.4%,  Validation Loss: 0.473
Training Epoch 125 --- Training Accuracy: 100.0%, Validation Accuracy:  96.9%,  Validation Loss: 0.115
Training Epoch 126 --- Training Accuracy: 100.0%, Validation Accuracy:  96.9%,  Validation Loss: 0.270
Training Epoch 127 --- Training Accuracy: 100.0%, Validation Accuracy:  87.5%,  Validation Loss: 0.184
Training Epoch 128 --- Training Accuracy: 100.0%, Validation Accuracy:  93.8%,  Validation Loss: 0.278
Training Epoch 129 --- Training Accuracy: 100.0%, Validation Accuracy:  84.4%,  Validation Loss: 0.504
Training Epoch 130 --- Training Accuracy: 100.0%, Validation Accuracy:  96.9%,  Validation Loss: 0.136
Training Epoch 131 --- Training Accuracy: 100.0%, Validation Accuracy:  93.8%,  Validation Loss: 0.281
Training Epoch 132 --- Training Accuracy: 100.0%, Validation Accuracy:  90.6%,  Validation Loss: 0.198
Training Epoch 133 --- Training Accuracy: 100.0%, Validation Accuracy:  90.6%,  Validation Loss: 0.340
Training Epoch 134 --- Training Accuracy: 100.0%, Validation Accuracy:  81.2%,  Validation Loss: 0.545
Training Epoch 135 --- Training Accuracy: 100.0%, Validation Accuracy:  96.9%,  Validation Loss: 0.161
Training Epoch 136 --- Training Accuracy: 100.0%, Validation Accuracy:  93.8%,  Validation Loss: 0.294
Training Epoch 137 --- Training Accuracy: 100.0%, Validation Accuracy:  87.5%,  Validation Loss: 0.226
Training Epoch 138 --- Training Accuracy: 100.0%, Validation Accuracy:  90.6%,  Validation Loss: 0.446
Training Epoch 139 --- Training Accuracy: 100.0%, Validation Accuracy:  81.2%,  Validation Loss: 0.620
Training Epoch 140 --- Training Accuracy: 100.0%, Validation Accuracy:  96.9%,  Validation Loss: 0.189
Training Epoch 141 --- Training Accuracy: 100.0%, Validation Accuracy:  93.8%,  Validation Loss: 0.316
Training Epoch 142 --- Training Accuracy: 100.0%, Validation Accuracy:  87.5%,  Validation Loss: 0.252
Training Epoch 143 --- Training Accuracy: 100.0%, Validation Accuracy:  90.6%,  Validation Loss: 0.582
bill@bill-VirtualBox:~/Επιφάνεια εργασίας/InsiderThreats$
```

Εικόνα 4-19: Βήματα εκπαίδευσης του αλγορίθμου

Αναλυτικά τα αποτελέσματα της εκπαίδευσης του αλγορίθμου, παρουσιάζονται στον πίνακα που ακολουθεί:

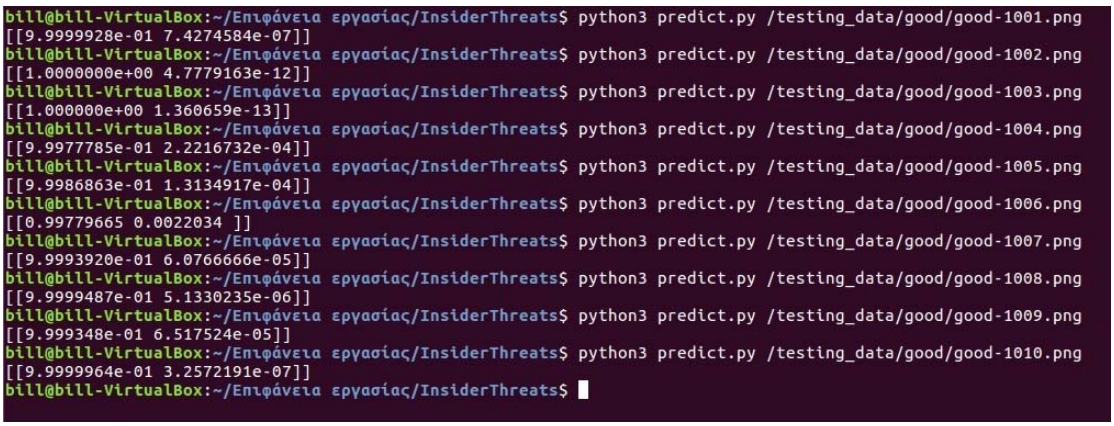
Αριθμός Βήματος Εκπαίδευσης (Training Epoch)	Training Accuracy	Validation Accuracy	Validation Loss (Cost)
1	50.0%	62.5%	0.686
2	50.0%	43.8%	0.708
3	50.0%	53.1%	0.676
4	78.1%	62.5%	0.654
5	81.2%	68.8%	0.618
6	78.1%	71.9%	0.623
7	87.5%	90.6%	0.381
8	84.4%	78.1%	0.528
9	81.2%	75.0%	0.528
10	84.4%	90.6%	0.384
11	87.5%	78.1%	0.557
12	84.4%	87.5%	0.269
13	84.4%	81.2%	0.394
14	87.5%	81.2%	0.350
15	90.6%	90.6%	0.263
16	90.6%	84.4%	0.324
17	96.9%	93.8%	0.190
18	96.9%	90.6%	0.245
19	100%	84.4%	0.270
20	100%	90.6%	0.174
21	100%	93.8%	0.199
22	100%	96.9%	0.154
23	100%	93.8%	0.183
24	100%	87.5%	0.240
25	100%	93.8%	0.122
26	100%	93.8%	0.160
27	100%	93.8%	0.150
28	100%	93.8%	0.166
29	100%	87.5%	0.237
30	100%	93.8%	0.095
31	100%	93.8%	0.148
32	100%	93.8%	0.151
33	100%	93.8%	0.154
34	100%	87.5%	0.242
35	100%	100%	0.075
36	100%	93.8%	0.148
37	100%	93.8%	0.155
38	100%	90.6%	0.152
39	100%	87.5%	0.253
40	100%	100%	0.062
41	100%	93.8%	0.152
42	100%	93.8%	0.159
43	100%	90.6%	0.148
44	100%	87.5%	0.261
45	100%	100%	0.055
46	100%	93.8%	0.157
47	100%	93.8%	0.165
48	100%	90.6%	0.148
49	100%	87.5%	0.272
50	100%	100%	0.050
51	100%	93.8%	0.164

52	100%	93.8%	0.169
53	100%	93.8%	0.144
54	100%	87.5%	0.283
55	100%	96.9%	0.048
56	100%	93.8%	0.170
57	100%	93.8%	0.174
58	100%	93.8%	0.141
59	100%	84.4%	0.294
60	100%	96.9%	0.047
61	100%	93.8%	0.175
62	100%	90.6%	0.180
63	100%	96.9%	0.136
64	100%	84.4%	0.310
65	100%	96.9%	0.047
66	100%	93.8%	0.186
67	100%	90.6%	0.184
68	100%	96.9%	0.135
69	100%	84.4%	0.329
70	100%	96.9%	0.047
71	100%	93.8%	0.195
72	100%	93.8%	0.187
73	100%	96.9%	0.133
74	100%	84.4%	0.350
75	100%	96.9%	0.047
76	100%	93.8%	0.203
77	100%	93.8%	0.191
78	100%	93.8%	0.134
79	100%	84.4%	0.370
80	100%	96.9%	0.048
81	100%	93.8%	0.210
82	100%	93.8%	0.191
83	100%	93.8%	0.138
84	100%	84.4%	0.388
85	100%	96.9%	0.048
86	100%	93.8%	0.220
87	100%	93.8%	0.190
88	100%	93.8%	0.141
89	100%	84.4%	0.403
90	100%	96.9%	0.051
91	100%	96.9%	0.229
92	100%	90.6%	0.187
93	100%	93.8%	0.144
94	100%	84.4%	0.416
95	100%	96.9%	0.053
96	100%	96.9%	0.236
97	100%	90.6%	0.184
98	100%	90.6%	0.149
99	100%	84.4%	0.425
100	100%	96.9%	0.057
101	100%	96.9%	0.242
102	100%	90.6%	0.181
103	100%	90.6%	0.154
104	100%	84.4%	0.431
105	100%	96.9%	0.064
106	100%	96.9%	0.249
107	100%	90.6%	0.176
108	100%	93.8%	0.163
109	100%	87.5%	0.435
110	100%	96.9%	0.073

111	100%	96.9%	0.253
112	100%	90.6%	0.173
113	100%	93.8%	0.176
114	100%	87.5%	0.443
115	100%	96.9%	0.084
116	100%	96.9%	0.256
117	100%	87.5%	0.172
118	100%	93.8%	0.198
119	100%	84.4%	0.455
120	100%	96.9%	0.097
121	100%	96.9%	0.264
122	100%	87.5%	0.174
123	100%	93.8%	0.233
124	100%	84.4%	0.473
125	100%	96.9%	0.115
126	100%	96.9%	0.270
127	100%	87.5%	0.184
128	100%	93.8%	0.278
129	100%	84.4%	0.504
130	100%	96.9%	0.136
131	100%	93.8%	0.281
132	100%	90.6%	0.198
133	100%	90.6%	0.340
134	100%	81.2%	0.545
135	100%	96.9%	0.161
136	100%	93.8%	0.294
137	100%	87.5%	0.226
138	100%	90.6%	0.446
139	100%	81.2%	0.620
140	100%	96.9%	0.189
141	100%	93.8%	0.316
142	100%	87.5%	0.252
143	100%	90.6%	0.582

Πίνακας 4-1: Συγκεντρωτικά αποτελέσματα εκπαίδευσης αλγορίθμου

Στη συνέχεια, εισήχθησαν είκοσι (20) εικόνες μία προς μία, δέκα (10) εκ των οποίων περιείχαν - κατά την κρίση μας - «φυσιολογική» και αντίστοιχος αριθμός «κακόβουλη» δραστηριότητα. Η πρόβλεψη των αποτελεσμάτων είχε απόλυτη επιτυχία, με ποσοστό, που άγγιξε το 100% . Ακολουθούν οι δύο (2) εικόνες, που περιέχουν τις προβλέψεις για την «φυσιολογική» και την «κακόβουλη» δραστηριότητα από τη δοκιμή του πειράματος:



Εικόνα 4-20: Δοκιμή της διαδικασίας πρόβλεψης του αλγορίθμου για φυσιολογική δραστηριότητα

Εικόνα 4-21: Δοκιμή της διαδικασίας πρόβλεψης του αλγορίθμου για κακόβουλη δραστηριότητα

Για την παρακολούθηση της πορείας εκπαίδευσης του αλγορίθμου δημιουργήθηκαν γραφικές παραστάσεις, οι οποίες απεικονίζουν την ακρίβεια εκπαίδευσης (training accuracy), την ακρίβεια επικύρωσης (validation accuracy) και το κόστος (cost). Οι απεικονίσεις υλοποιήθηκαν με το πρόγραμμα Tensorboard, που αποτελεί πακέτο του προγράμματος Tensorflow. Ειδικότερα, το TensorBoard αποτελεί ένα πακέτο εφαρμογών ιστού (web application) για την επιθεώρηση και κατανόηση των γραφημάτων μέσω του TensorFlow. Υποστηρίζει οπτικοποιήσεις, κλιμακωτές εικόνες, ήχο, ιστογράμματα και γραφήματα. Εξαιτίας του γεγονότος ότι οι υπολογισμοί, που χρησιμοποιούνται στο TensorFlow για την εκπαίδευση ενός νευρωνικού δικτύου, μπορεί να είναι αρκετά περίπλοκοι, το TensorBoard συμβάλλει στην ευκολότερη κατανόηση των προγραμμάτων του, τον εντοπισμό σφαλμάτων και τη βελτιστοποίηση τους. Εκτός αυτού, αποτελεί ένα μέσο τόσο για την παρακολούθηση και εξέλιξη των

ενδιάμεσων σταδίων εκπαίδευσης του αλγορίθμου, όσο και για την ομαλή εξέλιξη των σταδίων της εκπαίδευσης.

Για την υλοποίηση των γραφημάτων ο ήδη υπάρχων κώδικας συμπληρώθηκε με τις κατωτέρω εντολές:

Αρχικά, με την εντολή **tf.scalar_summary** δημιουργήθηκαν οι εικόνες των γραφημάτων και ορίστηκαν οι τίτλοι τους δηλώνοντας παράλληλα και από ποιές μεταβλητές θα εισαχθούν τα δεδομένα σε αυτά:

```
# create a summary for our cost and accuracy

tf.scalar_summary("Cost", cost)

training_acc_summary = tf.scalar_summary("Training Accuracy", accuracy)

validation_acc_summary = tf.scalar_summary("Validation Accuracy", accuracy)
```

Στη συνέχεια, όλες οι λειτουργίες των γραφημάτων αθροίστηκαν σε μία ενιαία λειτουργία με την εντολή **tf.merge_all_summaries**():

```
# merge all summaries into a single "operation" which we can execute in a session

merged_summary_op = tf.merge_all_summaries()
```

Έπειτα, ορίστηκε η μεταβλητή, η οποία συγκέντρωσε όλα τα δεδομένα όλων των γραφικών παραστάσεων καθώς και ο φάκελος στον οποίο θα εγγράφονταν τα αρχεία καταγραφής (logs):

```
# Set the logs writer to the folder /Tensorboard/logs

summary_writer = tf.train.SummaryWriter('Tensorboard/logs', graph=tf.get_default_graph())
```

Τέλος, μέσα στο βρόγχο (loop) εκπαίδευσης του προγράμματος εισήχθησαν οι παράμετροι και οι μεταβλητές, που θα συγκέντρωναν όλα τα δεδομένα που θα παρουσιάζονταν στις γραφικές παραστάσεις:

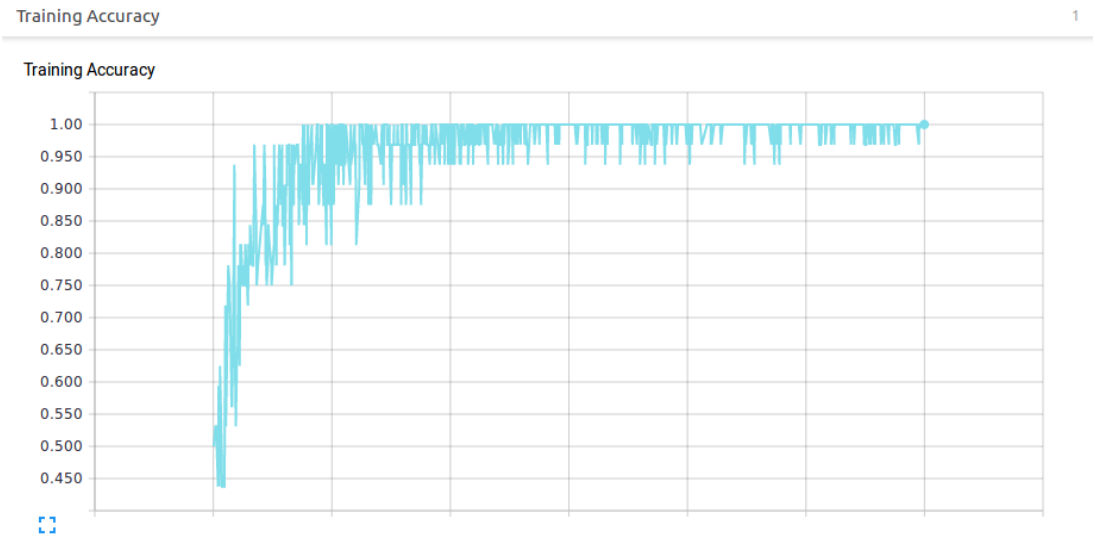
```
# Write logs for each iteration

summary_str = session.run(merged_summary_op, feed_dict={x: x_batch, y_true: y_true_batch})

summary_writer.add_summary(summary_str, i)

training_summary_str = session.run(training_acc_summary, feed_dict={x: x_batch, y_true: y_true_batch})
```

summary_writer.add_summary(training_summary_str, i)

validation_summary_str = session.run(validation_acc_summary, feed_dict={x: x_valid_batch, y_true: y_valid_batch})

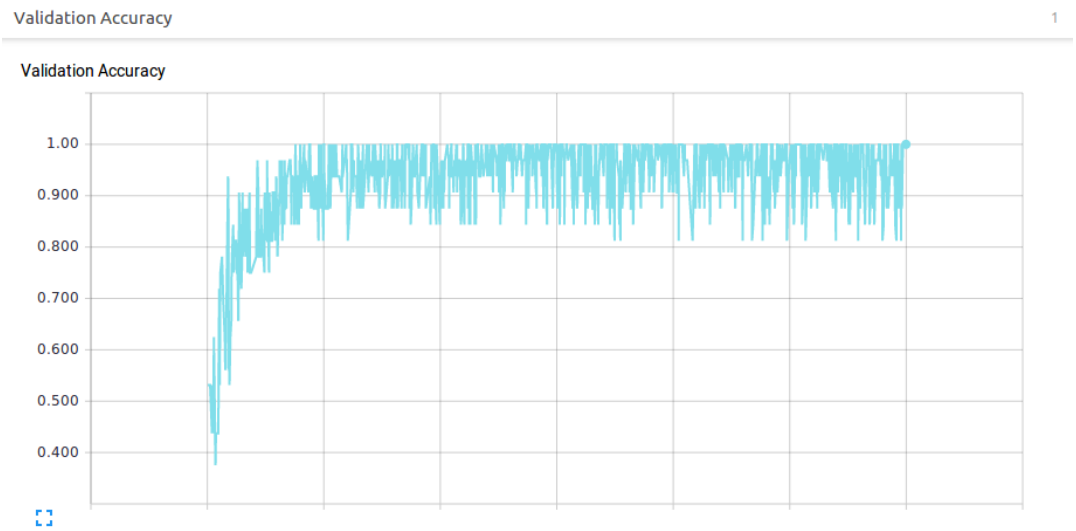
summary_writer.add_summary(validation_summary_str, i)

Κατωτέρω παρουσιάζονται αναλυτικά οι τρεις (3) γραφικές παραστάσεις (training accuracy, validation accuracy, cost) και το γράφημα υλοποίησης της εκπαίδευσης του αλγορίθμου CNN:



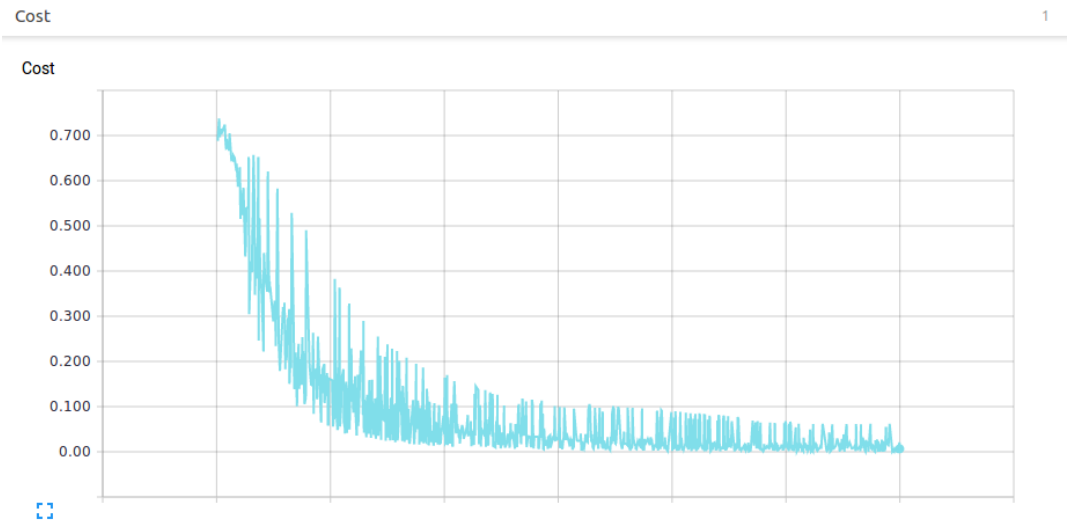
Γραφική παράσταση 4-1: Γραφική αναπαράσταση Training Accuracy

Σύμφωνα με τη γραφική παράσταση η ακρίβεια εκπαίδευσης (training accuracy) εκκινεί από μία τιμή κοντά στο 0,450 δηλ. 45% και έχει αυξανόμενη τιμή για να καταλήξει σε ποσοστό, που προσεγγίζει το 100%.



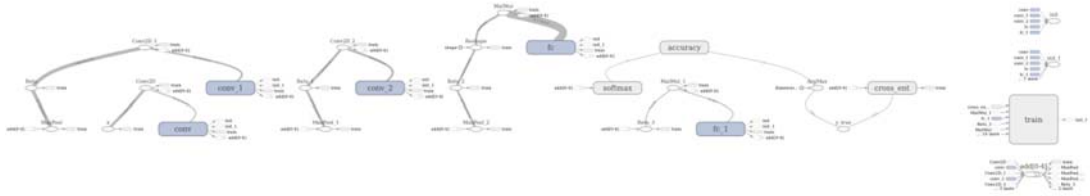
Γραφική παράσταση 4-2: Γραφική αναπαράσταση Validation Accuracy

Σύμφωνα με τη γραφική παράσταση η ακρίβεια επικύρωσης (validation accuracy) εκκινεί από μία τιμή κοντά στο 0,400 δηλ. 40% και έχει αυξανόμενη τιμή για να καταλήξει σε ποσοστό, που προσεγγίζει το 90%.



Γραφική παράσταση 4-3: Γραφική αναπαράσταση Cost

Σύμφωνα με τη γραφική παράσταση το κόστος (cost) εκκινεί από μία τιμή κοντά στο 0,700 δηλ. 70% και έχει μειούμενη τιμή για να καταλήξει σε τιμή, που προσεγγίζει το 0.



Εικόνα 4-22: Block διάγραμμα του CNN δικτύου που υλοποιήθηκε

4.6 Σύνοψη Κεφαλαίου

Στο κεφάλαιο αυτό παρουσιάστηκε και αναλύθηκε η μέθοδος πρόβλεψης και ανίχνευσης της συμπεριφοράς κακόβουλων χρηστών με τη χρήση CNN, που υλοποιήθηκε με το πρόγραμμα Tensorflow της Google. Η ανάλυση έγινε με την εξαγωγή δεδομένων από συγκεκριμένο dataset με τη βοήθεια προγραμμάτων (scripts) σε Java, τα οποία οδήγησαν στην οπτικοποίησή τους με τη χρήση της βιβλιοθήκης D3.js σε Javascript. Επίσης, ταυτόχρονα με την καταγραφή του πειράματος έλαβε χώρα και η ανάλυση των ευρημάτων και αποτελεσμάτων των διαφόρων σταδίων αυτού.

Κεφάλαιο 5
Επίλογος

5.1 Συμπεράσματα

Στόχος της παρούσας μεταπτυχιακής διατριβής ήταν η απάντηση στο ερώτημα εάν μπορεί η Τεχνητή Νοημοσύνη να χρησιμοποιηθεί επιτυχώς για την ανίχνευση κακόβουλης δραστηριότητας. Η απάντηση στο ερώτημα αυτό διήλθε από τρία (3) στάδια: α) της συλλογής, επεξεργασίας και ταξινόμησης των δεδομένων των εξεταζόμενων χρηστών, β) της οπτικοποίησης των εξαγόμενων δεδομένων και γ) της χρήσης της Τεχνητής Νοημοσύνης, δια της εκπαίδευσης ενός αλγορίθμου CNN ώστε να κατηγοριοποιεί την συμπεριφορά σε κακόβουλη ή φυσιολογική. Για την εκπαίδευση του αλγορίθμου χρησιμοποιήθηκε ένα μεγάλο μέρος από τις δημιουργηθείσες εικόνες του δευτέρου σταδίου, τις οποίες είχαμε προηγουμένως κατηγοριοποιήσει με συγκεκριμένα κριτήρια, που αναφέρθηκαν, σε εικόνες που περιείχαν κακόβουλη δραστηριότητα και σε εικόνες, που περιείχαν φυσιολογική δραστηριότητα. Με τον τρόπο αυτό ο αλγόριθμος «έμαθε» να ξεχωρίζει την κακόβουλη από τη φυσιολογική δραστηριότητα. Μετά την εκπαίδευση έγινε εισαγωγή ορισμένων εικόνων, που περιείχαν και κακόβουλη και φυσιολογική δραστηριότητα για να τις αναγνωρίσει και να τις κατηγοριοποιήσει ο αλγόριθμος.

Θα πρέπει να σημειωθεί ότι, όπως προέκυψε από το πείραμα, ο τρόπος που χαρακτηρίζεται μια συμπεριφορά ενός χρήστη ως κακόβουλη εξαρτάται και ποικίλλει ανάλογα με την Πολιτική Ασφαλείας, που υιοθετεί η εκάστοτε Εταιρεία ή Οργανισμός για να προστατεύσει το πληροφοριακό της/του σύστημα. Στην συγκεκριμένη περίπτωση π.χ θεωρήσαμε ότι η επίσκεψη σε ιστοσελίδες (sites) κοινωνικών δικτύων ή η επίσκεψη σε ιστοσελίδες (sites) ανεύρεσης εργασίας συνιστούν κακόβουλη δραστηριότητα. Πέραν αυτού, σημαντικό ρόλο στον χαρακτηρισμό της δραστηριότητας ενός χρήστη διαδραματίζει και η θέση, που κατέχει στην Εταιρεία/Οργανισμό. Στην υπό κρίση περίπτωση π.χ χαρακτηρίστηκε ως κακόβουλη δραστηριότητα η δραστηριότητα του χρήστη PLJ1771, ο οποίος - σύμφωνα με τα δεδομένα της εργασίας - κατέχει θέση IT Admin, οπότε ενδεχομένως η θέση του δικαιολογεί – έως ένα σημείο – την χαρακτηρισθείσα ως κακόβουλη δραστηριότητά του.

Περαιτέρω, από την μελέτη της οπτικοποίησης (visualization) των δεδομένων προέκυψε ότι η συμπεριφορά ενός χρήστη δεν χαρακτηρίζεται ως κακόβουλη, καθ' όλο το χρονικό διάστημα της περιόδου, που ερευνήθηκε. Υπήρχαν περίοδοι, που η δραστηριότητά του συμβάδιζε με τους κανόνες, που διέπουν την ορθή και νόμιμη χρήση του πληροφοριακού συστήματος της Εταιρείας για την οποία εργάζονταν.

Τελικώς, η παρούσα μεταπτυχιακή διατριβή απέδειξε ότι πράγματι με την εν λόγω μεθοδολογία, που χρησιμοποιήθηκε επετεύχθη η αναγνώριση της κακόβουλης δραστηριότητας των χρηστών του πληροφοριακού συστήματος. Η πρόβλεψη των αποτελεσμάτων είχε απόλυτη επιτυχία, με ποσοστό, που άγγιξε το 100%, όπως απεικονίστηκε στις αντίστοιχες εικόνες.

5.2 Σύγκριση μεθοδολογιών

Αντίθετα από τις μελέτες που παρουσιάστηκαν, η τεχνική που χρησιμοποιήθηκε στην παρούσα μεταπτυχιακή διατριβή βασίστηκε στην εξόρυξη δεδομένων, στην οπτικοποίησή τους και τελικά στην χρήση αλγορίθμου CNN μέσω Μηχανικής Μάθησης (Machine Learning), για την κατηγοριοποίηση της δραστηριότητας των υπό εξέταση χρηστών σε κακόβουλη και φυσιολογική. Επίσης, οι τιμές της ακρίβειας εκπαίδευσης και επικύρωσης (training και validation accuracy), προσέγγισαν ποσοστά 100% και 90% αντίστοιχα, σε αντίθεση με τις άλλες μεθοδολογίες των μελετών που αναφέρθηκαν, τα αποτελέσματα των οποίων παρουσίασαν χαμηλότερα ποσοστά. Εκτός αυτών, στην εν λόγω μεταπτυχιακή διατριβή χρησιμοποιήθηκε ένας αρκετά μεγάλος όγκος από πραγματικά διαθέσιμα δεδομένα για την κατασκευή και δοκιμή του μοντέλου και του μηχανισμού ανίχνευσης εσωτερικών απειλών. Περαιτέρω, τα δεδομένα αυτά κάλυπταν ένα μεγάλο χρονικό διάστημα ενάμιση έτους (01/2010 – 05/2011).

Επιπλέον, σε αντίθεση με άλλες μεθοδολογίες, οι οποίες χρησιμοποίησαν παραλλαγές νευρωνικών δικτύων και επαναλαμβανόμενα νευρωνικά δίκτυα ώστε να αναγνωρίζουν την συμπεριφορά του χρήστη σε πραγματικούς χρόνους προσαρμόζοντας το μοντέλο τους στα μεταβαλλόμενα πρότυπα των δεδομένων, το μοντέλο μας δεν λειτούργησε σε πραγματικό χρόνο ώστε να αναγνωρίζει την κακόβουλη συμπεριφορά τη στιγμή, που συμβαίνει και να προβλέπει αυτήν σε πραγματικό χρόνο αλλά κατηγοριοποίησε όλη τη δραστηριότητα ενός χρήστη εντός του εξεταζόμενου χρονικού διαστήματος. Με τον τρόπο αυτό αποφεύχθηκε ο κίνδυνος της μίμησης, που ελλοχεύει στις ως άνω μεθοδολογίες, μίας φυσιολογικής συμπεριφοράς από έναν κακόβουλο χρήστη διότι δεν εξετάζεται η συμπεριφορά του μόνο μία συγκεκριμένη χρονική στιγμή αλλά εντός μίας μεγάλης χρονικής περιόδου.

Επιπρόσθετα, αντίθετα από μελέτες που αναζητούν δραστηριότητες που παρουσιάζουν ομοιότητες με τις φυσιολογικές συναλλαγές δεδομένων αλλά στην πραγματικότητα είναι δομικά

διαφορετικές από αυτές και χρησιμοποιούν γραφήματα και θεωρητικές προσεγγίσεις ή στατιστική επεξεργασία δεδομένων για την εξαγωγή συμπερασμάτων, η μέθοδος που χρησιμοποιήθηκε πλεονεκτεί στο ότι ανέπτυξε οπτικοποίηση (visualization) της δραστηριότητας του χρήστη ανά μήνα και ανά εβδομάδα καθιστώντας με τον τρόπο αυτό αμεσότερη την μοντελοποίηση της κακόβουλης ή μη δραστηριότητας.

Επίσης, άλλοι μελετητές χρησιμοποίησαν το ψυχολογικό προφίλ PP, καθώς και προφίλ που βασίζονται σε ρόλους και πρότειναν προσεγγίσεις που συνδύαζαν την ανίχνευση δομικών ανωμαλιών SA από κοινωνικά δίκτυα και δίκτυα πληροφόρησης, προκειμένου να παρατηρήσουν και να μετρήσουν τον τρόπο που οι χρήστες αποκλίνουν από τις παρατηρούμενες συμπεριφορές τους και εάν αυτό τελικά μπορεί να αποτελέσει πιθανή απειλή. Στην δική μας περίπτωση μελετήθηκε αποκλειστικά η δραστηριότητα του χρήστη, οι πράξεις του και οι ενέργειές του χωρίς να εξεταστεί το ψυχολογικό ή άλλο προφίλ δηλαδή η συμπεριφορά του με βάση τα χαρακτηριστικά, που θα πρέπει να περιλαμβάνει αλλά μόνο η συνολική συμπεριφορά του στο δεδομένο υπό εξέταση χρονικό διάστημα.

Τέλος, σε σύγκριση με άλλες μελέτες, στις οποίες τα δεδομένα εξετάστηκαν συνολικά, στην παρούσα μεταπτυχιακή διατριβή δόθηκε έμφαση και μεγάλη σημασία στην ταξινόμηση των δεδομένων. Για το λόγο αυτό αρχικά έλαβε χώρα ταξινόμηση των δεδομένων και κατηγοριοποίησή τους ανά χρήστη προκειμένου να μειωθεί ο όγκος ανάλυσής τους, γεγονός που συνέβαλε τόσο στην καλύτερη ανάλυσή τους, όσο και στην βελτιστοποίηση του χρόνου επεξεργασίας τους.

Ακολουθεί ένας συνοπτικός πίνακας, ο οποίος παρουσιάζει τα αποτελέσματα ορισμένων μεθόδων αναγνώρισης εσωτερικών, όσον αφορά τόσον στη τιμή της ακρίβειας επικύρωσης (validation accuracy), όσο και σε ορισμένα στοιχεία σύγκρισής τους, σε σχέση με την παρούσα διπλωματική διατριβή:

Συγγραφέας/Μελέτη	Accuracy	Σύγκριση
Our Method	90%	Στη μέθοδό μας χρησιμοποιήθηκε μηχανική μάθηση (machine learning) και εκπαίδευση του αλγορίθμου CNN για την υλοποίηση της κατηγοριοπόιησης της συμπεριφοράς του χρήστη σε φυσιολογική και κακόβουλη.
W. Eberle, L. Holder, Detecting Insider Threats Using a Graph-Based Approach	95%	Στη συγκεκριμένη μέθοδο χρησιμοποιήθηκε μια γραφική θεωρητική προσέγγιση για την ανίχνευση κακόβουλης συμπεριφοράς χρηστών σε ένα Πληροφοριακό Σύστημα.
O. Brdiczka, J. Liu, B. Price, J. Shen, A. Patil, R. Chow, E. Bart, N. Ducheneaut, Proactive Insider Threat Detection through Graph Learning and Psychological Context	Server Eitrigg: 82.74% Server Cenarion Circle: 89.09% Server Bleeding Hollow: 79.84%	Η συγκεκριμένη μελέτη, προτείνει μια προσέγγιση που συνδυάζει την ανίχνευση δομικών ανωμαλιών (Structural Anomaly Detection - SA) από κοινωνικά δίκτυα και δίκτυα πληροφόρησης καθώς και του ψυχολογικού προφίλ (Psychological Profiling - PP) των ατόμων.
W. T. Young, H. G. Goldberg, A. Memory, J. Sartain T. Senator, Use of Domain Knowledge to Detect Insider Threats in Computer Activities	Indicators: 87.4% Anomalies: 97.9% Scenarios: 80.6%	Η συγκεκριμένη μελέτη, επικεντρώνεται στη γνώση του πεδίου του Πληροφοριακού Συστήματος (domain knowledge) για να ανιχνεύσει κακόβουλες συμπεριφορές μέσω της χρήσης συγκεκριμένων αλγορίθμων ανίχνευσης δομικών ανωμαλιών (SA).
J. Breier, J. Branisova, A Dynamic Rule Creation Based Anomaly Detection Method for Identifying Security Breaches in Log Records		Το ποσοστό λάθους της μεθόδου που χρησιμοποιήθηκε, ήταν μικρότερο του 10%. Στη μέθοδο που χρησιμοποιήσαμε, το ποσοστό λάθους προσεγγίζει το μηδέν (0).
Ph. A. Legg, O. Buckley, M. Goldsmith, S. Creese, Caught in the Act of an Insider Attack: Detection and Assessment of Insider Threat		Η μέθοδος που χρησιμοποιήθηκε, βασίστηκε στη συλλογή δεδομένων καταγραφής, μέσω της κατασκευής προφίλ για κάθε χρήστη και της ιδιότητάς του. Στη μέθοδο που χρησιμοποιήσαμε, η ιδιότητα του χρήστη δεν αξιολογήθηκε.

Πίνακας 4-2: Συγκριτική αποτίμηση της μεθόδου που χρησιμοποιήθηκε με άλλες

5.3 Προοπτικές

Ένας σημαντικός παράγοντας ο οποίος θα συνέβαλε στην βελτιστοποίηση της παρούσας μεταπτυχιακής διατριβής είναι καταρχάς η γνώση της πολιτικής ασφαλείας του υπό εξέταση πληροφοριακού συστήματος. Με τον όρο «Πολιτική Ασφαλείας» νοείται (Security Policy) το σύνολο των κανόνων οι οποίοι περιγράφουν τους στόχους της ασφάλειας και τις αντίστοιχες διαδικασίες που πρέπει να ακολουθούνται ώστε να επιτευχθούν αυτοί οι στόχοι. Η Πολιτική Ασφαλείας καθορίζει τη δέσμευση της Διοίκησης και την προσέγγιση ενός οργανισμού ή μιας επιχείρησης αναφορικά με την ασφάλεια των πληροφοριακών συστημάτων και δικτύων και την προστασία προσωπικών δεδομένων που τηρεί ο υπεύθυνος επεξεργασίας. Γνωρίζοντας την συγκεκριμένη πολιτική ασφαλείας είναι δυνατό να βελτιωθούν οι κανόνες ανεύρεσης των ύποπτων συμπεριφορών και με αυτό τον τρόπο να υπάρξει ακριβέστερη ανάλυση των δεδομένων, καθ' όσον όπως είναι γνωστό η πολιτική ασφαλείας δεν είναι η ίδια για όλες τις Εταιρείες ή τους Οργανισμούς αλλά ποικίλλει και διαφοροποιείται ανάλογα με τις ανάγκες κάθε μίας εξ αυτών. Περαιτέρω, η εφαρμογή της μελέτης για μεγαλύτερο χρονικό διάστημα καθώς και η ανάλυση των αποτελεσμάτων για μεγαλύτερο αριθμό χρηστών θα αποτελούσαν επίσης βοηθητικούς παράγοντες για την βελτίωση της εν λόγω μελέτης λόγω του μεγαλύτερου όγκου των δεδομένων. Τέλος, η δοκιμή τόσον με άλλα σχέδια εικόνων, όπως τρισδιάστατες εικόνες ή δυναμικά σχήματα, όσο και με άλλες βιβλιοθήκες οπτικοποίησης θα συνέβαλε στην βελτίωση του πειράματος καθώς από την εξέταση και άλλων τρόπων και μορφών σχημάτων και την συγκριτική αποτίμηση αυτών θα εξάγονταν πιο ασφαλή συμπεράσματα.

Βιβλιογραφία

Ελληνική

[01] Γ. Ηλιάδης, Κακόβουλο Λογισμικό στο Σ. Κάτσικας, Δ. Γκρίτζαλης και Σ. Γκρίτζαλης, (Επιστ. Επιμ.), Ασφάλεια Πληροφοριακών Συστημάτων, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 2004, σελ. 235 επ.

[02] Μ. Καρύδα, Πολιτικές Ασφάλειας Πληροφοριακών Συστημάτων, στο Σ. Κάτσικας, Δ. Γκρίτζαλης και Σ. Γκρίτζαλης, (Επιστ. Επιμ.), *Ασφάλεια Πληροφοριακών Συστημάτων*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 2004, σελ. 378 επ.

[03] Σ. Κάτσικας, Διαχείριση της ασφάλειας πληροφοριών, Εκδόσεις Πεδίο, Αθήνα, 2014.

[04] Ε. Κεραυνού, Τεχνητή Νοημοσύνη και έμπειρα συστήματα, τόμος α΄, Πάτρα 2000.

[05] Γ. Κρασαδάκης, Τεχνητή Νοημοσύνη: Τι είναι και πώς αλλάζει δραματικά τον κόσμο μας, προσβάσιμο από το: http://www.naftemporiki.gr/story/1309187/texniti-noimosuni-ti-einai-kai-pos-allazei-dramatika-ton-kosmo-mas.

[06] Ε. Μάγκος, Ασφάλεια υπολογιστών και προστασία δεδομένων, σημειώσεις μαθήματος δ΄ εξαμήνου Τμήματος Πληροφορικής Ιονίου Πανεπιστημίου, Κέρκυρα, 2007.

[07] Γ. Πάγκαλος – Ι. Μαυρίδης, Ασφάλεια Πληροφοριακών Συστημάτων και Δικτύων, Εκδόσεις ΑΝΙΚΟΥΛΑ, Θεσσαλονίκη, 2002.

[08] Α. Πλέρου, Τεχνητά νευρωνικά δίκτυα προσομοίωσης του ανθρωπίνου εγκεφάλου. Ανοικτή Εκπαίδευση: το περιοδικό για την Ανοικτή και εξ Αποστάσεως Εκπαίδευση και την Εκπαιδευτική Τεχνολογία, 2016, 8(1), σελ. 128-135, προσβάσιμο από το: https://ejournals.epublishing.ekt.gr/index.php/openjournal/article/view/9794/9923

[09] Χ. Σχίζας, Κ. Νεοκλέους, Τεχνητή Νοημοσύνη και τεχνητά νευρωνικά δίκτυα, προσβάσιμο από το: http://www.pemptousia.gr/2017/08/texniti-noimosini-ke-texnita-nevronika-diktia/

Ξενόγλωσση

[10] O. Brdiczka, J. Liu, B. Price, J. Shen, A. Patil, R. Chow, E. Bart, N. Ducheneaut, Proactive Insider Threat Detection through Graph Learning and Psychological Context, προσβάσιμο από το: https://www.parc.com/content/attachments/proactive-insider-threat-detection.pdf.

[11] J. Breier, J. Branisova, Anomaly Detection from Log Files Using Data Mining Techniques, in: Lecture Notes in Electrical Engineering, Ιανουάριος 2015, προσβάσιμο από το: https://www.researchgate.net/publication/284244364_A_Dynamic_Rule_Creation_Based_Anomaly_Detection_Method_for_Identifying_Security_Breaches_in_Log_Records.

[12] J. Breier, J. Branisova, A Dynamic Rule Creation Based Anomaly Detection Method for Identifying Security Breaches in Log Records, προσβάσιμο από το: http://jbreier.com/files/papers/wps_2015.pdf.

[13] W. Eberle, L. Holder, Detecting Insider Threats Using a Graph-Based Approach, προσβάσιμο από το: https://pdfs.semanticscholar.org/482a/a61a5c417ce20a4bbf3e2a17d73f7f7afbb4.pdf.

[14] W. Eberle, L. Holder, Insider Threat Detection Using Graph-Based Approaches, in: Cybersecurity Applications & Technology Conference for Homeland Security, προσβάσιμο από το:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.720.639&rep=rep1&type=pdf.

[15] I. Graham, Τεχνητή Νοημοσύνη- στην αιχμή της επιστήμης, Εκδόσεις Σαββάλας, 2004.

[16] J. Haugeland, Τεχνητή Νοημοσύνη: Σχεδιάζοντας τη νόηση: από την υπολογιστική θεωρία στις σύγχρονες ευφυείς μηχανές, Εκδόσεις κάτοπτρο, Αθήνα, 2011.

[17] J. M. Kizza, Computer Communications and Networks, Guide to Computer Network Security, Springer, London, 2009.

[18] Ph. A. Legg, O. Buckley, M. Goldsmith, S. Creese, Caught in the Act of an Insider Attack: Detection and Assessment of Insider Threat in: IEEE International Symposium on

Technologies for Homeland Security, Waltham, USA, 14th-16th April 2015, προσβάσιμο από το: https://www.researchgate.net/publication/276976957_Caught_in_the_Act_of_an_Insider_Attack_Detection_and_Assessment_of_Insider_Threat.

[19] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997.

[20] M. Negnevitsky, Τεχνητή Νοημοσύνη, Αρχές και εφαρμογές για την ανάπτυξη συστημάτων με τεχνολογίες νοημοσύνης, Εκδόσεις Τζιόλα, Θεσσαλονίκη, 2018[3].

[21] A. Sanzgiri, D. Dasgupta, Classification of Insider Threat Detection Techniques, προσβάσιμο από το: http://ais.cs.memphis.edu/files/papers/a25-Sanzgiri.pdf.

[22] M. T. Simpson, K. Backman, J. E. Corley, Hands - on ethical hacking and network defense, 2013, Course Technology.

[23] W. Stallings, Cryptography and Network Security, Principles and Practice, 2011[5,] Pearson Education.

[24] W. Stallings, Data and Computer Communications, 2007[8,] Pearson Education.

[25] P. Trim, D. Upton, Cyber Security Culture, Counteracting Cyber Threats through Organizational Learning and Training, 2013.

[26] A. Tuor, S. Kaplan, B. Hutchinson, Deep Learning for Unsupervised Insider Threat Detection in Structured Cybersecurity Data Streams, The AAAI-17 Workshop on Artificial Intelligence for Cyber Security WS-17-04.

[27] W. T. Young, H. G. Goldberg, A. Memory, J. Sartain T. Senator, Use of Domain Knowledge to Detect Insider Threats in Computer Activities, IEEE Security and Privacy Workshops, 2013, προσβάσιμο από το: http://www.ieee-security.org/TC/SPW2013/papers/data/5017a060.pdf.

ΔΙΑΔΙΚΤΥΑΚΟΙ ΤΟΠΟΙ

[28] http://physics4u.gr/blog/2018/01/18/%CF%84%CE%B9-%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9-%CE%B7-%CF%84%CE%B5%CF%87%CE%BD%CE%B7%CF%84%CE%AE-%CE%BD%CE%BF%CE%B7%CE%BC%CE%BF%CF%83%CF%8D%CE%BD%CE%B7-%CE%BA%CE%B1%CE%B9-%CF%80%CF%89%CF%82-%CE%B8%CE%B1/.

[29] https://www.sei.cmu.edu/about/divisions/cert/index.cfm

[30] https://www.tensorflow.org/

[31] https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

[32] https://el.wikipedia.org/wiki/%CE%9D%CE%B5%CF%85%CF%81%CF%89%CE%BD%CE%B9%CE%BA%CF%8C_%CE%B4%CE%AF%CE%BA%CF%84%CF%85%CE%BF

Παράρτημα Α

Κώδικες

A.1 Κώδικας σε Java

A.1.1 Πρώτου βήματος διαδικασίας εξόρυξης δεδομένων

```
package Main;

import org.eclipse.swt.widgets.Display;

import org.eclipse.swt.widgets.FileDialog;

import org.eclipse.swt.widgets.Shell;

import org.eclipse.swt.widgets.TabFolder;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.Date;

import java.util.Scanner;

import java.util.regex.Pattern;

import org.eclipse.swt.SWT;

import org.eclipse.swt.custom.StyledText;

import org.eclipse.swt.widgets.TabItem;

import org.eclipse.swt.widgets.Group;

import org.eclipse.swt.widgets.Label;

import org.eclipse.swt.widgets.Text;

import org.eclipse.swt.widgets.Button;

import org.eclipse.swt.widgets.DirectoryDialog;

import org.eclipse.swt.events.MouseAdapter;

import org.eclipse.swt.events.MouseEvent;

public class StepOne {
```

```
protected Shell shell;

private Text sourceFileTxt;

private Text destFoldertxt;

private Text searchKeytxt;

StyledText statusTxt;

Label recordsTxt;

Button btnNextStep;

StepTwo st;

/**

 * Launch the application.

 * @param args

 */

public static void main(String[] args) {



try {

		StepOne window = new StepOne();

		window.open();

		} catch (Exception e) {

			e.printStackTrace();

		}

}



/**

 * Open the window.

 */

public void open() {

		st = new StepTwo();
```

```
		Display display = Display.getDefault();
		createContents();
		shell.open();
		shell.layout();
		while (!shell.isDisposed()) {
			if (!display.readAndDispatch()) {
			display.sleep();
			}
		}
	}
	/**
	 * Create contents of the window.
	 */
	protected void createContents() {
		shell = new Shell();
		shell.setSize(450, 374);
		shell.setText("Step One");
		shell.setLayout(null);

		Label lblFromFile = new Label(shell, SWT.NONE);
		lblFromFile.setBounds(10, 10, 55, 15);
		lblFromFile.setText("From File:");

		sourceFileTxt = new Text(shell, SWT.BORDER);
		sourceFileTxt.setBounds(10, 32, 333, 21);
		sourceFileTxt.setText("F:\\r6.2\\file.csv");
```

```
Button btnBrowse = new Button(shell, SWT.NONE);
btnBrowse.addMouseListener(new MouseAdapter() {
		@Override
		public void mouseUp(MouseEvent e) {
FileDialog dialog = new FileDialog(new Shell());
String[] filterExt = { "*.csv", "*.tsv","*.xml","*.txt", "*.*" };
dialog.setFilterExtensions(filterExt);
		  String path = dialog.open();
		if(path != null)
		{
		 sourceFileTxt.setText(path);
		  }
		}
});
btnBrowse.setBounds(349, 30, 75, 25);
btnBrowse.setText("Browse");

Label lblToFolder = new Label(shell, SWT.NONE);
lblToFolder.setBounds(10, 59, 55, 15);
lblToFolder.setText("To Folder:");

destFoldertxt = new Text(shell, SWT.BORDER);
destFoldertxt.setBounds(10, 80, 333, 21);
destFoldertxt.setText("F:\\r6.2\\Project_Files\\");

Button btnNewButton = new Button(shell, SWT.NONE);
btnNewButton.addMouseListener(new MouseAdapter() {
```

```
		@Override
		public void mouseUp(MouseEvent e) {
		DirectoryDialog dialog = new DirectoryDialog(new Shell());
		String path = dialog.open();

		if(path != null)
		{
		destFoldertxt.setText(path);
		}
		}
});
btnNewButton.setBounds(349, 78, 75, 25);
btnNewButton.setText("Browse");

Label lblSearchFor = new Label(shell, SWT.NONE);
lblSearchFor.setBounds(10, 107, 55, 15);
lblSearchFor.setText("Search for:");

searchKeytxt = new Text(shell, SWT.BORDER);
searchKeytxt.setBounds(10, 128, 414, 21);
searchKeytxt.setText("SAB1954");

Button btnSearch = new Button(shell, SWT.NONE);
btnSearch.addMouseListener(new MouseAdapter() {
		@Override
		public void mouseUp(MouseEvent e) {
			String sf = sourceFileTxt.getText();
```

```
                    System.out.println(sf);

                    String separator = "\\";

                    String separatorDot = ".";

                    String pathArray[] = sf.split(Pattern.quote(separator));



                    System.out.println( pathArray[pathArray.length-1]);

                    String pathArayDot[] = pathArray[pathArray.length-1].split(Pattern.quote(separatorDot));



            System.out.println( pathArayDot[0]);



            String filename = pathArayDot[0];

            String result1 = identifyUser(sourceFileTxt.getText(),destFoldertxt.getText(),filename,searchKeytxt.getText());

            }

});

btnSearch.setBounds(10, 162, 414, 25);

btnSearch.setText("Search");



Group grpStatus = new Group(shell, SWT.NONE);

grpStatus.setText("Status");

grpStatus.setBounds(10, 193, 414, 142);



Label lblRecordsFound = new Label(grpStatus, SWT.NONE);

lblRecordsFound.setBounds(10, 20, 97, 15);

lblRecordsFound.setText("Records found:");



recordsTxt = new Label(grpStatus, SWT.NONE);

recordsTxt.setBounds(113, 20, 279, 15);
```

```
recordsTxt.setText("0");


statusTxt = new StyledText(grpStatus, SWT.WRAP);
statusTxt.setEditable(false);
statusTxt.setBounds(10, 41, 396, 60);
statusTxt.setText("Idle");


btnNextStep = new Button(grpStatus, SWT.NONE);
btnNextStep.addMouseListener(new MouseAdapter() {
	@Override
	public void mouseUp(MouseEvent e) {
		if (st.blacklist == null)st.open();
		else {
			if (st.shell.isDisposed()) {
			st.open();
			}
		}

	try {
		st.setFormFile(result1Path);
		st.setToFile(destFoldertxt.getText());
		st.setUser(searchKeytxt.toString());
		} catch (Exception e2) {
			// TODO: handle exception
		}
	}
});
```

```
                btnNextStep.setBounds(10, 107, 396, 25);

                btnNextStep.setText("Next step");

                btnNextStep.setVisible(false);



        }

        String result1Path;

        long lastUpdate = 0l;

        public String identifyUser(String fromFileStr, String toFolderStr,String fileName, String searchStr) {

                int threatsFound = 0;

                File file =new File(fromFileStr);

Scanner in = null;



        statusTxt.setText("Scanning... please wait");

try {

        in = new Scanner(file);

        statusTxt.setText("Scanning... please wait");

        result1Path = (toFolderStr.endsWith("\\")?toFolderStr :toFolderStr + "\\") +searchStr+"_"+ fileName+"_result1.csv";

        System.out.println(result1Path);

        boolean createPath = FilesUtil.createPath(result1Path);

        if (!createPath) {

        statusTxt.setText("Error... Could not create path");

                return null;

        }

        boolean firstLine = true;

    while(in.hasNext())

    {

      String line=in.nextLine();
```

```
            if (firstLine) {
            FilesUtil.writeToTextFile(result1Path, line + "\n");
            }
            firstLine = false;

    if(line.contains(searchStr)) {
      System.out.println(line);
      FilesUtil.appendNewLineToTextFile(result1Path, line);
      threatsFound++;
      if (new Date().getTime() -500 > lastUpdate ) {
                        lastUpdate = new Date().getTime();
                        recordsTxt.setText(threatsFound + "");
            }
    }
  }
  recordsTxt.setText(threatsFound + "");
            statusTxt.setText("Done... file created:\n" +result1Path);
            btnNextStep.setVisible(true);
            return result1Path;
} catch (FileNotFoundException e) {
            statusTxt.setText("Error... caught exception: FileNotFoundException");
  e.printStackTrace();
  return null;
} catch (IOException e) {
  e.printStackTrace();
            statusTxt.setText("Error... caught exception: IOException");
            return null;
```

```
}

        }

}
```

A.1.2 Δεύτερου βήματος διαδικασίας εξόρυξης δεδομένων

```
package Main;

import org.eclipse.swt.widgets.Display;

import org.eclipse.swt.widgets.FileDialog;

import org.eclipse.swt.widgets.Shell;

import org.eclipse.swt.widgets.Label;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.Locale;

import java.util.Scanner;

import org.eclipse.swt.SWT;

import org.eclipse.swt.custom.StyledText;

import org.eclipse.swt.widgets.Text;

import org.eclipse.swt.widgets.Button;

import org.eclipse.swt.widgets.DirectoryDialog;

import org.eclipse.swt.events.SelectionAdapter;
```

```
import org.eclipse.swt.events.SelectionEvent;

import org.eclipse.swt.widgets.Group;

import org.eclipse.wb.swt.SWTResourceManager;

import org.eclipse.swt.events.MouseAdapter;

import org.eclipse.swt.events.MouseEvent;

import org.eclipse.swt.events.MouseListener;

import java.util.function.Consumer;

import java.util.regex.Pattern;

import javax.sound.midi.Soundbank;


public class StepTwo {

	protected Shell shell;

	private Text sourceFileTxt;

	private Text toFolderTxt;

	private Text currentBlacklistedTxt;

	private Text currentBlacklisted2Txt;

	Group grpBlacklist;

	Group grpBlacklist2;

	StyledText statusTxt;

	Label threatsCountTxt;


	StepThree st;

	Button makeReportsBtn;


	private StyledText blacklistedLabel;

	private StyledText blacklistedLabel2;
```

```
ArrayList<String> blacklist;
ArrayList<String> blacklist2;

String headers = "date,user,pc,keyword,number";
String resultpath = "";
String user;

public String getUser() {
	return user;
}
public void setUser(String user) {
	this.user = user;
}
/**
 * Launch the application.
 * @param args
 */
public static void main(String[] args) {
	try {
		StepTwo window = new StepTwo();
		window.open();
	} catch (Exception e) {
		e.printStackTrace();
	}
}
/**
 * Open the window.
```

```
 */
public void open() {
		Display display = Display.getDefault();
		createContents();
		shell.open();
		shell.layout();
		while (!shell.isDisposed()) {
			if (!display.readAndDispatch()) {
				display.sleep();
			}
		}
}
/**
 * Create contents of the window.
 */
protected void createContents() {
		st = new StepThree();
		blacklist = new ArrayList<String>();
		blacklist2 = new ArrayList<String>();
		shell = new Shell();
		shell.setSize(550, 532);
		shell.setText("Step Two");

		Label lblFromFile = new Label(shell, SWT.NONE);
		lblFromFile.setBounds(10, 10, 55, 15);
		lblFromFile.setText("From file:");
```

```
sourceFileTxt = new Text(shell, SWT.BORDER);
sourceFileTxt.setBounds(10, 31, 433, 21);

Button btnNewButton = new Button(shell, SWT.NONE);
btnNewButton.addSelectionListener(new SelectionAdapter() {
        @Override
        public void widgetSelected(SelectionEvent e) {
        FileDialog dialog = new FileDialog(new Shell());
String[] filterExt = { "*.csv", "*.tsv","*.xml","*.txt", "*.*" };
dialog.setFilterExtensions(filterExt);
        String path = dialog.open();

        if(path != null)
        {
        sourceFileTxt.setText(path);
          }
        }
});
btnNewButton.setBounds(449, 29, 75, 25);
btnNewButton.setText("Browse");

Label lblToFolder = new Label(shell, SWT.NONE);
lblToFolder.setBounds(10, 69, 55, 15);
lblToFolder.setText("To folder:");

toFolderTxt = new Text(shell, SWT.BORDER);
toFolderTxt.setBounds(10, 90, 433, 21);
```

```
Button btnBrowse = new Button(shell, SWT.NONE);
btnBrowse.addMouseListener(new MouseAdapter() {
		@Override
		public void mouseUp(MouseEvent e) {
		  DirectoryDialog dialog = new DirectoryDialog(new Shell());
		  String path = dialog.open();

		  if(path != null)
		  {
		  toFolderTxt.setText(path);
		  }
		}
});
btnBrowse.setBounds(449, 88, 75, 25);
btnBrowse.setText("Browse");

Group grpInputType = new Group(shell, SWT.NONE);
grpInputType.setText("Input Type");
grpInputType.setBounds(10, 117, 97, 124);

Button btnFile = new Button(grpInputType, SWT.RADIO);
btnFile.addSelectionListener(new SelectionAdapter() {
		@Override
		public void widgetSelected(SelectionEvent e) {
			renderFile();
		}
```

```
});
btnFile.setBounds(10, 30, 66, 16);
btnFile.setText("file");

Button btnEmail = new Button(grpInputType, SWT.RADIO);
btnEmail.addSelectionListener(new SelectionAdapter() {
	@Override
	public void widgetSelected(SelectionEvent e) {
		renderEmail();
	}
});
btnEmail.setBounds(10, 52, 77, 16);
btnEmail.setText("email");

Button btnWebsite = new Button(grpInputType, SWT.RADIO);
btnWebsite.addSelectionListener(new SelectionAdapter() {
	@Override
	public void widgetSelected(SelectionEvent e) {
		renderWebsite();
	}
});
btnWebsite.setBounds(10, 74, 77, 16);
btnWebsite.setText("website");

Button btnLogon = new Button(grpInputType, SWT.RADIO);
btnLogon.addSelectionListener(new SelectionAdapter() {
	@Override
```

```
		public void widgetSelected(SelectionEvent e) {
			renderLogon();
		}
});
btnLogon.setBounds(10, 96, 66, 16);
btnLogon.setText("logon");

grpBlacklist = new Group(shell, SWT.NONE);
grpBlacklist.setText("blacklist1");
grpBlacklist.setBounds(113, 117, 200, 180);
grpBlacklist.setVisible(false);

currentBlacklistedTxt = new Text(grpBlacklist, SWT.BORDER);
currentBlacklistedTxt.setBounds(10, 20, 104, 21);

Button button = new Button(grpBlacklist, SWT.NONE);
button.addMouseListener(new MouseAdapter() {
		@Override
		public void mouseUp(MouseEvent e) {

		if (currentBlacklistedTxt.getText() != null && !currentBlacklistedTxt.getText().equalsIgnoreCase("")) {
		if (!blacklist.contains(currentBlacklistedTxt.getText()))blacklist.add(currentBlacklistedTxt.getText());
					StringBuilder sb = new StringBuilder();
					for (String s : blacklist) {
						if (!sb.toString().equalsIgnoreCase("")) {
						sb.append(", ");
						}
```

```
                                        sb.append(s);
                                }
                blacklistedLabel.setText(sb.toString());
                currentBlacklistedTxt.setText("");
                currentBlacklistedTxt.forceFocus();
                }
                }
});
button.setBounds(120, 18, 32, 25);
button.setText("+");

Button button_1 = new Button(grpBlacklist, SWT.NONE);
button_1.addMouseListener(new MouseAdapter() {
                @Override
                public void mouseUp(MouseEvent e) {
                for (String s : blacklist) {
                if (currentBlacklistedTxt.getText() != null && currentBlacklistedTxt.getText().equalsIgnoreCase(s)) {
                                        blacklist.remove(s);
                                        break;
                                        }
                        }
                StringBuilder sb = new StringBuilder();
                for (String s : blacklist) {
                        if (!sb.toString().equalsIgnoreCase("")) {
                        sb.append(", ");
                                        }
                        sb.append(s);
```

```
                        }

                        blacklistedLabel.setText(sb.toString());

                        currentBlacklistedTxt.setText("");

                        currentBlacklistedTxt.forceFocus();

                }

});

button_1.setBounds(158, 18, 32, 25);

button_1.setText("-");



blacklistedLabel = new StyledText(grpBlacklist, SWT.V_SCROLL | SWT.WRAP);

blacklistedLabel.setEditable(false);

blacklistedLabel.setBounds(10, 47, 180, 123);

blacklistedLabel.setBackground(SWTResourceManager.getColor(SWT.COLOR_WHITE));



grpBlacklist2 = new Group(shell, SWT.NONE);

grpBlacklist2.setText("blacklist2");

grpBlacklist2.setBounds(324, 117, 200, 180);

grpBlacklist2.setVisible(false);



currentBlacklisted2Txt = new Text(grpBlacklist2, SWT.BORDER);

currentBlacklisted2Txt.setBounds(10, 20, 104, 21);



Button button_2 = new Button(grpBlacklist2, SWT.NONE);

button_2.addMouseListener(new MouseAdapter() {

@Override

public void mouseUp(MouseEvent e) {

        if (currentBlacklisted2Txt.getText() != null && !currentBlacklisted2Txt.getText().equalsIgnoreCase("")) {
```

```
                if (!blacklist2.contains(currentBlacklisted2Txt.getText()))blacklist2.add(currentBlacklisted2Txt.getText());
                        StringBuilder sb = new StringBuilder();
                        for (String s : blacklist2) {
                                if (!sb.toString().equalsIgnoreCase("")) {
                                        sb.append(", ");
                                        }
                                sb.append(s);
                                }
                        blacklistedLabel2.setText(sb.toString());
                        currentBlacklisted2Txt.setText("");
                        currentBlacklisted2Txt.forceFocus();
                        }
                }
});
button_2.setBounds(120, 18, 32, 25);
button_2.setText("+");

Button button_3 = new Button(grpBlacklist2, SWT.NONE);
button_3.addMouseListener(new MouseAdapter() {
@Override
public void mouseUp(MouseEvent e) {
                for (String s : blacklist2) {
                if (currentBlacklisted2Txt.getText() != null && currentBlacklisted2Txt.getText().equalsIgnoreCase(s)) {
                        blacklist2.remove(s);
                        break;
                        }
                }
```

```
StringBuilder sb = new StringBuilder();
for (String s : blacklist2) {
        if (!sb.toString().equalsIgnoreCase("")) {
                sb.append(", ");
        }
        sb.append(s);
        }
blacklistedLabel2.setText(sb.toString());
currentBlacklisted2Txt.setText("");
currentBlacklisted2Txt.forceFocus();
}
});
        button_3.setBounds(158, 18, 32, 25);
        button_3.setText("-");

        blacklistedLabel2 = new StyledText(grpBlacklist2, SWT.V_SCROLL | SWT.WRAP);
        blacklistedLabel2.setEditable(false);
        blacklistedLabel2.setBounds(10, 47, 180, 123);

        Button btnScan = new Button(shell, SWT.NONE);
        btnScan.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseUp(MouseEvent e) {
                String sf = sourceFileTxt.getText();
                System.out.println(sf);
                String separator = "\\";
                String separatorDot = ".";
```

```
                String pathArray[] = sf.split(Pattern.quote(separator));


                System.out.println( pathArray[pathArray.length-1]);
                String pathArayDot[] = pathArray[pathArray.length-1].split(Pattern.quote(separatorDot));


                System.out.println( pathArayDot[0]);
                String filename = pathArayDot[0];


                resultpath = identifyThreat(sourceFileTxt.getText(), toFolderTxt.getText(), filename);
            }
        });
            btnScan.setBounds(10, 303, 514, 31);
            btnScan.setText("Scan");

            Group grpStatus = new Group(shell, SWT.NONE);
            grpStatus.setText("Status");
            grpStatus.setBounds(10, 341, 514, 142);

            Label lblThreatsIdentified = new Label(grpStatus, SWT.NONE);
            lblThreatsIdentified.setBounds(10, 21, 115, 15);
            lblThreatsIdentified.setText("Threats identified:");

            threatsCountTxt = new Label(grpStatus, SWT.NONE);
            threatsCountTxt.setBounds(131, 21, 373, 15);
            threatsCountTxt.setText("0");

            statusTxt = new StyledText(grpStatus, SWT.WRAP);
```

```
		statusTxt.setText("Idle");
		statusTxt.setEditable(false);
		statusTxt.setBounds(10, 40, 494, 60);

		makeReportsBtn = new Button(grpStatus, SWT.NONE);
		makeReportsBtn.addMouseListener(new MouseAdapter() {
		@Override
		public void mouseUp(MouseEvent e) {

		if (st.shell == null || st.shell.isDisposed()) {
					st.open();
			}
		if (user != null)st.setUser(user);
st.setDestinationFolder(toFolderTxt.getText().endsWith("\\")?toFolderTxt.getText() + "reports" : (toFolderTxt.getText() + "\\reports"));

}
});
		makeReportsBtn.setBounds(10, 107, 494, 25);
		makeReportsBtn.setText("Merge Files");
		makeReportsBtn.setVisible(false);

		Group grpLogontime = new Group(shell, SWT.NONE);
		grpLogontime.setText("LogonTime");
		grpLogontime.setLocation(113, 117);
		grpLogontime.setSize(200, 180);
		grpLogontime.setVisible(false);
}
```

```
public void setFormFile(String input) {
		try {
		sourceFileTxt.setText(input);
		}catch (Exception e) {
		// TODO: handle exception
		}
}
public void setToFile(String input) {
		try {
			toFolderTxt.setText(input);
		}catch (Exception e) {
			// TODO: handle exception
		}
}
public void renderLogon() {
		emailSelected = false;
		websiteSelected = false;
		fileSelected = false;
		logonSelected = true;
		grpBlacklist.setVisible(false);
		grpBlacklist2.setVisible(false);
}
public void renderEmail() {
		emailSelected = true;
		websiteSelected = false;
		fileSelected = false;
		logonSelected = false;
```

```
                grpBlacklist.setVisible(true);

                grpBlacklist2.setVisible(false);

                grpBlacklist.setText("Email Threats");

        }

        public void renderFile() {

                fileSelected = true;

                websiteSelected = false;

                emailSelected = false;

                logonSelected = false;

                grpBlacklist.setVisible(true);

                grpBlacklist2.setVisible(false);

                grpBlacklist.setText("File Threats");



        }

        public void renderWebsite() {

                websiteSelected = true;

                emailSelected = false;

                fileSelected = false;

                logonSelected = false;

                grpBlacklist.setVisible(true);

                grpBlacklist2.setVisible(true);

                grpBlacklist.setText("Web-Seek Threats");

                grpBlacklist2.setText("File-Share Threats");



        }

        boolean emailSelected = false;

        boolean fileSelected = false;
```

```
boolean websiteSelected = false;

boolean logonSelected = false;

		public String blameLine(String inputLine,String keyWord, int severity) {

		if (inputLine == null)return null;



		String result = "";

		String separatorcomma = ",";

		String lineArray[] =  inputLine.split(Pattern.quote(separatorcomma));

		if (lineArray!= null && lineArray.length>5) {

			result =lineArray[1] + ","+lineArray[2] + ","+lineArray[3] + ","+keyWord + ","+severity;

		}

		return result;

}

public String identifyThreat(String fromFileStr, String toFolderStr,String fileName) {

		String prefix = "";

		if (emailSelected)prefix = "email_";

		else if (fileSelected)prefix = "files_";

		else if (websiteSelected)prefix = "web_";

		else if (logonSelected) prefix = "logon_";



String result2Path = (toFolderStr.endsWith("\\")?toFolderStr :toFolderStr +"\\") +prefix + fileName.replace("result1", "result2") + ".csv";

System.out.println(result2Path);

statusTxt.setText("Identifying threats");

System.out.println("Identifying threats in " +fromFileStr);

		String separatorcomma = ",";

if(emailSelected || websiteSelected || fileSelected || logonSelected) {

		boolean createPath = FilesUtil.createPath(result2Path);
```

```
        if (!createPath) {
                        statusTxt.setText("Error");
                        System.out.println("Error: FileNotFoundException " );
                        return null;
        }
        }
        boolean firstLine = true;
Scanner in = null;
int threatsFound = 0;
long lastUpdate=0l;
                        File file =new File(fromFileStr);
try {
                        in = new Scanner(file);
                        } catch (FileNotFoundException e1) {
                        // TODO Auto-generated catch block

                        statusTxt.setText("Error");
                        System.out.println("Error: FileNotFoundException " );
                                e1.printStackTrace();
                                return null;
                        }
        if (emailSelected) {
                        statusTxt.setText("Identifying threats- InputType: Email");
                        System.out.println("Identifying threats- InputType: Email" );
  while(in.hasNext())
  {
        String line=in.nextLine();
```

```
            if (firstLine) {

            try {

                        FilesUtil.writeToTextFile(result2Path, headers + "\n");

                        } catch (IOException e) {

                        e.printStackTrace();

}

            }

            firstLine = false;

            String lineStr = "";

            if (line != null && line.length()>1) {



            String lineArray[] =  line.split(Pattern.quote(separatorcomma));

            if (lineArray != null && lineArray.length > 10) {

                        String sizeStr = lineArray[9];

                        try {

                                    long l = Long.parseLong(sizeStr);

                                    if (l > 1024l * 200) {

                                    System.out.println(line);

              try {

                        FilesUtil.appendNewLineToTextFile(result2Path, blameLine(line,"Attachments_200k", 8));

                                    } catch (IOException e) {

                                    // TODO Auto-generated catch block

                                    e.printStackTrace();

                                    }

              threatsFound++;

            if (new Date().getTime() -500 > lastUpdate ) {

                        lastUpdate = new Date().getTime();
```

```
                threatsCountTxt.setText(threatsFound + "");

        }

                        }else if (l > 1024l * 100) {

                         try {

                FilesUtil.appendNewLineToTextFile(result2Path, blameLine(line,"Attachments_100k", 7));

                                        } catch (IOException e) {

                                        // TODO Auto-generated catch block

                                        e.printStackTrace();

                                        }

          threatsFound++;

                if (new Date().getTime() -500 > lastUpdate ) {

                        lastUpdate = new Date().getTime();

                        threatsCountTxt.setText(threatsFound + "");

                }

                        }else if (l > 1024l * 50) {

                 try {

                        FilesUtil.appendNewLineToTextFile(result2Path, blameLine(line,"Attachments_50k", 6));

                                        } catch (IOException e) {

                                        // TODO Auto-generated catch block

                                        e.printStackTrace();

                                        }

          threatsFound++;

                if (new Date().getTime() -500 > lastUpdate ) {

                        lastUpdate = new Date().getTime();

                        threatsCountTxt.setText(threatsFound + "");

                }

                        }
```

```
                    }catch(Exception e) {
                            e.printStackTrace();
                    }
        }
        }
        for (String s: blacklist) {
                    if(line.contains(s)) {
    System.out.println(line);
    try {
                    FilesUtil.appendNewLineToTextFile(result2Path, blameLine(line,"EmailThreat", 3));
                    } catch (IOException e) {
                    e.printStackTrace();
                    }
    threatsFound++;

    if (new Date().getTime() -500 > lastUpdate ) {
                    lastUpdate = new Date().getTime();
                    threatsCountTxt.setText(threatsFound + "");
        }
        break;
   }
        }
}
        }else if (fileSelected) {
                    statusTxt.setText("Identifying threats- InputType: File");
                    System.out.println("Identifying threats- InputType: File" );
while(in.hasNext())
```

```
{
  String line=in.nextLine();
        if (firstLine) {
        try {
                        FilesUtil.writeToTextFile(result2Path, headers + "\n");
                        } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                        }
        }
        firstLine = false;

        for (String s: blacklist) {
                        if(line.contains(s)) {
      System.out.println(line);
      try {
                        FilesUtil.appendNewLineToTextFile(result2Path, blameLine(line,"FileThreat", 4));
                        } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                        }
      threatsFound++;
      if (new Date().getTime() -500 > lastUpdate ) {
                        lastUpdate = new Date().getTime();
                        threatsCountTxt.setText(threatsFound + "");
        }
        break;
```

```
        }
            }
    }

}else if (websiteSelected) {
            statusTxt.setText("Identifying threats- InputType: Website");
            System.out.println("Identifying threats- InputType: Website" );
    while(in.hasNext())
    {
      String line=in.nextLine();
            if (firstLine) {
            try {
                        FilesUtil.writeToTextFile(result2Path, headers + "\n");
                        } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                        }
            }
            firstLine = false;

            for (String s: blacklist) {
                        if(line.contains(s)) {
          System.out.println(line);
          try {
                        FilesUtil.appendNewLineToTextFile(result2Path, blameLine(line,"Webseek", 0));
                        } catch (IOException e) {
                        // TODO Auto-generated catch block
```

```
                    e.printStackTrace();
                    }
    threatsFound++;
    if (new Date().getTime() -500 > lastUpdate ) {
                    lastUpdate = new Date().getTime();
                    threatsCountTxt.setText(threatsFound + "");
      }
      break;
  }else {
      for (String s2: blacklist2) {
                    if(line.contains(s2)) {
        System.out.println(line);
        try {
                    FilesUtil.appendNewLineToTextFile(result2Path, blameLine(line,"FileShare", 1));
                    } catch (IOException e) {
                    e.printStackTrace();
                    }
        threatsFound++;
        if (new Date().getTime() -500 > lastUpdate ) {
                    lastUpdate = new Date().getTime();
                    threatsCountTxt.setText(threatsFound + "");
      }
      break;
                    }
      }
  }
      }
```

```
}

        }else if (logonSelected){

                        statusTxt.setText("Identifying threats- InputType: Logon");

                        System.out.println("Identifying threats- InputType: Logon" );

                         while(in.hasNext())

                    {

                      String line=in.nextLine();

                        if (firstLine) {

                        try {

                                        FilesUtil.writeToTextFile(result2Path, headers + "\n");

                                        } catch (IOException e) {

                                        // TODO Auto-generated catch block

                                        e.printStackTrace();

                                        }

                        }

                        firstLine = false;

                        try {

                                        String lineArray[] =  line.split(Pattern.quote(separatorcomma));

                                        DateFormat dfin = new SimpleDateFormat("dd/MM/yyy hh:mm:ss", Locale.ENGLISH);

                                        Date date =  dfin.parse(lineArray[1]);

                                        if (date.getHours() <= 6 || date.getHours()>= 22) {

                                         try {

                                                        FilesUtil.appendNewLineToTextFile(result2Path, blameLine(line,"LogonThreat", 10));

                                                        } catch (IOException e) {

                                                        e.printStackTrace();

                                                        }

        threatsFound++;
```

```
                                }

                        } catch (Exception e) {

                        e.printStackTrace();

                        }

                }

        }

        threatsCountTxt.setText(threatsFound + "");

        statusTxt.setText("Done- "+result2Path );

        System.out.println("Done- Blacklisted records" );

        System.out.println("Done- "+result2Path );

        makeReportsBtn.setVisible(true);

        return result2Path;

        }

}
```

A.1.3 Τρίτου βήματος διαδικασίας εξόρυξης δεδομένων

```
package Main;

import org.eclipse.swt.widgets.Display;

import org.eclipse.swt.widgets.FileDialog;

import org.eclipse.swt.widgets.Shell;

import org.eclipse.swt.custom.StyledText;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Scanner;
```

```
import org.eclipse.swt.SWT;

import org.eclipse.wb.swt.SWTResourceManager;

import org.eclipse.swt.widgets.Group;

import org.eclipse.swt.widgets.Button;

import org.eclipse.swt.widgets.DirectoryDialog;

import org.eclipse.swt.events.MouseAdapter;

import org.eclipse.swt.events.MouseEvent;

import org.eclipse.swt.widgets.Label;

import org.eclipse.swt.widgets.Text;



public class StepThree {

	protected Shell shell;

	StyledText fileList;

	ArrayList<String> files;

	private Text toFolderTxt;

	private Text userNameTxt;

	Button btnNextStep;

	StepFour st;

	String headers = "date,user,pc,keyword,number";

	/**

	 * Launch the application.

	 * @param args

	 */

	public static void main(String[] args) {

		try {

		StepThree window = new StepThree();

		window.open();
```

```
                } catch (Exception e) {
                e.printStackTrace();
            }
            }
            /**
             * Open the window.
             */
            public void open() {
            Display display = Display.getDefault();
            createContents();
            shell.open();
            shell.layout();
            while (!shell.isDisposed()) {
            if (!display.readAndDispatch()) {
`           display.sleep();
            }
            }
}
            /**
             * Create contents of the window.
             */
            protected void createContents() {
            st = new StepFour();
            files = new ArrayList<>();
            shell = new Shell();
            shell.setSize(450, 537);
            shell.setText("Step Three");
```

```
Group grpMergeFiles = new Group(shell, SWT.NONE);

grpMergeFiles.setText("Merge Files:");

grpMergeFiles.setBounds(10, 10, 414, 197);



fileList = new StyledText(grpMergeFiles, SWT.WRAP);

fileList.setBounds(10, 23, 394, 123);

fileList.setEditable(false);

fileList.setBackground(SWTResourceManager.getColor(SWT.COLOR_WHITE));



Button btnClear = new Button(grpMergeFiles, SWT.NONE);

btnClear.addMouseListener(new MouseAdapter() {

@Override

public void mouseUp(MouseEvent e) {

fileList.setText("");

files.clear();

}

});

btnClear.setBounds(329, 162, 75, 25);

btnClear.setText("Clear");



Button btnAddFile = new Button(grpMergeFiles, SWT.NONE);

btnAddFile.addMouseListener(new MouseAdapter() {

@Override

public void mouseUp(MouseEvent e) {

FileDialog dialog = new FileDialog(new Shell(),SWT.MULTI);

    String[] filterExt = { "*.csv", "*.tsv","*.xml","*.txt", "*.*" };
```

```
                dialog.setFilterExtensions(filterExt);
                if (dialog.open() != null) {
                String[] names = dialog.getFileNames();
String path = dialog.getFilterPath().toString().endsWith("\\")?dialog.getFilterPath().toString() : dialog.getFilterPath().toString() + "\\";
            if (names != null && names.length > 0) {

                    files.clear();
                    for (String name : names) {
                    System.out.println("fileName: " +name);
                    files.add(path+ name);
                    fileList.setText(fileList.getText() +"\n" +path + name);
                    }
                    }
            }
}
});
            btnAddFile.setBounds(248, 162, 75, 25);
            btnAddFile.setText("Add File");

            Group grpTo = new Group(shell, SWT.NONE);
            grpTo.setText("To:");
            grpTo.setBounds(10, 213, 414, 132);

            Label lblFolder = new Label(grpTo, SWT.NONE);
            lblFolder.setBounds(10, 25, 55, 15);
            lblFolder.setText("Folder:");
```

```
toFolderTxt = new Text(grpTo, SWT.BORDER);

toFolderTxt.setText("F:\\r6.2\\tmp2");

toFolderTxt.setBounds(10, 46, 313, 21);



Button browseBtn = new Button(grpTo, SWT.NONE);

browseBtn.addMouseListener(new MouseAdapter() {

@Override

public void mouseUp(MouseEvent e) {

DirectoryDialog dialog = new DirectoryDialog(new Shell());

String path = dialog.open();



if(path != null)

{

                toFolderTxt.setText(path);

                }

                }

});

browseBtn.setBounds(329, 44, 75, 25);

browseBtn.setText("Browse");



Label lblUsername = new Label(grpTo, SWT.NONE);

lblUsername.setBounds(10, 73, 119, 15);

lblUsername.setText("UserName:");



userNameTxt = new Text(grpTo, SWT.BORDER);

userNameTxt.setBounds(10, 94, 313, 21);
```

```
		Button btnMerge = new Button(shell, SWT.NONE);
		btnMerge.addMouseListener(new MouseAdapter() {
		@Override
		public void mouseUp(MouseEvent e) {
		mergeFiles();
		}
	});
		btnMerge.setBounds(10, 351, 414, 25);
		btnMerge.setText("Merge");

		Group grpStatus = new Group(shell, SWT.NONE);
		grpStatus.setText("Status");
		grpStatus.setBounds(10, 385, 414, 103);
		btnNextStep = new Button(grpStatus, SWT.NONE);
		btnNextStep.addMouseListener(new MouseAdapter() {
	@Override
	public void mouseUp(MouseEvent e) {
		if (st.shell == null || st.shell.isDisposed()) {
			st.open();
		}
st.setFormFile( toFolderTxt.getText().endsWith("\\") ? toFolderTxt.getText() : toFolderTxt.getText() + "\\" + userNameTxt.getText() + "_merged_result.csv");
st.setToFolder(toFolderTxt.getText().endsWith("\\")?toFolderTxt.getText() + "reports" : (toFolderTxt.getText() + "\\reports"));
	}
		});
		btnNextStep.setBounds(10, 68, 394, 25);
		btnNextStep.setText("Next Step");
		btnNextStep.setVisible(false);
```

```
}

public void mergeFiles() {
String filePath = toFolderTxt.getText().endsWith("\\") ? toFolderTxt.getText() : toFolderTxt.getText() + "\\";
String name = userNameTxt.getText() + "_merged_result.csv";

try {
        FilesUtil.checkIfExistsAndHeadersAndMdr(filePath, name, headers);
        FilesUtil.writeToTextFile(filePath + name, headers + "\n");
        } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
}
Scanner in = null;
boolean firstLine = true;
for (String fileName : files) {
        File file =new File(fileName);
try {

        System.out.println("fileName: " +fileName);
        in = new Scanner(file);
        while(in.hasNext())
{
String line=in.nextLine();
if (!firstLine) {
try {
        FilesUtil.appendNewLineToTextFile(filePath + name, line);
```

```
                    } catch (IOException e) {
                    e.printStackTrace();
                    }
                }else {
                firstLine = false;
                }
            }
                } catch (FileNotFoundException e1) {
                e1.printStackTrace();
                return;
                }
            }
                btnNextStep.setVisible(true);
                }

                public void setUser(String user) {
                userNameTxt.setText(user);
                }

                public void setDestinationFolder(String toFolder) {
                toFolderTxt.setText(toFolder);
                }
            }
```

A.1.4 Τέταρτου βήματος διαδικασίας εξόρυξης δεδομένων

```
package Main;

import org.eclipse.swt.widgets.Display;

import org.eclipse.swt.widgets.FileDialog;

import org.eclipse.swt.widgets.Shell;

import org.eclipse.swt.widgets.Label;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.PrintWriter;

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.Date;

import java.util.Locale;

import java.util.Scanner;

import java.util.regex.Pattern;

import org.eclipse.swt.SWT;

import org.eclipse.swt.widgets.Text;

import org.eclipse.swt.widgets.Button;
```

```
import org.eclipse.swt.widgets.DirectoryDialog;

import org.eclipse.swt.widgets.Group;

import org.eclipse.swt.custom.StyledText;

import org.eclipse.swt.events.MouseAdapter;

import org.eclipse.swt.events.MouseEvent;

public class StepFour {

	protected Shell shell;

	private Text sourceFileTxt;

	private Text toFolderTxt;

	private Text userTxt;



	StyledText statusTxt;

	String headers = "date,user,pc,keyword,number";



	/**

	 * Launch the application.

	 * @param args

	 */

	public static void main(String[] args) {

	try {

		StepFour window = new StepFour();

		window.open();

		} catch (Exception e) {

		e.printStackTrace();

		}

	}

	/**
```

```
        * Open the window.
        */
        public void open() {
                Display display = Display.getDefault();
                createContents();
                shell.open();
                shell.layout();
                while (!shell.isDisposed()) {
                if (!display.readAndDispatch()) {
                        display.sleep();
                }
        }
}

        /**
        * Create contents of the window.
        */
        protected void createContents() {
                shell = new Shell();
                shell.setSize(469, 352);
                shell.setText("Step Four");

                Label label = new Label(shell, SWT.NONE);
                label.setText("From file:");
                label.setBounds(10, 10, 55, 15);
                sourceFileTxt = new Text(shell, SWT.BORDER);
                sourceFileTxt.setBounds(10, 31, 350, 21);
```

```
Button button = new Button(shell, SWT.NONE);

button.addMouseListener(new MouseAdapter() {

@Override

public void mouseUp(MouseEvent e) {

FileDialog dialog = new FileDialog(new Shell());

String[] filterExt = { "*.csv", "*.tsv","*.xml","*.txt", "*.*" };

dialog.setFilterExtensions(filterExt);

String path = dialog.open();



if(path != null)

{

sourceFileTxt.setText(path);

}

}

});

button.setText("Browse");

button.setBounds(366, 29, 75, 25);



Label label_1 = new Label(shell, SWT.NONE);

label_1.setText("To folder:");

label_1.setBounds(10, 58, 55, 15);



toFolderTxt = new Text(shell, SWT.BORDER);

toFolderTxt.setBounds(10, 79, 350, 21);

Button button_1 = new Button(shell, SWT.NONE);

button_1.addMouseListener(new MouseAdapter() {
```

```
		@Override
		public void mouseUp(MouseEvent e) {
		DirectoryDialog dialog = new DirectoryDialog(new Shell());
		String path = dialog.open();

		if(path != null)
		{
		toFolderTxt.setText(path);
		}
	}
		});
		button_1.setText("Browse");
		button_1.setBounds(366, 77, 75, 25);

		Button btnWeeklyReport = new Button(shell, SWT.NONE);
		btnWeeklyReport.addMouseListener(new MouseAdapter() {
		@Override
		public void mouseUp(MouseEvent e) {
		createWeeklyReports();
		}
	});
		btnWeeklyReport.setBounds(10, 190, 172, 25);
		btnWeeklyReport.setText("Weekly report");

		Button btnMonthlyReport = new Button(shell, SWT.NONE);
		btnMonthlyReport.addMouseListener(new MouseAdapter() {
		@Override
```

```
        public void mouseUp(MouseEvent e) {
        createMonthlyReports();
}
        });
        btnMonthlyReport.setBounds(188, 190, 172, 25);
        btnMonthlyReport.setText("Monthly Report");

        Group grpStatus = new Group(shell, SWT.NONE);
        grpStatus.setText("Status");
        grpStatus.setBounds(10, 221, 431, 82);

        Label lblFilesCreated = new Label(grpStatus, SWT.NONE);
        lblFilesCreated.setBounds(10, 22, 92, 15);
        lblFilesCreated.setText("Files created:");

        statusTxt = new StyledText(grpStatus, SWT.WRAP);
        statusTxt.setText("Idle");
        statusTxt.setEditable(false);
        statusTxt.setBounds(10, 43, 411, 29);

        Label countFiles = new Label(grpStatus, SWT.NONE);
        countFiles.setBounds(108, 22, 55, 15);
        countFiles.setText("0");

        Label lblUser = new Label(shell, SWT.NONE);
        lblUser.setBounds(10, 113, 55, 15);
        lblUser.setText("User:");
```

```
                userTxt = new Text(shell, SWT.BORDER);
                userTxt.setBounds(10, 134, 350, 21);
        }
        ArrayList<String> files;
        protected void createMonthlyReports() {
        if (files == null)files = new ArrayList<>();
                files.clear();

        File file =new File(sourceFileTxt.getText());
Scanner in = null;
        try {
                in = new Scanner(file);
                String toFolderStr = toFolderTxt.getText().endsWith("\\")? toFolderTxt.getText() : toFolderTxt.getText()+"\\";
                 while(in.hasNext())
        {
        String line=in.nextLine();
        String separatorcomma = ",";
        String lineArray[] =  line.split(Pattern.quote(separatorcomma));

                try {
                        String fileNameStr =  getDateMonth(lineArray[0] )+ ".csv";
                        boolean exists = false;
                try {
                        exists = FilesUtil.checkIfExistsAndHeadersAndMdr( toFolderStr,fileNameStr, headers);
                        } catch (IOException e) {
                        e.printStackTrace();
```

```
                    continue;
            }
            if (exists) {
            try {
                    FilesUtil.appendNewLineToTextFile(toFolderStr + fileNameStr, line);
                    } catch (IOException e) {
                    e.printStackTrace();
                    }
            }else {
                    try {
                    files.add(toFolderStr+fileNameStr);
                    FilesUtil.writeToTextFile(toFolderStr+fileNameStr, headers + "\n");
                    FilesUtil.appendNewLineToTextFile(toFolderStr+fileNameStr, line);
                    } catch (IOException e) {
                    e.printStackTrace();
            }
}
} catch (ParseException e) {
}
}
 sortFiles(toFolderStr);
            } catch (FileNotFoundException e1) {
            statusTxt.setText("Error: FileNotFoundException");
            System.out.println("Error: FileNotFoundException " );
            e1.printStackTrace();
            return ;
}
```

```
}
        protected void createWeeklyReports() {
        if (files == null)files = new ArrayList<>();
        files.clear();
        File file =new File(sourceFileTxt.getText());
Scanner in = null;
        try {
        String toFolderStr = toFolderTxt.getText().endsWith("\\")? toFolderTxt.getText() : toFolderTxt.getText()+"\\";
        in = new Scanner(file);

        while(in.hasNext())
            {
        String line=in.nextLine();
        String separatorcomma = ",";
        String lineArray[] =  line.split(Pattern.quote(separatorcomma));

        try {
                String fileNameStr =  "w_"+getDateWeek(lineArray[0] ) + ".csv";
                boolean exists = false;
                try {
                exists = FilesUtil.checkIfExistsAndHeadersAndMdr( toFolderStr,fileNameStr, headers);
                } catch (IOException e) {
                continue;
        }
if (exists) {
        try {
                FilesUtil.appendNewLineToTextFile(toFolderStr + fileNameStr, line);
```

```
                } catch (IOException e) {

                }

        }else {

        try {

                files.add(toFolderStr+fileNameStr);

                FilesUtil.writeToTextFile(toFolderStr+fileNameStr, headers + "\n");

                FilesUtil.appendNewLineToTextFile(toFolderStr+fileNameStr, line);

                } catch (IOException e) {

        }

}

                } catch (ParseException e) {

                e.printStackTrace();

                }



        }

        sortFiles(toFolderStr);

        } catch (FileNotFoundException e1) {

        statusTxt.setText("Error: FileNotFoundException");

        e1.printStackTrace();

        return ;

        }

}



        public void sortFiles(String toFolderStr) {

        for(String filePath : files) {

        try {

                System.out.println("fileName: " +toFolderStr+ "sorted_" + filePath.replace(toFolderStr,""));
```

```
                    FilesUtil.writeToTextFile(toFolderStr+ "sorted_" + filePath.replace(toFolderStr,""), headers + "\n");
                    writeAllLinesToFile(toFolderStr+ "sorted_" + filePath.replace(toFolderStr,""),convertToRecords(readAllLinesFromFile(filePath)));
                    } catch (Exception e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                    }
            }
}
            public ArrayList<String> readAllLinesFromFile(String path) throws IOException{

            ArrayList<String> aList = new ArrayList<>();
            FileReader fileReader = new FileReader(path);
            BufferedReader bufferedReader = new BufferedReader(fileReader);
            String line = null;
            while( (line = bufferedReader.readLine()) != null){
            aList.add(line);
            }
            bufferedReader.close();
            return aList;
            }
            public ArrayList<Record> convertToRecords(ArrayList<String> recordsStr) {
            ArrayList<Record> records = new ArrayList<>();
            recordsStr:remove(0);
            for(String rec : recordsStr) {
            String[] parts = rec.split(",");
            Record r = new Record();
            r.date = parts[0];
```

```
r.user = parts[1];

r.pc = parts[2];

r.keyword = parts[3];

r.number = parts[4];

r.createTimestamp();

records.add(r);

}

Collections.sort(records, new Comparator<Record>() {

@Override

public int compare(Record o1, Record o2) {

return (o1.timestamp - o2.timestamp)>=0 ? 1 : -1;

}

});

return records;

}

public static void writeAllLinesToFile(String path, ArrayList<Record> records) throws IOException {

ArrayList<String> aList = new ArrayList<String>();

for (Record rec : records) {

String l = rec.date + ","+rec.user + ","+rec.pc+","+rec.keyword + "," + rec.number ;

aList.add(l);

}



FileWriter fileWriter = new FileWriter(path, true);

PrintWriter printWriter = new PrintWriter(fileWriter);

for (String line : aList){

printWriter.printf("%s" + "%n", line);

}
```

```
printWriter.close();

}

DateFormat dfin = new SimpleDateFormat("MM/dd/yyyy HH:mm:ss", Locale.ENGLISH);

DateFormat dfoutMonthly = new SimpleDateFormat("yyyy_MM", Locale.ENGLISH);

DateFormat dfoutWeekly = new SimpleDateFormat("yyyy_MM_WW", Locale.ENGLISH);

public String getDateMonth(String inputDate) throws ParseException {

String outDate="";



Date date =  dfin.parse(inputDate);

outDate = dfoutMonthly.format(date);

return outDate;

}

public String getDateWeek(String inputDate) throws ParseException {

String outDate="";

Date date =  dfin.parse(inputDate);

outDate = dfoutWeekly.format(date);

return outDate;

}



public void setFormFile(String input) {

try {

        sourceFileTxt.setText(input);

        }catch (Exception e) {

// TODO: handle exception

}

}

public void setToFolder(String input) {
```

```
            try {
                        toFolderTxt.setText(input);
                        }catch (Exception e) {
            // TODO: handle exception
            }
}
}
```

A.2 Κώδικας οπτικοποίησης σε Javascript με τη βιβλιοθήκη D3.js.

```
<!DOCTYPE html>
<meta charset="utf-8">
<html>
 <head>
  <style>
   rect.bordered {
    stroke: #E6E6E6;
    stroke-width:2px;
   }

   text.mono {
    font-size: 9pt;
    font-family: Consolas, courier;
    fill: #000;
```

```
  }


  text.axis-workweek {
   fill: #000;
  }


  text.axis-worktime {
   fill: #000;
  }
 </style>
 <script src="http://d3js.org/d3.v3.js"></script>
</head>
<body>
 <div id="chart"></div>
 <div id="dataset-picker">
 </div>
 <script type="text/javascript">
  var margin = { top: 20, right: 0, bottom: 0, left: 30 },
  width = 1400 - margin.left - margin.right,
  height = 610 - margin.top - margin.bottom,
  gridSize = Math.floor(width / 69),
  legendElementWidth = gridSize*2,
  buckets = 10,
  colors = ["#0d00b0","#ff0000","#888888","#ff0080","#34b600","#888888","#f0ff00","#ffa500","#4f3300","#888888"],
  days = ["1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30",
"31"],
  times = ["1a", "2a", "3a", "4a", "5a", "6a", "7a", "8a", "9a", "10a", "11a", "12a", "1p", "2p", "3p", "4p", "5p", "6p", "7p", "8p", "9p", "10p", "11p", "12p"];
```

```
datasets = ["*.csv"];


var svg = d3.select("#chart").append("svg")
  .attr("width", width + margin.left + margin.right)
  .attr("height", height + margin.top + margin.bottom)
  .append("g")
  .attr("transform", "translate(" + margin.left + "," + margin.top + ")");


var dayLabels = svg.selectAll(".dayLabel")
  .data(days)
  .enter().append("text")
   .text(function (d) { return d; })
   .attr("x", 0)
   .attr("y", function (d, i) { return i * gridSize; })
   .style("text-anchor", "end")
   .attr("transform", "translate(-6," + gridSize / 1.5 + ")")
   .attr("class", function (d, i) { return ((i >= 0 && i <= 4) ? "dayLabel mono axis axis-workweek" : "dayLabel mono axis"); });


var timeLabels = svg.selectAll(".timeLabel")
  .data(times)
  .enter().append("text")
   .text(function(d) { return d; })
   .attr("x", function(d, i) { return i * gridSize; })
   .attr("y", 0)
   .style("text-anchor", "middle")
   .attr("transform", "translate(" + gridSize / 2 + ", -6)")
   .attr("class", function(d, i) { return ((i >= 7 && i <= 16) ? "timeLabel mono axis axis-worktime" : "timeLabel mono axis"); });
```

```
var color = function(csvFile) {
 d3.csv(dataset[j]",
 function(d) {
  return {
   date: +d.date,
   user: +d.user,
   pc: +d.pc,
   day: +d.date.split(" ")[0].split("/")[1],
   hour: +d.date.split("/")[2].split(" ")[1].split(":")[0],
   keyword: +d.keyword,
   number: +d.number
  };
 },
 function(error, data) {
  var colorScale = d3.scale.quantile()
    .domain([0, buckets -1])
    .range(colors);

  var cards = svg.selectAll(".hour")
    .data(data, function(d) {return d.day+':'+d.hour;});

  cards.append("title");

  cards.enter().append("rect")
    .attr("x", function(d) { return (d.hour - 1) * gridSize; })
    .attr("y", function(d) { return (d.day - 1) * gridSize; })
```

```
    .attr("rx", 4)

    .attr("ry", 4)

    .attr("class", "hour bordered")

    .attr("width", gridSize)

    .attr("height", gridSize)

    .style("fill", colors[0]);


  cards.transition().duration(1000)

     .style("fill", function(d) { return colorScale(d.number); });


  cards.select("title").text(function(d) { return d.date+" : "+d.keyword; });


  cards.exit().remove();


 });

};


color(datasets[0]);


var html = d3.select("svg")

    .attr("version", 1.1)

    .attr("xmlns", "http://www.w3.org/2000/svg")

    .node().parentNode.innerHTML;


 var image = new Image();

 image.src = '' + window.btoa(html);
```

```
    image.onload = function() {
      var canvas = document.createElement("canvas");
      canvas.width = image.width;
      canvas.height = image.height;
      var context = canvas.getContext("2d");
      context.drawImage(image, 0, 0);

      function appendCSS(element) {
        var styleElement = document.createElement("style");
        styleElement.setAttribute("type");

      }

    var a = document.createElement("a");
    a.download = "image.png";
    a.href = canvas.toDataURL("image/png");
    document.body.appendChild(a);
    a.click();
   }
  </script>
 </body>
</html>
```

A.3 Κώδικας υλοποίησης και εκπαίδευσης CNN αλγορίθμου

A.3.1 Υλοποίησης CNN αλγορίθμου σε Python

```
import dataset

import tensorflow as tf

import time

from datetime import timedelta

import math

import random

import numpy as np



#Adding Seed so that random initialization is consistent

from numpy.random import seed

seed(1)

from tensorflow import set_random_seed

set_random_seed(2)



batch_size = 32



#Prepare input data

classes = ['good','bad']

num_classes = len(classes)



# 20% of the data will automatically be used for validation
```

```
validation_size = 0.2

img_size = 128

num_channels = 3

train_path='training_data'



# We shall load all the training and validation images and labels into memory using openCV and use that during training

data = dataset.read_train_sets(train_path, img_size, classes, validation_size=validation_size)



print("Complete reading input data. Will Now print a snippet of it")

print("Number of files in Training-set:\t{}".format(len(data.train.labels)))

print("Number of files in Validation-set:\t{}".format(len(data.valid.labels)))



session = tf.Session()



x = tf.placeholder(tf.float32, shape=[None, img_size,img_size,num_channels], name='x')



## labels

y_true = tf.placeholder(tf.float32, shape=[None, num_classes], name='y_true')

y_true_cls = tf.argmax(y_true, dimension=1)



##Network graph params

filter_size_conv1 = 3

num_filters_conv1 = 32



filter_size_conv2 = 3

num_filters_conv2 = 32
```

```
filter_size_conv3 = 3

num_filters_conv3 = 64



fc_layer_size = 128



def create_weights(shape):

return tf.Variable(tf.truncated_normal(shape, stddev=0.05), name='weights')



def create_biases(size):

#with tf.name_scope("biases"):

return tf.Variable(tf.constant(0.05, shape=[size]), name='biases')



def create_convolutional_layer(input,

num_input_channels,

conv_filter_size,

num_filters, name='conv'):

with tf.name_scope(name):

        ## We shall define the weights that will be trained using create_weights function.

        weights = create_weights(shape=[conv_filter_size, conv_filter_size, num_input_channels, num_filters])

        ## We create biases using the create_biases function. These are also trained.

        biases = create_biases(num_filters)



## Creating the convolutional layer

layer = tf.nn.conv2d(input=input,

filter=weights,

strides=[1, 1, 1, 1],

padding='SAME')
```

```
    layer += biases

    ## We shall be using max-pooling.
    layer = tf.nn.max_pool(value=layer,
                           ksize=[1, 2, 2, 1],
                           strides=[1, 2, 2, 1],
                           padding='SAME')
    ## Output of pooling is fed to Relu which is the activation function for us.
    layer = tf.nn.relu(layer)
    return layer

def create_flatten_layer(layer):
    #We know that the shape of the layer will be [batch_size img_size img_size num_channels]
    # But let's get it from the previous layer.
    layer_shape = layer.get_shape()

    ## Number of features will be img_height * img_width* num_channels. But we shall calculate it in place of hard-coding it.
    num_features = layer_shape[1:4].num_elements()

    ## Now, we Flatten the layer so we shall have to reshape to num_features
    layer = tf.reshape(layer, [-1, num_features])

    return layer

def create_fc_layer(input,
                    num_inputs,
```

```
        num_outputs,
        use_relu=True, name='fc'):
    with tf.name_scope(name):
            #Let's define trainable weights and biases.
            weights = create_weights(shape=[num_inputs, num_outputs])
            biases = create_biases(num_outputs)

    # Fully connected layer takes input x and produces wx+b.Since, these are matrices, we use matmul function in Tensorflow
    layer = tf.matmul(input, weights) + biases
    if use_relu:
      layer = tf.nn.relu(layer)

    return layer

layer_conv1 = create_convolutional_layer(input=x,
        num_input_channels=num_channels,
        conv_filter_size=filter_size_conv1,
        num_filters=num_filters_conv1)
layer_conv2 = create_convolutional_layer(input=layer_conv1,
        num_input_channels=num_filters_conv1,
        conv_filter_size=filter_size_conv2,
        num_filters=num_filters_conv2)

layer_conv3= create_convolutional_layer(input=layer_conv2,
        num_input_channels=num_filters_conv2,
        conv_filter_size=filter_size_conv3,
        num_filters=num_filters_conv3)
```

```
layer_flat = create_flatten_layer(layer_conv3)


layer_fc1 = create_fc_layer(input=layer_flat,
          num_inputs=layer_flat.get_shape()[1:4].num_elements(),
          num_outputs=fc_layer_size,
          use_relu=True)


layer_fc2 = create_fc_layer(input=layer_fc1,
          num_inputs=fc_layer_size,
          num_outputs=num_classes,
          use_relu=False)


with tf.variable_scope("softmax"):
          y_pred = tf.nn.softmax(layer_fc2)
          y_pred_cls = tf.argmax(y_pred, dimension=1)


session.run(tf.initialize_all_variables())


with tf.name_scope("cross_ent"):
          cross_entropy = tf.nn.softmax_cross_entropy_with_logits(logits=layer_fc2,
                    labels=y_true)
          cost = tf.reduce_mean(cross_entropy)


with tf.name_scope("train"):
          optimizer = tf.train.AdamOptimizer(learning_rate=1e-4).minimize(cost)
```

```
with tf.name_scope("accuracy") as scope:
        correct_prediction = tf.equal(y_pred_cls,y_true_cls)
        accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))

# create a summary for our cost and accuracy
tf.scalar_summary("Cost", cost)
training_acc_summary = tf.scalar_summary("Training Accuracy", accuracy)
validation_acc_summary = tf.scalar_summary("Validation Accuracy", accuracy)

# merge all summaries into a single "operation" which we can execute in a session
merged_summary_op = tf.merge_all_summaries()

init=session.run(tf.initialize_all_variables())

# Set the logs writer to the folder /Tensorboard/logs
summary_writer = tf.train.SummaryWriter('Tensorboard/logs', graph=tf.get_default_graph())

def show_progress(epoch, feed_dict_train, feed_dict_validate, val_loss):
  acc = session.run(accuracy, feed_dict=feed_dict_train)
  val_acc = session.run(accuracy, feed_dict=feed_dict_validate)
  msg = "Training Epoch {0} --- Training Accuracy: {1:>6.1%}, Validation Accuracy: {2:>6.1%},  Validation Loss: {3:.3f}"
  print(msg.format(epoch + 1, acc, val_acc, val_loss))

total_iterations = 0

saver = tf.train.Saver()
```

```
def train(num_iteration):
  global total_iterations

  for i in range(total_iterations,
            total_iterations + num_iteration):

    x_batch, y_true_batch, _, cls_batch = data.train.next_batch(batch_size)
    x_valid_batch, y_valid_batch, _, valid_cls_batch = data.valid.next_batch(batch_size)

    feed_dict_tr = {x: x_batch,
                y_true: y_true_batch}
    feed_dict_val = {x: x_valid_batch,
                 y_true: y_valid_batch}

    session.run(optimizer, feed_dict=feed_dict_tr)

    # Write logs for each iteration
    summary_str = session.run(merged_summary_op, feed_dict={x: x_batch, y_true: y_true_batch})
    summary_writer.add_summary(summary_str, i)
    training_summary_str = session.run(training_acc_summary, feed_dict={x: x_batch, y_true: y_true_batch})
    summary_writer.add_summary(training_summary_str, i)
    validation_summary_str = session.run(validation_acc_summary, feed_dict={x: x_valid_batch, y_true: y_valid_batch})
    summary_writer.add_summary(validation_summary_str, i)

    if i % int(data.train.num_examples/batch_size) == 0:
      val_loss = session.run(cost, feed_dict=feed_dict_val)
      epoch = int(i / int(data.train.num_examples/batch_size))
```

```
        show_progress(epoch, feed_dict_tr, feed_dict_val, val_loss)
        saver.save(session, 'good-bad-model')

    total_iterations += num_iteration

train(num_iteration=3000)
```

A.3.2 Υλοποίησης προγράμματος εισαγωγής εικόνων σε Python

```
import cv2
import os
import glob
from sklearn.utils import shuffle
import numpy as np

def load_train(train_path, image_size, classes):
    images = []
    labels = []
    img_names = []
    cls = []

    print('Going to read training images')
    for fields in classes:
        index = classes.index(fields)
        print('Now going to read {} files (Index: {})'.format(fields, index))
        path = os.path.join(train_path, fields, '*g')
```

```
    files = glob.glob(path)
    for fl in files:
        image = cv2.imread(fl)
        image = cv2.resize(image, (image_size, image_size),0,0, cv2.INTER_LINEAR)
        image = image.astype(np.float32)
        image = np.multiply(image, 1.0 / 255.0)
        images.append(image)
        label = np.zeros(len(classes))
        label[index] = 1.0
        labels.append(label)
        flbase = os.path.basename(fl)
        img_names.append(flbase)
        cls.append(fields)
  images = np.array(images)
  labels = np.array(labels)
  img_names = np.array(img_names)
  cls = np.array(cls)

  return images, labels, img_names, cls

class DataSet(object):

 def __init__(self, images, labels, img_names, cls):
  self._num_examples = images.shape[0]

  self._images = images
  self._labels = labels
```

```
    self._img_names = img_names
    self._cls = cls
    self._epochs_done = 0
    self._index_in_epoch = 0

  @property
  def images(self):
    return self._images

  @property
  def labels(self):
    return self._labels

  @property
  def img_names(self):
    return self._img_names

  @property
  def cls(self):
    return self._cls

  @property
  def num_examples(self):
    return self._num_examples

  @property
  def epochs_done(self):
```

```
    return self._epochs_done

  def next_batch(self, batch_size):
    """Return the next `batch_size` examples from this data set."""
    start = self._index_in_epoch
    self._index_in_epoch += batch_size

    if self._index_in_epoch > self._num_examples:
      # After each epoch we update this
      self._epochs_done += 1
      start = 0
      self._index_in_epoch = batch_size
      assert batch_size <= self._num_examples
    end = self._index_in_epoch

    return self._images[start:end], self._labels[start:end], self._img_names[start:end], self._cls[start:end]

def read_train_sets(train_path, image_size, classes, validation_size):
  class DataSets(object):
    pass
  data_sets = DataSets()

  images, labels, img_names, cls = load_train(train_path, image_size, classes)
  images, labels, img_names, cls = shuffle(images, labels, img_names, cls)

  if isinstance(validation_size, float):
    validation_size = int(validation_size * images.shape[0])
```

```
validation_images = images[:validation_size]

validation_labels = labels[:validation_size]

validation_img_names = img_names[:validation_size]

validation_cls = cls[:validation_size]



train_images = images[validation_size:]

train_labels = labels[validation_size:]

train_img_names = img_names[validation_size:]

train_cls = cls[validation_size:]



data_sets.train = DataSet(train_images, train_labels, train_img_names, train_cls)

data_sets.valid = DataSet(validation_images, validation_labels, validation_img_names, validation_cls)



return data_sets
```

A.3.3 Υλοποίησης προγράμματος πρόβλεψης δραστηριότητας χρήστη σε Python

```
import tensorflow as tf

import numpy as np

import os,glob,cv2

import sys,argparse



# First, pass the path of the image

dir_path = os.path.dirname(os.path.realpath(__file__))

image_path=sys.argv[1]

filename = dir_path +'/' +image_path
```

```
image_size=128

num_channels=3

images = []

# Reading the image using OpenCV

image = cv2.imread(filename)

# Resizing the image to our desired size and preprocessing will be done exactly as done during training

image = cv2.resize(image, (image_size, image_size),0,0, cv2.INTER_LINEAR)

images.append(image)

images = np.array(images, dtype=np.uint8)

images = images.astype('float32')

images = np.multiply(images, 1.0/255.0)

#The input to the network is of shape [None image_size image_size num_channels]. Hence we reshape.

x_batch = images.reshape(1, image_size,image_size,num_channels)



## Let us restore the saved model

sess = tf.Session()

# Step-1: Recreate the network graph. At this step only graph is created.

saver = tf.train.import_meta_graph('good-bad-model.meta')

# Step-2: Now let's load the weights saved using the restore method.

saver.restore(sess, tf.train.latest_checkpoint('./'))



# Accessing the default graph which we have restored

graph = tf.get_default_graph()



# Now, let's get hold of the op that we can be processed to get the output.

# In the original network y_pred is the tensor that is the prediction of the network

y_pred = graph.get_tensor_by_name("y_pred:0")
```

```
## Let's feed the images to the input placeholders

x= graph.get_tensor_by_name("x:0")

y_true = graph.get_tensor_by_name("y_true:0")

y_test_images = np.zeros((1, 2))



### Creating the feed_dict that is required to be fed to calculate y_pred

feed_dict_testing = {x: x_batch, y_true: y_test_images}

result=sess.run(y_pred, feed_dict=feed_dict_testing)

# result is of this format [probabiliy_of_good_probability_of_bad_user]

print(result)
```