Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή
Στην Ασφάλεια Υπολογιστών και Δικτύων

Αντιμετώπιση Εσωτερικών Απειλών Με Χρήση Τεχνητής Νοημοσύνης

Νικόλαος Πελτέκης

Επιβλέπων Καθηγητής
Σταύρος Σιαηλής

Μάιος 2019

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Αντιμετώπιση Εσωτερικών Απειλών με χρήση Τεχνητής Νοημοσύνης

Νικόλαος Πελτέκης

Επιβλέπων Καθηγητής
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 2019

Περίληψη

Ένας οργανισμός κατέχει ένα πλήθος αγαθών τα οποία φροντίζει να προστατεύει από φθορά. Υπάρχουν διάφορες αιτίες οι οποίες οδηγούν σε αυτό το γεγονός και μία εξ' αυτών είναι οι εσωτερικές απειλές: υπάλληλοι, μέλη ή συνεργάτες του οργανισμού (ή αλλιώς εσωτερικοί χρήστες) οι οποίοι έχουν πρόσβαση σε αυτά, να προσπαθήσουν να εκμεταλλευτούν αδυναμίες του Πληροφοριακού Συστήματος του οργανισμού και να προξενήσουν ζημιά στα αγαθά.

Η αντιμετώπιση εσωτερικών απειλών, γίνεται είτε με τη χρήση πολύπλοκων συστημάτων ή με πολύ περιοριστικές πολιτικές χρήσης ή δεν γίνεται καθόλου. Αρωγός σε αυτή την προσπάθεια μπορεί να είναι η Τεχνητή Νοημοσύνη (Deep/Machine Learning – DL/ML) και πιο συγκεκριμένα η χρήση Νευρωνικών δικτύων (NN, CNN, DNN, RNN). Το κλειδί σε αυτή τη στρατηγική είναι η χρήση συγκεκριμένων αλγορίθμων οι οποίοι, αφού εκπαιδευτούν κατάλληλα, θα χρησιμοποιηθούν ώστε να εξαχθούν συμπεράσματα σχετικά για την ύπαρξη ή όχι εσωτερικής απειλής.

Στόχος της παρούσας μεταπτυχιακής διατριβής είναι η υλοποίηση και δοκιμή τριών (3) πολύ γνωστών αλγορίθμων και δοκιμή αυτών με συγκεκριμένο dataset (Cert) για την εξαγωγή συμπερασμάτων αποτελεσματικότητας ώστε να αναγνωριστούν πιθανές εσωτερικές απειλές, καθώς και η εύρεση ευπαθειών στα συστήματα του οργανισμού. Το dataset περιέχει log files από ένα Πληροφοριακό Σύστημα και το καθένα παρέχεται σε μορφή csv, τα οποία αφού υποστούν κατάλληλη επεξεργασία, εισάγονται στο DL/ML σύστημα μας (στο Azure ML Studio) με σκοπό την επεξεργασία και ανάλυση με τη χρήση των Linear Regression, One-class Vector και PCA αλγορίθμων.

Summary

An organization owns a number of assets that needs and owes to protect. Assets are vulnerable to multiple kinds of exposure and one of them is Insider Threats: employees, vendors, etc, who have access, will try to exploit vulnerabilities of the Information System, in order to gain access to the assets.

Insider Threat Detection can be performed using complicated and resourceful systems or using well documented information policies or not at all. Artificial Intelligence with Machine/Deep Learning are providing help on this with the use advanced algorithms, and help the IT-Security analysts discover such threats in an easier manner.

This paper tries to combine the use of three well know algorithms, Linear Regression, One-Class Vector and PCA, in a specific, artificial dataset from Cert, in order to help to confirm possible insider threats. Also, Azure ML Studio is used because it provides an easy and visual way of conducting experiments and producing reports.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω όλους τους Καθηγητές και ιδιαίτερα τον κ. Σταύρο Σιαηλή για την γενική του επίβλεψη σε αυτήν την εργασία. Τέλος, θερμά ευχαριστώ τη σύζυγό μου Ελένη και τον γιο μου Άκη για την στήριξη και την συμπαράσταση τους.

Περιεχόμενα

Συντομογραφίες

Συντομογραφίες	Ορισμοί
AI	Artificial Intelligence
DL	Deep Learning
ML	Machine Learning
NN	Neural Network
DNN	Deep Neural Network
RNN	Recurrent Neural Network
CNN	Convolution Neural Network
ΠΣ	Πληροφοριακό Σύστημα
ΤΝ	Τεχνητή Νοημοσύνη
ΝΔ	Νευρωνικό Δίκτυο
SVM	Support Vector Machine
OC-SVM	One-class Support Vector Machine
PCA	Principal Analysis
PC	Principal Components
ONNX	Open Neural Network Exchange

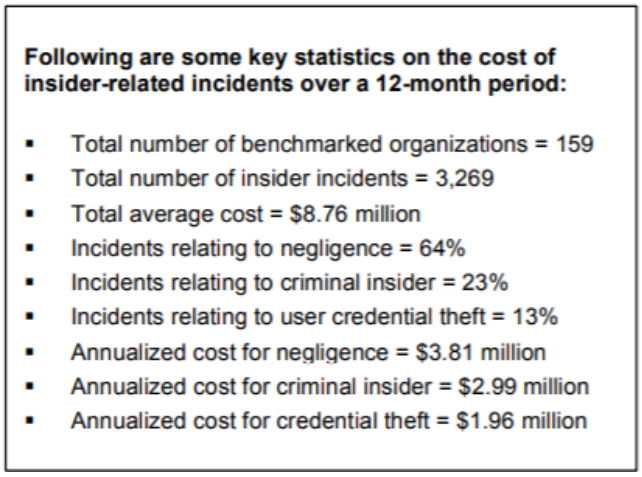
Κεφάλαιο 1
Εισαγωγή

1.1 Ασφάλεια Πληροφοριών

Η τεχνολογία των πληροφοριών είναι ζωτικής σημασίας για τους οργανισμούς και την κοινωνία. Η διακίνηση πληροφοριών, σήμερα είναι πιο εύκολη και σε αυτό συμβάλει το διαδίκτυο και οι δυνατότητες που προσφέρει. Κάθε οργανισμός το χρησιμοποιεί για να παρέχει τις υπηρεσίες του ή για να συλλέξει πληροφορίες. Η συλλογή τους είναι καθημερινότητα πλέον γι' αυτό και χρησιμοποιούνται συστήματα μεγάλης υπολογιστικής ισχύος και αποθηκευτικού χώρου, για την διαχείριση και επεξεργασία τους. Οι πληροφορίες αυτές, κατάλληλα επεξεργασμένες, αποτελούν αγαθά για τον οργανισμό, γεγονός που τους δίνει και προστιθέμενη αξία.

Ένας οργανισμός φροντίζει να προστατεύει τα αγαθά του, τους ιδιοκτήτες και το προσωπικό του. Ένα αγαθό φθείρεται όταν υλοποιείται κάποια απειλή και με τον όρο εσωτερική απειλή, περιγράφουμε τη φθορά του ΠΣ εκ των έσω, είτε λόγω κακόβουλης χρήσης είτε λόγω αμέλειας (misuse). Στη κατηγορία κακόβουλης χρήσης εμπεριέχονται οι εσωτερικοί χρήστες που έχουν σαν σκοπό τη πρόκληση βλάβης στο εταιρικό δίκτυο ή την εσκεμμένη κλοπή πληροφοριών από αυτό.

Στατιστικές και έρευνες ανά τον κόσμο, με ερωτηθέντες κυρίως διαχειριστές συστημάτων και υπεύθυνων ασφαλείας, καταλήγουν ότι οι εσωτερικές απειλές υλοποιούνται σε 64% λόγω αμέλειας, 23% λόγω εγκληματικής συμπεριφοράς. [1]



Εικόνα 1: Στατιστικά εσωτερικών απειλών. Ponemon Institute

Αρωγός στην προσπάθεια αναγνώρισης των εσωτερικών απειλών είναι η ΤΝ. Ένα νευρωνικό δίκτυο, με τα κατάλληλα δεδομένα, μπορεί να εξάγει συμπεράσματα και να δώσει το έναυσμα για εις βάθος έρευνα συγκεκριμένων περιπτώσεων.

1.2 Ερευνητικά Ερωτήματα

Η πολιτική ασφαλείας ενός οργανισμού μπορεί να είναι περιοριστική ή μη, και δημιουργεί το κατάλληλο υπόβαθρο ώστε ο οργανισμός να λάβει τα απαραίτητα μέτρα για την αυτοπροστασία του. Σε αυτήν την εργασία, τα ερωτήματα που τίθενται είναι:

1. Μπορεί μια πολιτική ασφάλειας, να εκφρασθεί προγραμματιστικά ώστε να χρησιμοποιηθεί σε ένα μοντέλο ΤΝ για την ανίχνευση εσωτερικών απειλών;

2. Επιτυγχάνει το μοντέλο ΤΝ να ανιχνεύσει τις εσωτερικές απειλές, σύμφωνα με την υλοποιημένη πολιτική ασφάλειας;

3. Μεταβάλλοντας την πολιτική ασφάλειας ώστε να γίνει πιο περιοριστική (GDPR, PCI-DSS), ανιχνεύονται περισσότερες εσωτερικές απειλές;

4. Υπάρχει τρόπος να ανιχνευθούν και άλλες ευπάθειες στο ΠΣ, ως αποτέλεσμα της εντατικής ανίχνευσης;

Στη συνέχεια στο Κεφάλαιο 2 γίνεται μια ανασκόπηση των σχετικών μελετών έως τώρα, ενώ στο Κεφάλαιο 3 γίνεται μια αναφορά στα συστήματα ΤΝ και τους αλγόριθμους που χρησιμοποιήθηκαν. Στο Κεφάλαιο 4 γίνεται αναφορά στην μεθοδολογία που ακολουθήθηκε και στο Κεφάλαιο 5 παρουσιάζονται τα αποτελέσματα και τα συμπεράσματα της μελέτης.

Κεφάλαιο 2
Ανασκόπηση Βιβλιογραφίας

2.1 Γενικά

Οι εσωτερικές απειλές ενάντια στα Πληροφοριακά Συστήματα υπήρχαν ανέκαθεν και, μάλιστα, συνυπήρχαν από τις πρώτες στιγμές των Πληροφοριακών Συστημάτων. Σε όλο το ιστορικό βάθος της εποχής της Πληροφορικής, η αναγνώριση τους και η αντιμετώπιση τους δεν ήταν έγκυρη και ταυτόχρονη, δηλαδή σε πολλές περιπτώσεις γίνονταν αντιληπτές μήνες μετά την υλοποίηση τους, ενώ τα μέτρα αντιμετώπισης, μετά από ένα τόσο μεγάλο χρονικό διάστημα, δεν είχαν κανένα νόημα να εφαρμοστούν [2]. Ερευνητές σε πολλά ακαδημαϊκά ιδρύματα αλλά και εκτός αυτών, συμβάλουν με τις έρευνες στους στην εύρεση τρόπων αναγνώρισης εσωτερικών απειλών, έγκαιρα, με τη χρήση Machine Learning / Deep Learning.

2.2 Προηγούμενες Μελέτες

Ο Burges [3], περιγράφει πως μπορεί να χρησιμοποιηθεί ο αλγόριθμος Support Vector Machine (SVM) στην ανίχνευση μοτίβων. Στη μελέτη του περιγράφει τον τρόπο με τον οποίο γίνεται αυτό, καταγράφοντας τις αποδείξεις με πλήρη τρόπο ώστε να μπορεί να χρησιμοποιηθεί ως εκπαιδευτικό υλικό. Η ανίχνευση μοτίβων περιλαμβάνει αναγνώριση γραμμάτων και γραφικού χαρακτήρα, προσώπων, φωνής, αντικειμένων, κατηγοριοποίηση κειμένου, κ.α. Ο Burges αναλύει τι είναι το VC-Dimension, περιγράφει τη χρήση του SVM για δεδομένα που μπορούν να διαχωριστούν ή όχι, παρουσιάζει αναλογικά, πότε τα αποτελέσματα είναι μοναδικά και πότε καθολικά. Επίσης, περιγράφει τι είναι το kernel-mapping, το οποίο χρησιμοποιείται όταν οι προτεινόμενες λύσεις δεν είναι γραμμικές με τα δεδομένα. Τέλος, περιγράφει ότι ο SVM μπορεί να έχει μεγάλη τιμή VC-dimension και έτσι δημιουργείται εμπόδιο στη γενίκευση. Τέλος, παρουσιάζει περιπτώσεις όπου ο SVM έχει καλή απόδοση.

Οι Breier, Branisova [4] χρησιμοποιούν τεχνικές Data Mining για την ανάλυση log files με τη χρήση ενός συστήματος Apache Handoop με το MapReduce Framework, το οποίο είναι διαμορφωμένο για παράλληλη επεξεργασία μεγάλου όγκου δεδομένων. Προσπαθούν να ανιχνεύσουν ανωμαλίες και διαρροές, στα δεδομένα των αρχείων καταγραφών, βασισμένοι στη δημιουργία κανόνων με δυναμικό τρόπο [5]. Αυτό, επιτρέπει την ανίχνευση νέων ανωμαλιών ή/και διαρροών με ελάχιστη ανθρώπινη παρέμβαση στην ενημέρωση των κανόνων. Η υλοποίηση του MapReduce σε cluster ενός κόμβου, τους επέτρεψε να εφαρμόσουν έναν αλγόριθμο ώστε να ελαττώσουν το χρόνο επεξεργασίας σε σύγκριση με τον βασικό αλγόριθμο βασισμένο σε δομή δέντρων.

Οι Tuor, Kaplan και Hutschinson [6] αναφέρονται στη δημιουργία ενός συστήματος αναγνώρισης απειλών που στηρίζεται στη σειρά από γεγονότα που έχουν σχέση με τις ενέργειες του χρήστη. Το σύστημα τους είναι μη εποπτευόμενο και φιλτράρει τα δεδομένα από αρχεία καταγραφών, τα οποία παρέχονται σε πραγματικό χρόνο, ως ροές. Αντί να δημιουργήσουν ένα μοντέλο όπου καταγράφεται η συμπεριφορά μιας ή περισσοτέρων απειλών, χρησιμοποιούν DNN και RNN τα οποία καταγράφουν τις ενέργειες του χρήστη και αποφασίζουν αν είναι κανονικές ή παρουσιάζουν κάποια ανωμαλία. Ως ένα επιπλέον μέτρο, χρησιμοποιούν τους ρόλους των χρηστών ώστε να διαπιστωθεί η κανονικότητα των ενεργειών μιας ομάδας και συμπερασματικά ενός χρήστη που ανήκει σ' αυτήν.

Οι Yuan, Can, Shang, Liu και Fang [7], στην εργασία τους υλοποιούν ένα DNN το οποίο μαθαίνει τη συμπεριφορά του χρήστη. Χρησιμοποιείται ένα ενδιάμεσο σύστημα Long Short Team Memory (LSTM) για αυτή τη λειτουργία και εξάγει vectors, τα οποία μετατρέπονται σε Πίνακες σταθερού μεγέθους, που τροφοδοτούν ένα CNN. Σκοπός τους δεν είναι να κατηγοριοποιήσουν τις ενέργειες ενός χρήστη ανά ημέρα και μετά να αποφασιστεί αν υπάρχει εσωτερική απειλή, αλλά να προβλέψουν την κίνηση της επόμενης ημέρας του χρήστη, ώστε να κατηγοριοποιηθεί ως normal ή anomaly.

Οι Liu, Ting και Zhou [8], μελετούν ανωμαλίες στα δεδομένα έχοντας υπόψη ότι αυτού του είδους τα δεδομένα έχουν διαφορετικές ιδιότητες, ενώ είναι λιγότερες από τα κανονικά δεδομένα. Αυτή η ιδιαιτερότητα, τους επιτρέπει να ανιχνευθούν με τη μέθοδο της απομόνωσης. Το σύστημα τους, δημιουργεί με τα δεδομένα, ένα σύνολο από iTrees (binary trees) και βασίζεται στο φαινόμενο ότι τα κανονικά δεδομένα θα έχουν μεγάλο μονοπάτι ενώ τα ανώμαλα θα είναι κοντά στη ρίζα. Υλοποιούν την δημιουργία μιας ομάδας δέντρων, iForest, την οποία μπορούν να επιτύχουν ώστε να έχει μεγάλη ακρίβεια στην απόδοση ενώ το πλήθος τους να είναι μικρό.

Οι Legg, Buckley, et al [9] παρουσιάζουν ένα σύστημα που κατασκευάζει το προφίλ του κάθε χρήστη από τις ενέργειες του, σε μορφή δέντρου, έτσι ώστε να είναι ευκολότερη η σύγκριση τους με άλλους χρήστες του ίδιου ρόλου. Στη συνέχεια, με βάση τη βαθμολόγηση των ενεργειών από κάποια ήδη προϋπάρχοντα metrics, τα οποία έχουν δημιουργηθεί από την ανάλυση των δεδομένων που συλλέγονται, το σύστημα επισημαίνει του χρήστες που έχουν ξεπεράσει ένα κατώφλι και τους συγκρίνει με παρόμοιους, ώστε να διαπιστώσει αν πρόκειται για ανωμαλία και πιθανή εσωτερική απειλή. Επίσης, παρέχεται κάποιο UI για του υπεύθυνους ασφαλείας, οι οποίοι θα επισημάνουν μια ειδοποίηση σαν αληθή ή ψευδή και το σύστημα θα συλλέξει αυτήν την πληροφορία και θα μεταβάλλει το μοντέλο.

Οι Lo, Buchanan, Griffiths και Macfarlane [10], δούλεψαν στο Cert r4.2 και σύγκριναν αλγόριθμους υπολογισμού απόστασης, αναλύοντας μόνο τη δραστηριότητα του χρήστη. Δεν λαμβάνεται υπόψη η προσωπικότητα του, το περιεχόμενο της δραστηριότητας ή η κατηγορία του. Μόνο τα δεδομένα που έχουν σχέση με σύνδεση/αποσύνδεση στο σύστημα, σύνδεση/αποσύνδεση συσκευής, αποστολή email και εργασίες αρχείων. Συγκρίνουν τη μέθοδο τους με άλλες εργασίες που έχουν υλοποιηθεί με τη χρήση Markov Chains, υλοποιώντας και αυτοί με τη σειρά τους Hidden Markov Chains, Damerau-Levenshtein Distance, Jaccard Distance και Cosine Distance. Το μοντέλο τους χτίζεται στηριζόμενο στους ήδη καταγεγραμμένους κακόβουλους χρήστες οι οποίοι παρέχονται μαζί με τα αρχεία του dataset.

Οι Malhotra, Ramakrishnan, Anand, Vig, Agarwal, Shroff [11], δημιουργούν ένα Long Short-Term Memory με σκοπό την ανίχνευση ανωμαλιών σε χρονοσειρές δεδομένων. Χρησιμοποιούν ένα κωδικοποιητή ο οποίος μαθαίνει να αναπαριστά διανύσματα με τα δεδομένα της εισόδου ενώ ο αποκωδικοποιητής χρησιμοποιεί αυτήν την αναπαράσταση για να ανακατασκευάσει τα αρχικά δεδομένα. Το LSTM γνωρίζει πως να ανακατασκευάζει τα normal δεδομένα, οπότε σε περίπτωση λάθους, αναγνωρίζονται ανωμαλίες.

2.3 Συμβολή της Παρούσας Μεταπτυχιακής Διατριβής

Σε προϋπάρχουσες μελέτες, χρησιμοποιούνταν διάφορα dataset δεδομένων, είτε στο σύνολο τους, είτε μέρος αυτών και υλοποιήθηκαν μοντέλα τα οποία επεξεργάζονταν τα δεδομένα, ενώ χρησιμοποιήθηκαν διάφοροι αλγόριθμοί και μεθοδολογίες (anomaly detection με SVM [3], με κανόνες [5], με DNN/RNN [6], χρησιμοποιώντας Isolation Forests [8], πρόβλεψη επόμενης ενέργειας [7], ανίχνευση ανωμαλιών στη συμπεριφοράς των χρηστών [10]), χωρίς να αναφέρονται στην υλοποίηση κάποιας πολιτικής ασφάλειας ή θεωρώντας την δεδομένη. Σε γενικές γραμμές, λόγω του γεγονότος ότι τα σενάρια εσωτερικών απειλών ήταν ήδη γνωστά, οι έρευνες μπορεί να κατευθύνονταν από τα σενάρια. Προχωρώντας τον συλλογισμό μας ένα ακόμα βήμα, τα περισσότερα συστήματα ΤΝ προσπαθούν να ανιχνεύσουν εσωτερικές απειλές γενικευμένα, στηριζόμενα στην ανίχνευση ανωμαλιών στις ενέργειες των χρηστών ή συγκρίνοντας τους με τους αντίστοιχους χρήστες που έχουν τον ίδιο ρόλο.

Η παρούσα εργασία, εξετάζει κατά πόσο είναι δυνατόν, οι πολιτικές ασφάλειας που καταγράφονται από τους οργανισμούς με σκοπό την προστασία των αγαθών, μπορούν να υλοποιηθούν από κάποιο σύστημα ανίχνευσης, ώστε να είναι χρηστικές και να φέρουν αποτελέσματα, παρέχοντας τη δυνατότητα στο αντίστοιχο τμήμα ασφαλείας του οργανισμού να επιβεβαιώσει μια πιθανή εσωτερική απειλή.

Ποιο συγκεκριμένα, θα εξετασθεί μια εφαρμοσμένη πολιτική υλοποιημένη προγραμματιστικά, σε αντιπαραβολή με γνωστό dataset που περιέχει στοιχεία από log files με ενέργειες χρηστών, ώστε να διαπιστωθεί αν υπάρχει η ικανότητα το σύστημα να ανιχνεύσει πιθανές εσωτερικές απειλές. Στη συνέχεια, θα προγραμματίσουμε μια πιο περιοριστική εσωτερική πολιτική (PCI-DSS) και θα επαναλάβουμε το πείραμα. Στο τέλος θα συγκρίνουμε τα αποτελέσματα των δύο πειραμάτων και να αποφανθούμε κατά πόσο οι περιοριστικές πολιτικές ασφαλείας είναι χρήσιμες στην ανίχνευση εσωτερικών απειλών.

Στην εργασία αυτή, θα χρησιμοποιηθούν οι αλγόριθμοι ανίχνευσης ανωμαλιών, Support Vector Machine και PCA, και θα εξεταστούν στο Cert r6.2 dataset σύμφωνα με τις περιγραφές των πολιτικών ασφαλείας.

Κεφάλαιο 3
Τεχνητή Νοημοσύνη

3.1 Εισαγωγή

«Τι είναι η τεχνητή νοημοσύνη;» Υπάρχουν ορισμοί οι οποίοι εκφράστηκαν σε διάφορες εποχές αλλά ο ορισμός των Rich and Knight (1991) «Τεχνητή Νοημοσύνη είναι η μελέτη του πώς να κάνουμε τον υπολογιστή να πράξει κάτι που επί του παρόντος ο άνθρωπος μπορεί να πράξει καλύτερα» είναι αρκετά χειροπιαστός και δυναμικός, ενώ μπορούμε να εκτιμήσουμε ότι θα διατηρηθεί σε βάθος χρόνου, γιατί οι ανθρώπινες ικανότητες υπερέχουν του υπολογιστή και τείνουν να μεταβάλλονται με το χρόνο.[12]

Είναι αναμενόμενο, ότι η εξέλιξη της ΤΝ σήμερα, στηρίζεται στα αποτελέσματα που προήλθαν από την παρατήρηση της φύσης, και ανάμεσα σε αυτά είναι η λειτουργία των νευρωνικών δικτύων. Ο άνθρωπος αντέγραψε τα βιολογικά νευρωνικά δίκτυα των ζώντων οργανισμών και προσπάθησε να φτιάξει τα δικά του. Το επόμενο ερώτημα που πρέπει να απαντηθεί λοιπόν, είναι «Τι είναι τα νευρωνικά δίκτυα;» και πιο συγκεκριμένα, «τί είναι τα τεχνητά νευρωνικά δίκτυα;». Είναι υπολογιστικές δομές οι οποίες, όπως τα αντίστοιχα βιολογικά, μπορούν να επεξεργαστούν πληροφορίες οπτικές, ακουστικές, κ.α. παίρνοντας γνώση μέσα από εξάσκηση και εμπειρία ενώ η κύρια διαφορά τους είναι ότι ακολουθούν προκαθορισμένους κανόνες.[13]

3.2 Τα πρώτα βήματα

Ο άνθρωπος είναι ον το οποίο μιμείται για να αποκτήσει γνώση μέσα από τη διαδικασία trial-error, η οποία λαμβάνει χώρα καθημερινά και πολλές φορές στον εγκέφαλο μας. Η γνώση είναι αυτό που διαχωρίζει και τον άνθρωπο από τη μηχανή.

Η πρώτη προσπάθεια στον τομέα ΤΝ έγιναν όταν ο άνθρωπος ήθελε οι μηχανές να μπορούν να παίζουν λογικά παιχνίδια, σκάκι, ντάμα, τρίλιζα, κλπ. Το απόγειο της προσπάθειας ήταν το 1996 με 1997, όταν ο Deep Blue έπαιξε σκάκι με τον Garry Kasparov. Οι προγραμματιστές είχαν ανακοινώσει ότι κατάφεραν να «φορτώσουν» ό,τι βιβλίο υπάρχει σχετικά με το σκάκι μέσα στο σύστημα, του οποίου οι ικανότητες, τελικά, απορρέαν από τους υπολογισμούς και όχι από τη «βαθιά» κατανόηση των εννοιών του παιχνιδιού. Κάτι που δεν αναδεικνύεται όταν συζητείται το γεγονός, είναι ό,τι ο Kasparov και ο Deep Blue δεν έπαιξαν μόνο μία παρτίδα αλλά δώδεκα, εκ των οποίων τέσσερις ήταν νίκες του πρώτου, τρεις του δεύτερου και 5 ισοπαλίες. Έτσι, διαμορφώθηκε η αντίληψη ότι τα λογικά παιχνίδια δεν απαιτούν γνώση, αλλά αντιθέτως, χρειάζεται η δυνατότητα επιλογής της καλύτερης από μια ένα πλήθος εναλλακτικών λύσεων.

Σύντομα, όμως, φάνηκε ότι είναι αναγκαία και η εκτενής γνώση αλλά μια πρωτοπόρα προσπάθεια από τους Newell και Simon, με ένα σύστημα που θα μπορούσε να λύσει οποιοδήποτε πρόβλημα σε οποιοδήποτε τομέα με τη χρήση ΤΝ, απόβηκε μάταιη και αυτό οδήγησε στη αλλαγή του προσανατολισμού στους στόχους της.

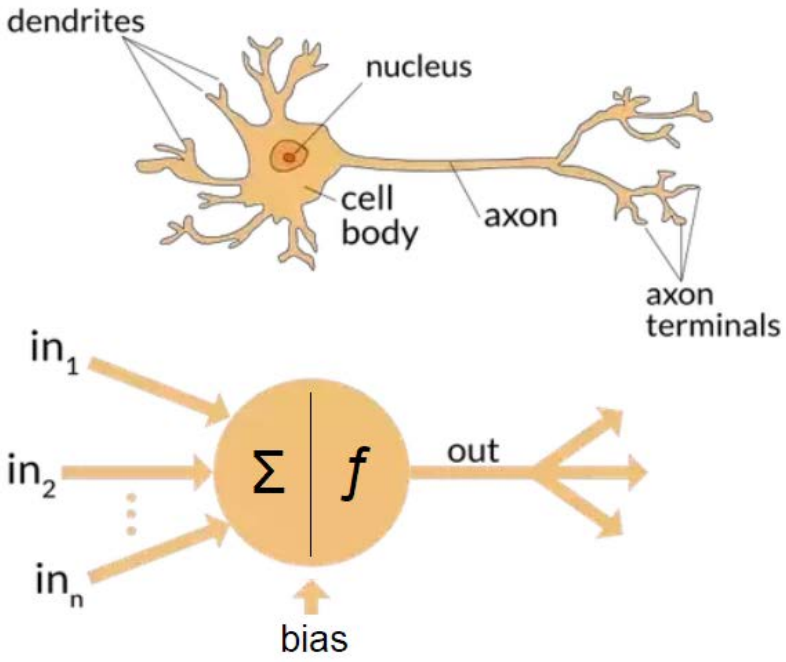
Το επόμενο βήμα ήταν η προσπάθεια αναπαράστασης γνώσης με συμβολικό τρόπο, κάτι που οδήγησε στην ανάπτυξη νέων γλωσσών και περιβαλλόντων προγραμματισμού, ειδικών για συστήματα ΤΝ. Τη όλη προσπάθεια έρχονται να ενισχύσουν τα έμπειρα συστήματα των οποίων ο στόχος είναι η αυτοματοποίηση της εξειδικευμένης γνώσης. Σήμερα, καλύπτεται ένα ευρύ φάσμα θεμάτων και η μάθηση εξακολουθεί να είναι το κεντρικότερο σημείο εστίασης. Πλέον, η ευφυής ανάλυση (intelligent data analysis – IDA), η εξόρυξη (data mining – DM) από μεγάλες βάσεις δεδομένων (knowledge discovery in databases – KDD) έχουν εξελιχθεί σε αρκετά μεγάλο βαθμό ενώ εξελίσσεται παράλληλα και μια άλλη κατεύθυνση, η μάθηση με ενίσχυση (reinforcement learning), η οποία έχει άμεση εφαρμογή σε έμπειρα συστήματα, σε συστήματα ρομποτικής, σε συστήματα βάσεων περιστατικών, κτλ.[12]

Στη συνέχεια, θα δούμε ένα βασικό συστατικό της ΤΝ, τα νευρωνικά δίκτυα (neural networks).

3.3 Νευρωνικά Δίκτυα

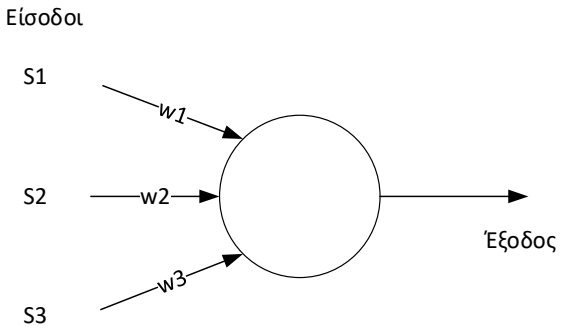
Το βιολογικό νευρωνικό δίκτυο, αποτελείται από ένα σύνολο κυττάρων, τους νευρώνες, οι οποίοι είναι διασυνδεμένοι μεταξύ τους με τρόπο ώστε να μεταφέρονται νευρικά/ηλεκτρικά σήματα, είτε από ένα σε ένα είτε από ένα σε πολλούς. Δεδομένου, ότι αυτό το μοτίβο μπορεί να επαναληφθεί για όλους τους νευρώνες, αντιλαμβανόμαστε πόσο πολύπλοκο μπορεί να γίνει ένα τέτοιο δίκτυο. Ο άνθρωπος, αν και έχει κάνει αυτήν τη σημαντική ανακάλυψη από το 1836 [14], δεν είναι επακριβώς γνωστό πως η λειτουργεί η σκέψη, η αναγνώριση αντικειμένων, η αναγνώριση κάποιου ήχου, κλπ. Έχουν επιβεβαιωθεί απλούστερες λειτουργίες όμως και σε αυτό το γεγονός στηρίχθηκε και η επιστήμη της Πληροφορικής, όταν ασχολήθηκε με τα νευρωνικά δίκτυα.

Τα τεχνητά νευρωνικά δίκτυα, προσπαθώντας να εξομοιώσουν τη βιολογική λειτουργία, αντέγραψαν προγραμματιστικά τον τρόπο λειτουργίας τους: ένας νευρώνας, συνδεμένος με άλλους μέσω συνάψεων και η μετάδοση ηλεκτρικών σημάτων σε όλο το δίκτυο, ως αποτέλεσμα μιας διεργασίας απόφασης.



Εικόνα 2: Ένας βιολογικός και ένας τεχνητός νευρώνας [4]

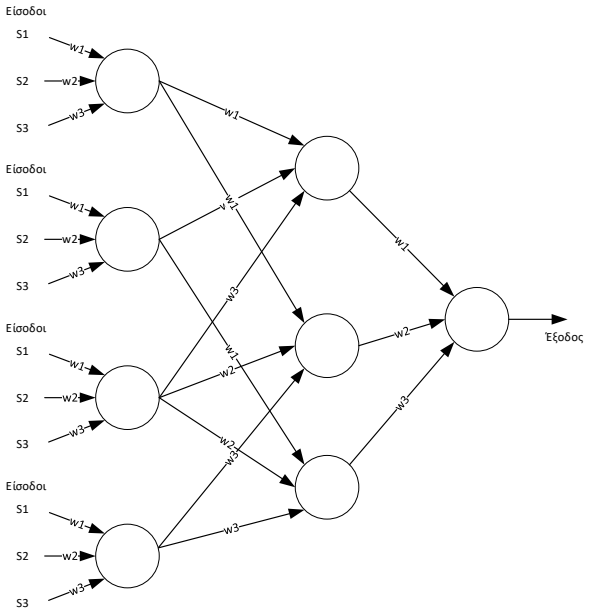
Ο τεχνητός νευρώνας μπορεί να έχει μία ή παραπάνω εισόδους, αλλά μία έξοδο. Δέχεται σαν είσοδο κάποια δεδομένα και στηριζόμενος στις πιθανές καταστάσεις της εσωτερικής δομής του, εμφανίζοντας στην έξοδο ένα αποτέλεσμα.



Εικόνα 3: Ο νευρώνας

Στους βιολογικούς νευρώνες υπάρχει ο χημικός δεσμός και στους τεχνητούς αντιπροσωπεύεται από το βάρος *w*. Κάθε είσοδος έχει ένα βάρος *w* το οποίο δείχνει πόσο μεγάλη είναι η συνεισφορά των προηγούμενων νευρώνων. Η διασύνδεση γίνεται κατ' επιλογήν: κάθε έξοδος μπορεί να είναι είσοδος για περισσότερους από έναν επόμενους νευρώνες.

Η μετάδοση των σημάτων γίνεται παράλληλα ενώ όλο το δίκτυο συμμετέχει στην εξαγωγή μιας απόφασης στην έξοδο. Και αυτό συμβαίνει, γιατί οι σχετικοί νευρώνες είναι οργανωμένοι κατά επίπεδα/στρώματα/layers.



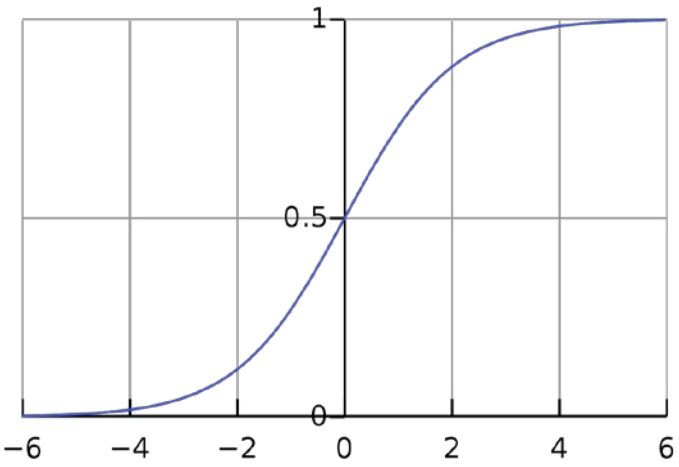
Εικόνα 4: Δίκτυο νευρώνων

Όταν υπάρχουν σήματα εισόδου, το στρώμα εισόδου θα τα λάβει για επεξεργασία, όλοι οι νευρώνες μαζί και παράλληλα. Αφού εκτελεστούν οι εσωτερικές διεργασίες σε κάθε έναν, το αποτέλεσμα είναι ένα σήμα στην έξοδο του καθενός. Το επόμενο στρώμα, για να ενεργοποιηθεί, πρέπει να λάβει σήματα εισόδου από το προηγούμενο και αυτό με τη σειρά του εκτελεί παρόμοιες διεργασίες. Στο τέλος, ο τελευταίος νευρώνας παράγει το σήμα εξόδου. Ένα δίκτυο μπορεί να αποτελείται από *n* αριθμό στρωμάτων, από τα οποία της εισόδου και εξόδου είναι τα φανερά ενώ όλα τα ενδιάμεσα είναι κρυφά.

Οι εσωτερικές διεργασίες σε κάθε νευρώνα, μπορούν να εκφραστούν με συναρτήσεις και η πιο κοινή είναι η σιγμοειδής συνάρτηση:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Το χαρακτηριστικό αυτής της συνάρτησης είναι ότι μπορεί και «περιορίζει» τις τιμές εξόδου μεταξύ 0 και 1, γεγονός που διευκολύνει την αριθμητική επεξεργασία:



Εικόνα 5: Γράφημα της Σιγμοειδούς συναρτήσεως

Κάθε νευρώνας, εφαρμόζει την σιγμοειδή συνάρτηση σε κάθε είσοδο, στη συνέχεια αθροίζει τα αποτελέσματα και παράγει την έξοδο. Τα βάρη *w*, παίζουν πολύ σημαντικό ρόλο και για να παράγει σωστές αποφάσεις το δίκτυο, θεμιτό είναι να μεταβάλλονται. Αυτό γίνεται κατά τη διάρκεια της εκπαίδευσης του δικτύου.

3.4 Αλγόριθμοι ΤΝ και εκπαίδευση τους

Η διαδικασία της εκπαίδευσης, γίνεται με εποπτευόμενο ή μη-εποπτευόμενο τρόπο (supervised/unsupervised). Κατά τον πρώτο, δίνουμε ένα σύνολο σημάτων στην είσοδο και ενώ γνωρίζουμε την επιθυμητή τιμή εξόδου, την συγκρίνουμε με την τιμή που θα μας δώσει το Νευρωνικό Δίκτυο. Η απόκλιση (error), χρησιμοποιείται ώστε να μεταβάλλουμε τα βάρη του και να μπορέσει έτσι να πλησιάσει όσο μπορεί την επιθυμητή τιμή. Γι' αυτό και όσο περισσότερα δείγματα σημάτων του παρέχουμε για εκπαίδευση, τόσο ελαττώνουμε την απόκλιση στην έξοδο, μέχρι αυτή να γίνει σταθερή. Τότε, λέμε ότι το δίκτυο έχει εκπαιδευτεί και είναι έτοιμο για πραγματικές αξιολογήσεις.

Κατά τη μη-εποπτευόμενη εκπαίδευση, η όλη διαδικασία είναι «αυτόνομη»: εμείς απλά παρέχουμε τα δεδομένα στο δίκτυο αλλά δεν δίνουμε αντίστοιχους στόχους και δεν συμμετέχουμε στην αλλαγή των βαρών. Το δίκτυο έχει εκπαιδευτεί όταν σταματάει να αλλάζει τις τιμές των βαρών και όταν η απόκλιση τείνει στο μηδέν ή είναι μηδέν.

Στην εποπτευόμενη εκπαίδευση, υπάρχουν τρεις κύριες κατηγορίες αλγορίθμων: Classification, Regression και Anomaly Detection. Στην πρώτη, το ΝΔ μας καλείται να προβλέψει μια απάντηση από ένα σύνολο κατηγοριοποιημένων δεδομένων (πχ χρώμα, κλπ) ενώ στη δεύτερη καλείται να προβλέψει μια αριθμητική τιμή. Στην τρίτη, καλείται να διερευνήσει δεδομένα τα οποία είναι ασυνήθιστα, εξ' ου και η ονομασία Anomaly. Βασικό είναι το ΝΔ να έχει εκπαιδευτεί πρωταρχικά σε «κανονικά» δεδομένα ώστε στη συνέχεια να μπορεί να ανιχνεύσει τις ανωμαλίες.

Κατά την επιλογή του αλγόριθμου, πρέπει να έχουμε υπόψη μας ότι κάθε αλγόριθμος, ανάλογα με την υλοποίηση του, έχει και συγκεκριμένα χαρακτηριστικά: ακρίβεια, χρόνος εκπαίδευσης, γραμμικότητα, πλήθος παραμέτρων, πλήθος χαρακτηριστικών (features).

Με την ακρίβεια, εννοούμε το πόσο κοντά στην πραγματική τιμή έχει πλησιάσει η έξοδος του αλγορίθμου και μερικές φορές είναι αποδεκτό να είναι κατά προσέγγιση. Φυσικά, εξαρτάται από το είδος του προβλήματος που θέλουμε να λύσουμε, οπότε και ανάλογα με το πόσο θέλουμε να προσεγγίσουμε την πραγματική τιμή, αλλάζει και ο χρόνος επεξεργασίας.

Ο χρόνος εκπαίδευσης είναι στενά συνδεδεμένος με την ακρίβεια και ανάλογα με το είδος και πλήθος των δεδομένων που έχουμε να επεξεργαστούμε, μπορεί να καθοριστεί και ο αλγόριθμος που θα επιλεγεί.

Με την γραμμικότητα εννοούμε τη δυνατότητα που υπάρχει από τους Linear Regression αλγόριθμους, ώστε να διαχωρίσουμε κάποια από τα αποτελέσματα με μια ευθεία (στο γράφημα τιμών των δεδομένων). Οι αλγόριθμοι logistic regression και support vector machines είναι αυτού του τύπου, οι οποίοι απαιτούν και τα δεδομένα να έχουν μια γραμμικότητα. Εδώ πρέπει να είμαστε και προσεκτικοί, διότι εφαρμόζοντας δεδομένα μη γραμμικά σε γραμμικούς αλγόριθμους, προφανώς θα έχουμε μεγάλα ποσοστά λαθών.

Οι παράμετροι, είναι ένας σημαντικός παράγοντας διότι επιτρέπουν να ρυθμίσουμε τον αλγόριθμο όπως χρειάζεται ώστε να επιτύχουμε το επιθυμητό αποτέλεσμα. Αυτό που γενικά ισχύει στην ΤΝ και στα ΝΔ, είναι ότι δεν υπάρχουν προκαθορισμένες βέλτιστες παράμετροι. Όσοι ασχολούνται με τον τομέα αυτό, παραδέχονται ότι χρησιμοποιούν trial and error ώστε να ρυθμίσουν τα συστήματα τους για να λάβουν όσο πιο σωστά, κατά προσέγγιση, αποτελέσματα μπορούν.

Τα χαρακτηριστικά αφορούν στα πρωτογενή δεδομένα που θα υποστούν επεξεργασία και τις περισσότερες φορές, χρειάζονται σωστή προετοιμασία. Ένας μεγάλος αριθμός χαρακτηριστικών (features) μπορεί να επιβραδύνει το ΝΔ ή να το καταστήσει μη αποτελεσματικό, γι' αυτό και συχνά, αφαιρούμε επιπλέον χαρακτηριστικά γιατί γνωρίζουμε ότι δεν θα επηρεάσουν δραματικά το αποτέλεσμα.

Στη συνέχεια, θα περιγράψουμε τους τρεις αλγόριθμους που θα χρησιμοποιήσουμε.

3.4.1 Αλγόριθμος Linear Regression

Ο αλγόριθμος χρησιμοποιείται για να προσπαθήσει να υπολογίσει μια τιμή, στηριζόμενο σε ένα σύνολο σχετικών δεδομένων. Αυτό το σύνολο δεδομένων μπορεί να εκφραστεί ως (x_i, y_i) όπου το x_i είναι διάνυσμα μεγέθους k, $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,k})$ και αν y_i είναι η επιθυμητή έξοδος, τότε υπάρχει η συνάρτηση $f: X \to Y$, $και\ θα\ θέλαμε\ να\ ισχύει\ f(x_i) \approx y_i$ για κάθε $i = 1, \dots, k$

Σε ένα νευρωνικό δίκτυο, έστω τα βάρη των κόμβων $w_0, w_1, \dots, w_k$, τότε η f γράφεται:

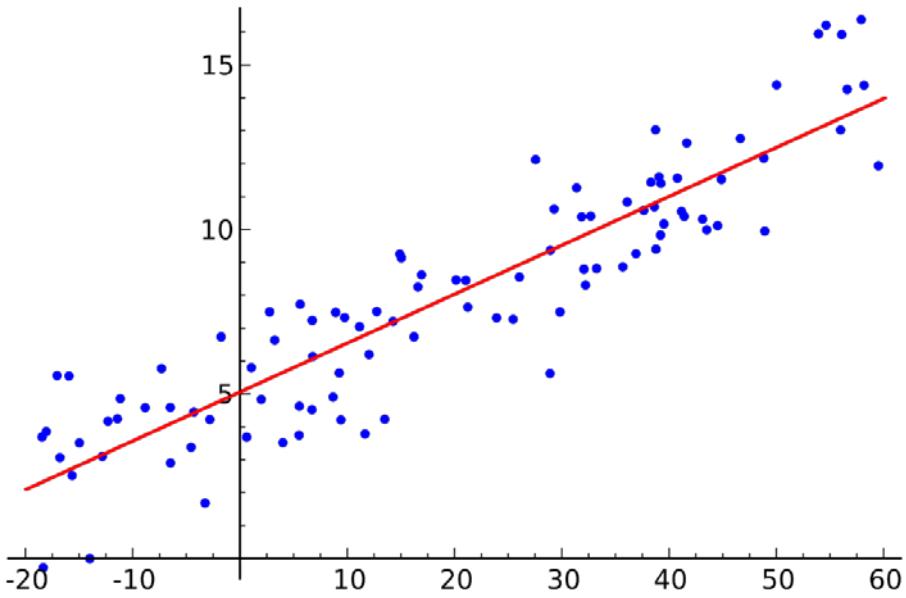
$$f(x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_k x_k = w_0 + \sum_{j=1}^{k} w_j x_j$$

Όταν τα βάρη αντιπροσωπεύουν διανύσματα, η f περιγράφεται

(όπου $w_0 = b, και\ \sum_{j=1}^{k} w_j x_j = w^T x$):

$$f(x) = b + w^T x$$

και αντιπροσωπεύει την ευθεία Linear Regression:



Εικόνα 6: Γράφημα Linear Regression

Αναζητώντας την βέλτιστη ευθεία, η οποία προσαρμόζεται «καλύτερα» στα σημεία του συνόλου δεδομένων, χρησιμοποιούμε τη μέθοδο των ελαχίστων τετραγώνων κατά την οποία θέλουμε τα ελάχιστα b, w^T:

$$J_n = \frac{1}{n}\sum_{i=1}^{n}(y_i - b - w^T x_i)^2$$

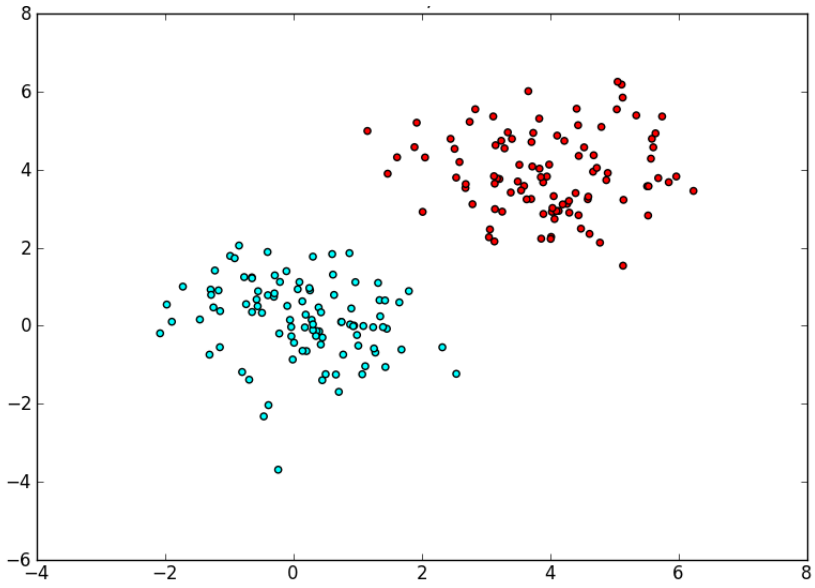
Επίσης χρησιμοποιείται ο αλγόριθμος της απότομης καθόδου (Gradient Decent) που χρησιμοποιείται για να ελαχιστοποιήσει την συνάρτηση J και με το $\frac{1}{2m}$, μπορούμε να ρυθμίσουμε το learning rate, το οποίο είναι μεταβαλλόμενο μέχρι να βρεθεί το συνολικό ελάχιστο:

$$minJ(b, w^T) = \frac{1}{2m}\sum_{\iota=1}^{m}(f(x^i) - y^i)^2$$

3.4.2 Αλγόριθμος Support Vector Machine

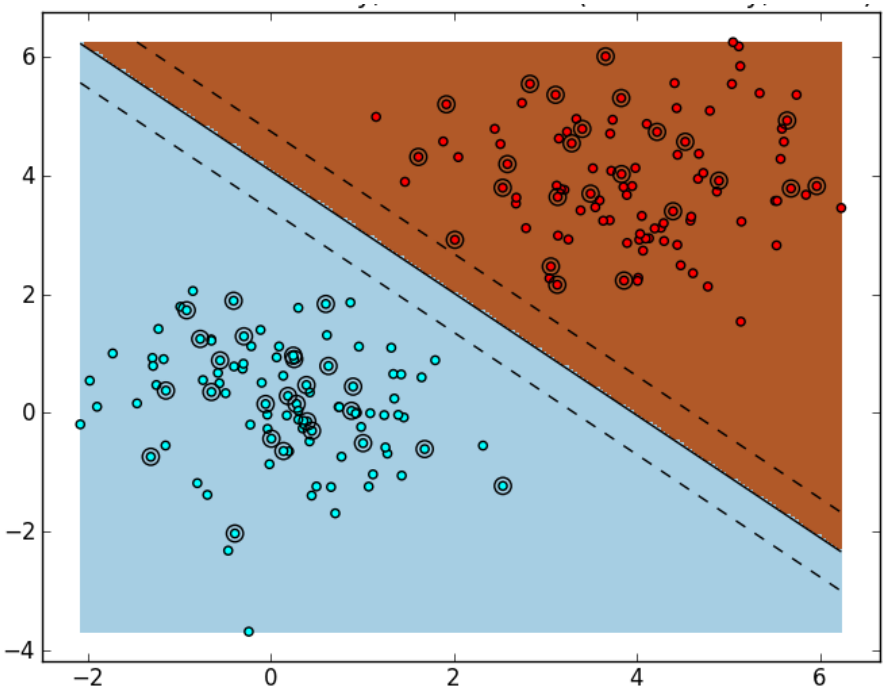
Ο αλγόριθμος SVM [3] χρησιμοποιείται για τον διαχωρισμό των δεδομένων σε δύο κλάσεις, τα οποία μπορεί να ανήκουν σε ένα (N+1)-διάστατο χώρο. Είναι δυνατό να διαχωρίσει δεδομένα που είναι γραμμικά διαχωρίσιμα είτε όχι, μετασχηματίζοντας τα και χρησιμοποιώντας Kernel functions. Το μέσο γι' αυτόν τον διαχωρισμό είναι ένα βέλτιστο *υπερεπίπεδο*: μια N-διάστατη αναλογία της γραμμής ή του επιπέδου, που διαχωρίζει τον (N+1)-διάστατο χώρο στα δύο.

Στην παρακάτω εικόνα όπου το N=2, τα δεδομένα είναι γραμμικώς διαχωρίσιμα:



Εικόνα 7: Γράφημα διασποράς των δεδομένων

και ο διαχωρισμός τους με τη χρήση του υπερεπιπέδου, N-1=1 διάστασης. Το βέλτιστο υπερεπίπεδο είναι αυτό το οποίο απέχει το μέγιστο από το κοντινότερο σημείο εκπαίδευσης (κυκλωμένα σημεία στην εικόνα 8). Οι διακεκομμένες βοηθητικές γραμμές, είναι και αυτές υπερεπίπεδα, πρέπει να απέχουν εξίσου από τη γραμμή του μεσαίου υπερεπιπέδου (margin) αλλά πρέπει να έχουν τη μέγιστη δυνατή απόσταση, και συνεπώς δημιουργείται «ο δρόμος» ή «το κανάλι». Έτσι, έχουμε ένα διακριτό διαχωρισμό σε δύο κλάσεις (στο σχήμα είναι τα καφέ και τα γαλάζια σημεία):



Εικόνα 8: Γράφημα διασποράς των δεδομένων διαχωρισμένων σε δύο κλάσεις

Έστω **n** σημεία τα οποία καταγράφονται, $(\overrightarrow{x_1}, y_1), \ldots, (\overrightarrow{x_n}, y_n)$, όπου το y_i θα πάρει τιμές 1 ή -1 και έτσι θα ξεκαθαριστεί σε ποια κατηγορία ανήκει. Το $\overrightarrow{x_i}$ είναι ένα vector p-διαστάσεων. Το υπερεπίπεδο καταγράφεται ως:

$$\vec{w} \times \vec{x} - b = 0$$

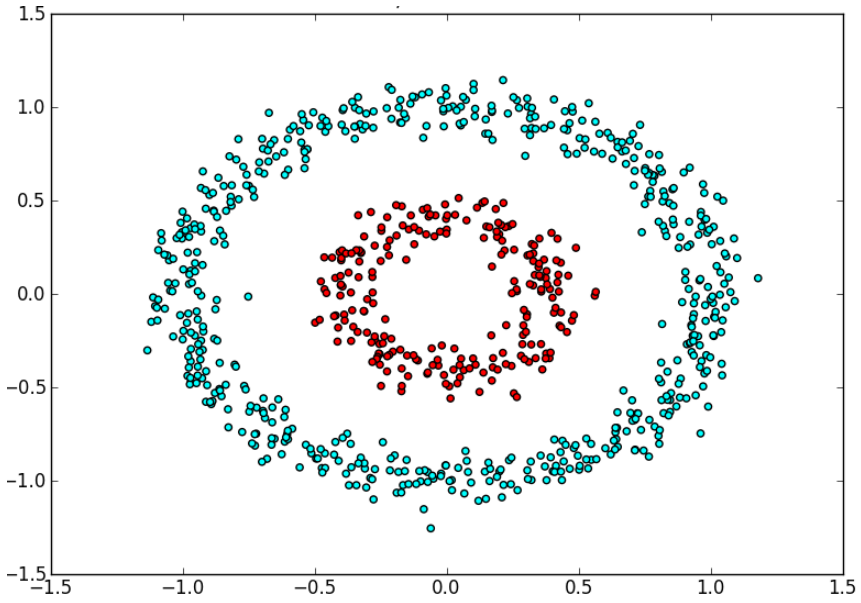
και οι παράλληλες:

$$\vec{w} \times \vec{x} - b \geq 1 \; \gamma\iota\alpha \; y_i = 1$$

$$\kappa\alpha\iota \; \vec{w} \times \vec{x} - b \leq -1 \; \gamma\iota\alpha \; y_i = -1$$

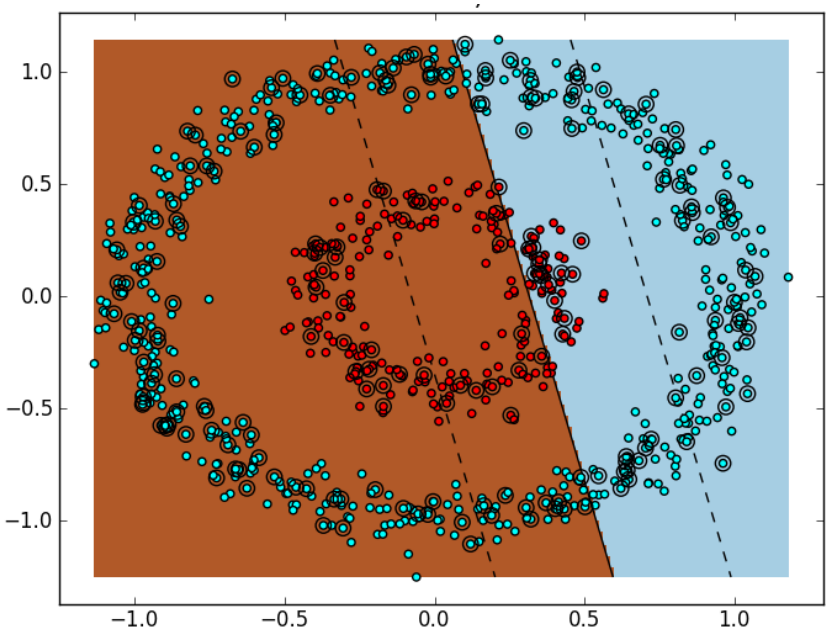
Υπάρχουν σημεία που πλησιάζουν ή εφάπτονται στα υπερεπίπεδα που είναι στα άκρα του δρόμου. Αυτά τα σημεία καθορίζουν και ποιο θα είναι το πλάτος του, γι' αυτό και λέγονται Support Vectors.

Στην εικόνα 9, τα δεδομένα δεν είναι γραμμικώς διαχωρίσιμα ή γενικά δεν υπάρχει ευθεία γραμμή στο $\boldsymbol{R^2}$ η οποία να μπορεί να τα διαχωρίσει:



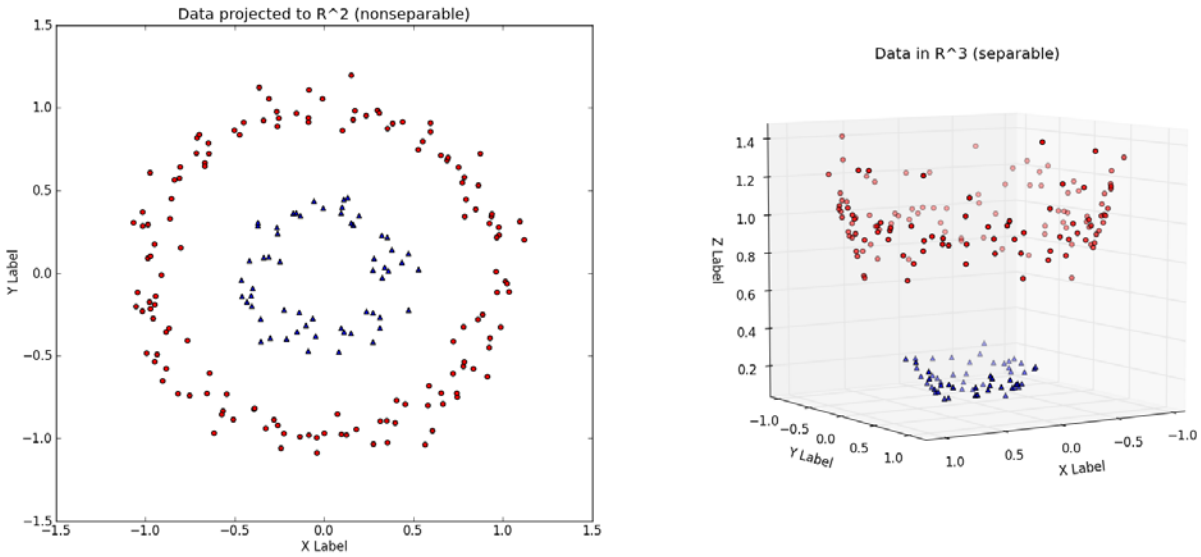
Εικόνα 9: Γράφημα διασποράς δεδομένων μη-γραμμικώς διαχωρίσιμων

Για παράδειγμα, κάτι τέτοιο θα είχε μεγάλα ποσοστά αποτυχιών (εικόνα 10):



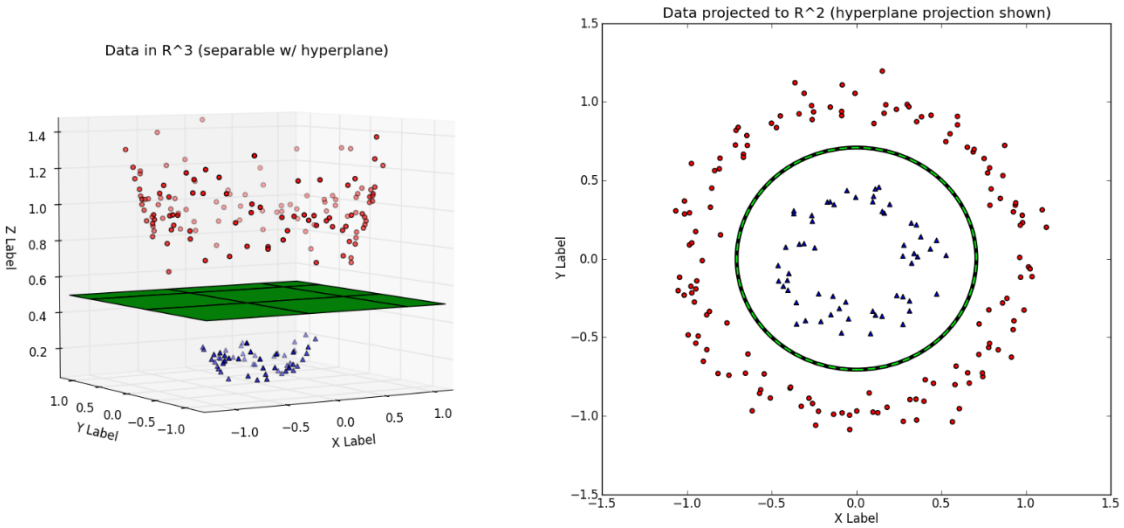
Εικόνα 10: Γράφημα διασποράς των δεδομένων, αν διαχωριστούν γραμμικώς

Η απάντηση στο πρόβλημα βρίσκεται στο μετασχηματισμό των δεδομένων: εφόσον ορίσαμε ότι ανήκουν σε ένα (N+1)-διάστατο χώρο, τότε χρησιμοποιώντας το υπερεπίπεδο N-διάστασης, μπορούμε να τα διαχωρίσουμε γραμμικά. Το σύνολο μας αν αναπαρασταθεί στο $\boldsymbol{R^3}$, βλέπουμε ότι υπάρχει δυνατότητα γραμμικού διαχωρισμού τους (εικόνα 11):



Εικόνα 11: Γράφημα διασποράς των δεδομένων αναπαριστάμενα στο R^3

Έτσι, μετασχηματίζουμε το σύνολο δεδομένων μας σε ένα νέο, μεγαλύτερων διαστάσεων, μέσω μιας συνάρτησης μετασχηματισμού φ και εφόσον το νέο σύνολο μπορεί να διαχωριστεί, τότε χρησιμοποιούμε τον SVN για να βρούμε το υπερεπίπεδο που θα το πράξει αυτό. Μετασχηματίζοντας αντίστροφα το διαχωρισμένο σύνολο στην αρχική διάσταση, θα δούμε ότι έγινε διαχωρισμός αλλά όχι γραμμικός (εικόνα 12):



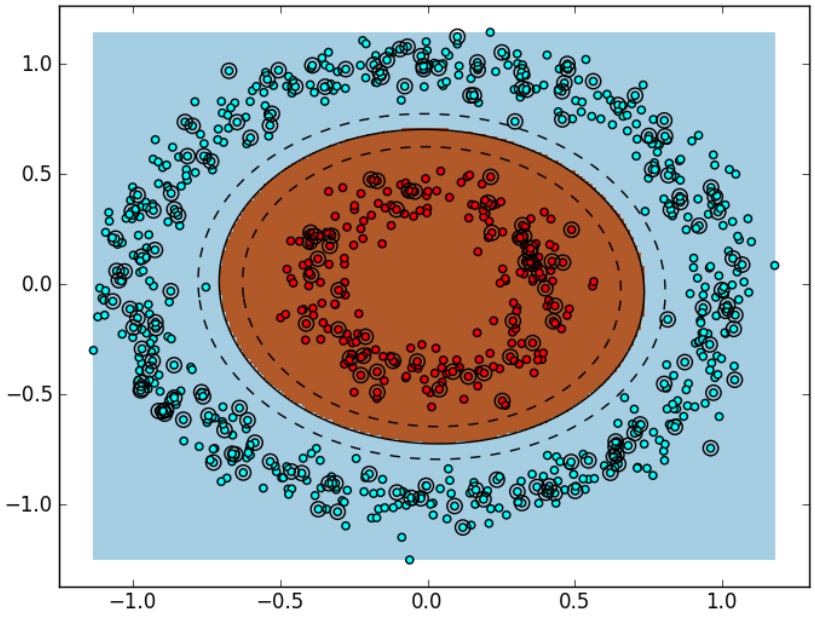
Εικόνα 12: Γράφημα διασποράς των διαχωρισμένων δεδομένων στο R^3 και μετασχηματισμένα στο R^2

Τα παραπάνω βήματα εξελίσσονται ομαλά όταν έχουμε να διαχειριστούμε λίγες διαστάσεις. Σε πολλές διαστάσεις όμως, ο αλγόριθμος απαιτεί πολύ χρόνο και υπολογιστή ισχύ για να καταφέρει να κάνει τους υπολογισμούς.

Οι kernel functions (ή συναρτήσεις πυρήνα) δίνουν λύση στο πρόβλημα, υπολογίζοντας το εσωτερικό γινόμενο δύο διανυσμάτων σε υψηλότερη διάσταση, χωρίς να χρειαστεί αυτά να μετασχηματιστούν. Μερικές κοινές συναρτήσεις είναι:

- Γραμμική: $k(x_1, x_2) = \langle x_1, x_2 \rangle$
- Πολυωνυμική: $k(x_1, x_2) = (y\langle x_1, x_2 \rangle + c_0)^d$
- Radial Basis Function (RBF): $k(x_1, x_2) = \mathrm{e}^{-\gamma(x_i - x_j)^2}$
- Sigmoid: $k(x_1, x_2) = \tanh(k_1\langle x_1, x_2 \rangle + k_2)$

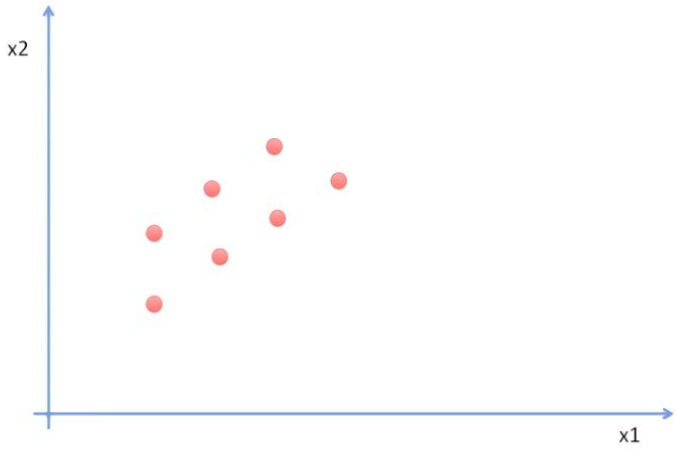
Αν εφαρμόσουμε την πολυωνυμική συνάρτηση έχουμε τον εξής διαχωρισμό (εικόνα 13):



Εικόνα 13: Γράφημα διασποράς των διαχωρισμένων δεδομένων

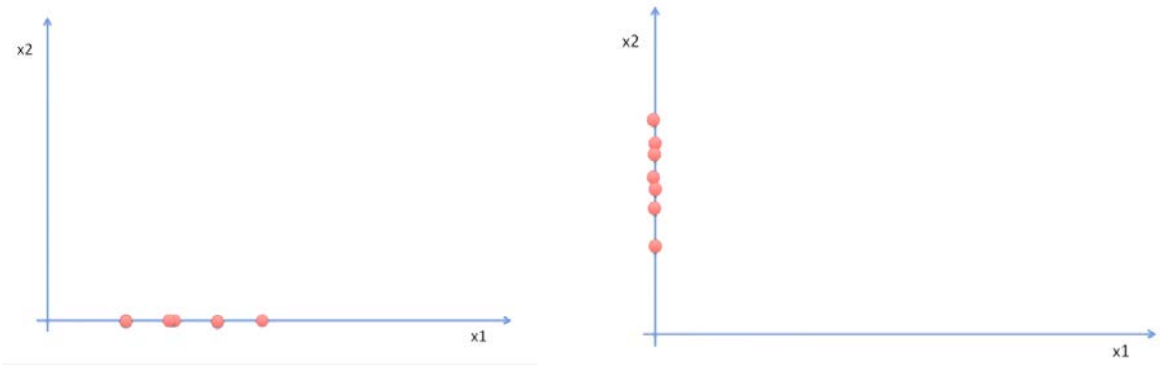
3.4.3 Αλγόριθμος Principal Component Analysis

Ο αλγόριθμος PCA [15] χρησιμοποιείται για να κατηγοριοποίηση των δεδομένων ενός συνόλου πολλών διαστάσεων. Πιο συγκεκριμένα, επειδή σε σύνολα πολλών διαστάσεων είναι δύσκολο να βρεθούν μοτίβα που κατηγοριοποιούν τα δεδομένα, προσπαθούμε να βρούμε ένα νέο σύνολο δεδομένων, το οποίο θα παραχθεί από το προηγούμενο με μετασχηματισμό και με μείωση των διαστάσεων του, χωρίς μεγάλη απώλεια δεδομένων. Αν το σύνολο μας είναι δύο διαστάσεων και αναπαρίσταται όπως παρακάτω (εικόνα 14):



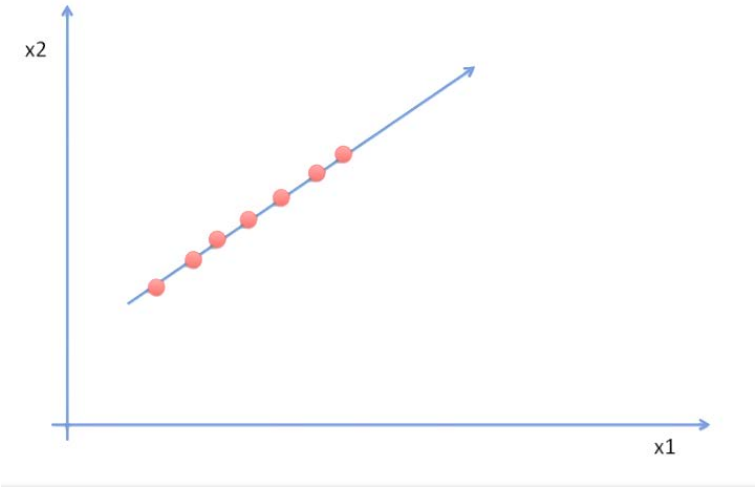
Εικόνα 14: Γράφημα διασποράς των δεδομένων

μπορούμε να το μετασχηματίσουμε σε μία διάσταση με έναν από τους δύο τρόπους (εικόνα 15):



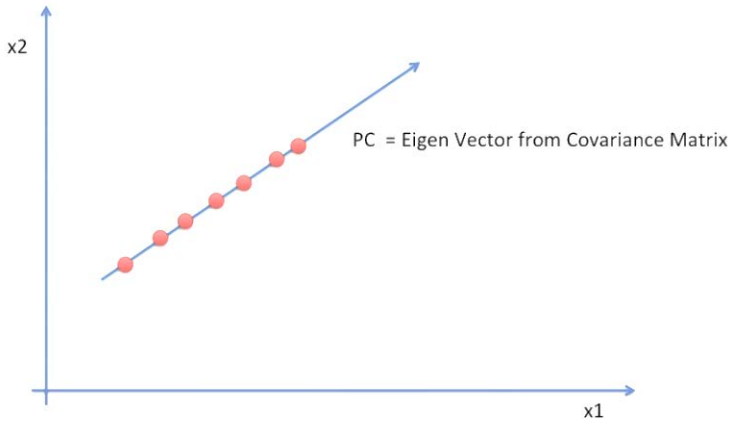
Εικόνα 15: Γραφήματα των μετασχηματισμένων δεδομένων

Είναι εμφανής η απώλεια της λεπτομέρειας και της πληροφορίας και στις δύο περιπτώσεις (διότι τα σημεία επικαλύπτονται). Μια εναλλακτική λύση είναι η παρακάτω (εικόνα 16):



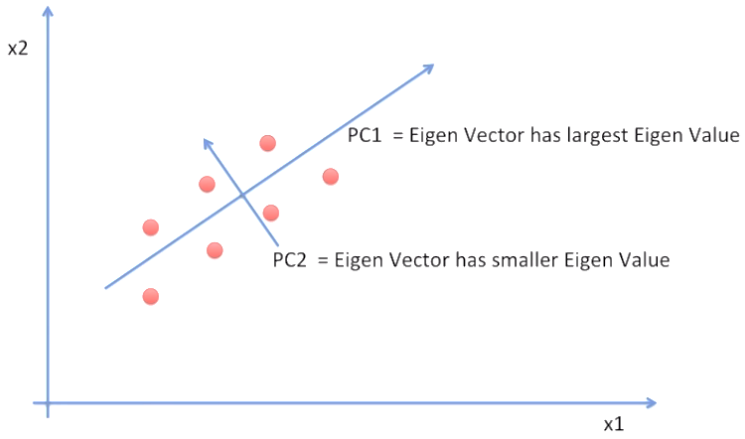
Εικόνα 16: Γράφημα μετασχηματισμένων δεδομένων

όπου αναπαριστούμε τα σημεία σε ένα διάνυσμα, με ελάχιστη απώλεια πληροφορίας. Τέτοια διανύσματα υπάρχουν περισσότερα του ενός, με διάφορα μήκη και ονομάζονται ιδιοδιανύσματα (eigen vectors). Το πλήθος τους εξαρτάται από τις διαστάσεις του συνόλου:



Εικόνα 17: Γράφημα μετασχηματισμένων δεδομένων με ένα Eigen Vector

Σε ένα δισδιάστατο σύνολο δεδομένων, υπάρχουν 2 ιδιοδιανύσματα τα οποία είναι Principal Components:



Εικόνα 18: Γράφημα μετασχηματισμένων δεδομένων με δύο Eigen Vectors

Τα PC είναι πάντα κάθετα μεταξύ τους. Για τον υπολογισμό τους, χρειάζεται να υπολογίσουμε:

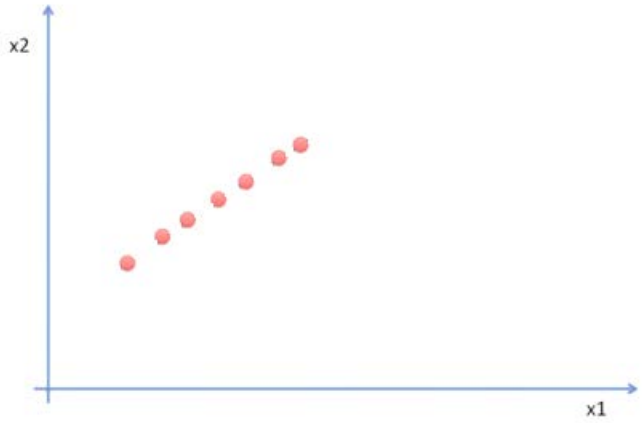
- Μέση τιμή: $\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$ $και$ $\bar{Y} = \frac{\sum_{i=1}^{n} Y_i}{n}$
- Υπολογισμός του RawDataAdjust πίνακα με τα δεδομένα, ως εξής:
    - Όλες οι τιμές x θα έχουν μειωθεί κατά $\bar{x}, δηλ$ $(x - \overline{x})$ και ομοίως οι y
- Συνδιακύμανση (convariance): $cov(X, Y) = \sum_{i=1}^{n} \frac{(X_i - \bar{X})(Y_i - \bar{Y})}{n-1}$
- Πίνακα συνδιακύμανσης ανά δύο διαστάσεις:

$$C^{n \times n} = (c_{i,j}) \text{ όπου } c_{i,j} = cov(Dim_i, Dim_j)$$

- Εύρεση ιδιοδιανυσμάτων και ιδιοτιμών με χρήση του πίνακα συνδιακύμανσης: $Cv = \lambda v$
- Δημιουργία ενός FeatureVector από τα ιδιοδιανύσματα και υπολογισμός του FinalData:

$$FinalData = FeatureVectore^T \times RowDataAdjust$$

Ο Πίνακας FinalData έχει τα τελικά μας δεδομένα και το γράφημα είναι όπως παρακάτω (εικόνα 19):



Εικόνα 19: Γράφημα με τα τελικά μας δεδομένα

3.5 Το μέλλον των Νευρωνικών Δικτύων

Από τη στιγμή που το μέγεθος και το πλήθος των δεδομένων προς επεξεργασία έχει αρχίσει να γιγαντώνεται, είναι αναμενόμενο ότι τα υπάρχοντα υπολογιστικά συστήματα δεν επαρκούν. Οι υπηρεσίες νέφους, μπορούν να δώσουν λύση σε αυτό το πρόβλημα γι' αυτό οι εταιρίες πάροχοι έχουν αναπτύξει δικές τους πλατφόρμες και γλώσσες προγραμματισμού Νευρωνικών δικτύων, γεγονός που βοήθησε στην εκτόξευση του ενδιαφέροντος του τομέα της ΤΝ αλλά και στην εξέλιξη του κλάδου. Η υπολογιστική ισχύς που διαθέτουν (με συστήματα πολλαπλών επεξεργαστών αλλά και με τη συνδρομή καρτών γραφικών – CUDA) στον χρήστη, ωθεί την έρευνα σε άλλα επίπεδα και άλλα μεγέθη.

Τα επόμενα χρόνια, μία από τις ειδικότητες που αναμένεται να ανθίσουν στον τομέα της Πληροφορικής, θα έχει σχέση με τον προγραμματισμό των Νευρωνικών Δικτύων στο Νέφος, μιας και όλα συγκλίνουν προς αυτό.

Κεφάλαιο 4
Προτεινόμενη Μεθοδολογία

4.1 Εισαγωγή

Υπάρχουν αρκετοί πάροχοι υπηρεσιών νέφους οι οποίοι παρέχουν και την αντίστοιχη πλατφόρμα υλοποίησης Νευρωνικών Δικτύων. Χαρακτηριστικό είναι ότι, αν και υπάρχουν ήδη κάποια πρότυπα (ONNX), οι πάροχοι υποστηρίζουν το δικό τους σύστημα και φροντίζουν να είναι συμβατό με το ONNX ενώ δεν λείπει και η πολυδιάστατη υποστήριξη (ένας πάροχος να υποστηρίζει το δικό του πρότυπο, το ONNX αλλά και το πρότυπου των άλλων παρόχων). Για την υλοποίηση μας χρησιμοποιήσαμε το Microsoft Azure cloud όπου και δημιουργήσαμε το δικό μας workspace στο Azure ML Studio.

Με τη χρήση της γλώσσας προγραμματισμού Python 3 για επεξεργασία του μεγάλου όγκου δεδομένων, ενός Azure Virtual Machine (4c/16GB έως 16c/64Gb) με Centos 7.6 ως SQL Server client, του Azure SQL Server (10 έως 50 DTU) για καταχώρηση των εγγραφών, του Azure ML Studio για δημιουργία πειραμάτων και οπτικοποίηση των αποτελεσμάτων αλλά και του Azure DevOps (πρώην VSTS) για project management με Git repositories για αποθήκευση του κώδικα,

δημιουργήθηκε το project «**exypnos**». Βοηθητικά λογισμικά, όπως PyCharm 2019, Visual Studio Code, Azure Notebooks, Azure Data Studio και Git, συνετέλεσαν στη δημιουργία ενός περιβάλλοντος ανάπτυξης σε Windows 2010 και Mac OS 10.14.14 ενώ η εκτέλεση του κώδικα γινόταν αποκλειστικά στο Azure VM. Επικουρικά, η online πλατφόρμα εκπαίδευσης Coursera, πρόσφερε ένα training path για Machine και Deep Learning, με τα αντίστοιχα εργαλεία υλοποίησης (MATLab Online και Jupyter Notebooks).

4.2 Περιγραφή της Υλοποίησης

Στο μοντέλο που θα παρουσιασθεί, γίνεται χρήση τριών αλγορίθμων, οι οποίοι περιγράφηκαν αναλυτικά προηγουμένως, και τα βήματα της υλοποίησης είναι τα παρακάτω:

1. Η πολιτική ασφαλείας *A*, θα γίνει η περιγραφή της προγραμματιστικά και θα εκφραστεί ο σχετικός μαθητικός τύπος,

2. Ο αλγόριθμος Linear Regression θα χρησιμοποιηθεί για να προβλέψει κατά πόσο μια μεμονωμένη ενέργεια ενός χρήστη είναι απειλητική και σε τι βαθμό. Τα δεδομένα θα εισαχθούν με μορφή one-hot vector και θα ληφθεί ως αποτέλεσμα, ένας βαθμός απειλής (threat_score),

3. Στη συνέχεια, οι ενέργειες ενός χρήστη ανά ημέρα, βαθμολογημένες από το Linear Regression, θα κατευθυνθούν στην είσοδο του One-class Support Vector Machine ώστε να εκτιμηθούν αν ανήκουν στην κατηγορία κανονικών ή ανώμαλων γεγονότων,

4. Παράλληλα, θα τροφοδοτηθούν και στην είσοδο του PCA για τον ίδιο λόγο,

5. Στο τέλος, τα αποτελέσματα των δύο αλγορίθμων θα συγκριθούν για να συμπεράνουμε αν έγινε επιτυχώς (και σε τι ποσοστό) η ανίχνευση των πιθανών ανωμαλιών,

6. Θα απαντήσουμε στα ερευνητικά ερωτήματα 1, 2.

7. Στη συνέχεια, θα περιγράψουμε προγραμματιστικά την πολιτική ασφαλείας *B*, η οποία είναι περιοριστική (επιπέδου PCI-DSS, όπου δεν επιτρέπεται η διακίνηση αριθμών πιστωτικών καρτών). Θα εκτελέσουμε τα βήματα 2, 3, 4, 5 και θα απαντήσουμε στο ερευνητικό ερώτημα 3,

8. Αν κατά τη διάρκεια των πειραμάτων, προκύψουν περισσότερες πληροφορίες και δεδομένα, θα απαντηθεί το ερευνητικό ερώτημα 4.

4.3 Βήματα της Υλοποίησης

4.3.1 Περιγραφή του Dataset

Το Cert Division, ως μέλος του Software Engineering Institute και με τη συνεργασία άλλων φορέων, δημιούργησε ένα σύνολο από datasets για ερευνητικούς σκοπούς. Υπάρχουν αρκετές λεπτομέρειες για τον τρόπο που δημιουργήθηκε, τι είδους πληροφορίες παρέχει και είναι διαθέσιμα για λήψη από την ιστοσελίδα του (https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099). Τα dataset, έχουν οργανωθεί σε 10 ομάδες (r1, r2, r3.1, r3.2, r4.1, r4.2, r5.1, r5.2, r6.1, r6.2) και όσο αυξάνεται η αρίθμηση, τόσο πιο πολύπλοκο γίνεται αλλά και περισσότερες πληροφορίες περιέχει. [16]

Το περιεχόμενο του r6.2, προσπαθεί να εξομοιώσει αρχεία (log files) από ένα κεντρικό σύστημα καταγραφής (logging) και παρέχεται σε μορφή κειμένου (csv) για ευκολότερη διαχείριση. Περιέχει πληροφορίες για σύνδεση χρήστη σε κάποιο σύστημα (logon.csv), για σύνδεση κάποιας συσκευής USB (device.csv), για εργασίες αρχείων (file.csv), για αποστολή email (email.csv), για πλοήγηση στο διαδίκτυο (http.csv), για ψυχομετρικό προφίλ χρηστών (psychometric.csv) αλλά και όλον τον κατάλογο LDAP για τους τελευταίους 18 μήνες. Επιπλέον αρχείο είναι το devoy_file.csv που περιέχει πληροφορίες με τα αρχεία και τα συστήματα στα οποία είναι αποθηκευμένα.

Σε κάθε dataset, υπάρχει ένα αρχείο οδηγιών και γενικών κατευθύνσεων, για το περιεχόμενο αλλά και τον τρόπο συσχέτισης των δεδομένων. Τέλος, το αρχείο answers, περιγράφει τις εσκεμμένως καταχωρημένες εγγραφές, οι οποίες περιγράφουν τις απειλές ώστε ο ερευνητής να μπορέσει να επιβεβαιώσει με τη μελέτη του.

4.3.2 Ανάλυση του Dataset

Μετά την εύρεση του κατάλληλου dataset για πειραματισμό και αφού αναλύθηκε πρωτογενώς, διαπιστώθηκε ότι το πλήθος των εγγραφών και η ομαδοποίηση τους, απαιτούσε να προηγηθεί ένα στάδιο προετοιμασίας των δεδομένων ώστε η πλατφόρμα Azure ML Studio να μπορεί να τα επεξεργαστεί.

Αρχικά, το **r1** χρησιμοποιήθηκε ως οδηγός, λόγω του περιορισμένου αριθμού εγγραφών και αρχείων:

Logon	Device	http	LDAP
849.580	65669	3.451.665	1.001

Στο **r6.2** όμως, το πλήθος αυξήθηκε δραματικά:

Logon	Device	http	Email	File	Decoy-file	LDAP
3.530.286	1.551.829	117.025.217	109.994.958	2.014.884	31.096	4.001

Στο αρχείο **logon.csv** του r6.2, βλέπουμε εγγραφές όπως:

```
id,date,user,pc,activity
{F3X8-Y2GT43DR-4906OHBL},01/02/2010 02:19:18,DNS1758,PC-0414,Logon
{B4Q0-D0GM24KN-3704MAII},01/02/2010 02:31:12,DNS1758,PC-0414,Logoff
{T7J1-D4HK34KV-5476TCIJ},01/02/2010 02:34:02,DNS1758,PC-5313,Logon
{S4Y6-D8MQ05SA-0759HLIS},01/02/2010 02:53:30,DNS1758,PC-5313,Logoff
{F3P0-E7FH78CV-4874FRGZ},01/02/2010 04:07:31,DNS1758,PC-0012,Logon
```

Παρομοίως και στα παρακάτω:

device.csv:

```
id,date,user,pc,file_tree,activity
{Z2Q8-K3AV28BE-9353JIRT},01/02/2010 07:17:18,SDH2394,PC-
5849,R:\;R:\22B5gX4;R:\SDH2394,Connect
{C7F1-G7LE60RU-2483DAXS},01/02/2010 07:22:42,JKS2444,PC-
6961,R:\;R:\JKS2444,Connect
{T9A4-D4RV69OF-1704NINW},01/02/2010 07:31:42,CBA1023,PC-
1570,R:\;R:\42gY283;R:\48rr4y2;R:\59ntt61;R:\76xCQG0;R:\CBA1023,Connect
```

{S8L0-O6QQ15NL-0636OYNV},01/02/2010 07:33:28,GNT0221,PC-6427,R:\;R:\GNT0221,Connect
{U0F1-R1FX27FM-6954TTVU},01/02/2010 07:33:55,JKS2444,PC-6961,,Disconnect
{X4R9-W7HH90OA-2624LIUE},01/02/2010 07:37:13,SDH2394,PC-5849,,Disconnect
{F7Z4-U0BJ54IA-1102TDYV},01/02/2010 07:40:11,RCT1697,PC-5770,R:\;R:\488TX69;R:\58l8617;R:\RCT1697,Connect
{N0N4-S0KF96GO-9894WIGG},01/02/2010 07:41:02,ROR3483,PC-4365,R:\;R:\177Qcm8;R:\81c5yB4;R:\87zJ233;R:\ROR3483,Connect
{F0B4-J9BQ17DC-9497NDHP},01/02/2010 07:41:31,RCT1697,PC-5770,,Disconnect

http.csv:

id,date,user,pc,url,activity,content
{V1D3-W8BL16YA-2594OWGB},01/02/2010 06:21:31,ANC1950,PC-4921,http://icio.us/John_Edward_Brownlee_as_AttorneyGeneral_of_Alberta/ufa/Oenmvyvna_pehvfre_OnuvnCnegaref_va_Pevzr_Qbpgbe_Jub1324221901.asp,WWW Visit,"Further consultation with post-production team The Mill resulted in the ears and the singular fang each Adipose has. The preview version of the episode supplied to the press and aired at the press launch omitted the scene that features Rose; before broadcast, only the production team, Tate, and Tennant had seen the scene. On 2 March 1936, Bahia escorted Veinticinco de Mayo, which had the Argentine Navy Minister Rear Admiral Eleazar Videla embarked, and Almirante Brown in the last part of their journey to Rio de Janeiro. While joining the revolt, the crew of the scout cruiser murdered one of their officers."

email.csv:

id,date,user,pc,to,cc,bcc,from,activity,size,attachments,content
{I1O2-B4EB49RW-7379WSQW},01/02/2010 06:36:41,HDB1666,PC-6793,Louis.Bernard.Garza@dtaa.com,Emery.Ali.Holloway@dtaa.com,Hector.Donovan.Bray@dtaa.com,Hector.Donovan.Bray@dtaa.com,Send,45659,,"Now Sylvia, the object of Aminta's desire, arrives on the scene with her posse of hunters to mock the god of love. The piano arrangement was composed in 1876 and the orchestral suite was done in 1880. As writer Arnold Haskell said, ""... he accepts the challenge in Sylvia of coping with period music

without descending to pastiche; and never once does the movement he provides strike us as modern or as 'old world'"". Sylvia now grieves over Aminta, cherishing the arrow pulled from her breast nostalgically."

file.csv:

id,date,user,pc,filename,activity,to_removable_media,from_removable_media,content
{F3E2-X3MV05YQ-3516SZDT},01/02/2010 07:19:41,SDH2394,PC-5849,R:\60WBQE7S.doc,File Open,False,True,"D0-CF-11-E0-A1-B1-1A-E1 Ernesztin's brother, Lipot Hoffmann, provided for the family and acted much like a father to the boys. Deprived of his artist friends, he also found that Americans rejected having their photos taken on the street. Frustrated, Kertesz left Keystone after Prince left the company in 1937."

decoy-file.csv:

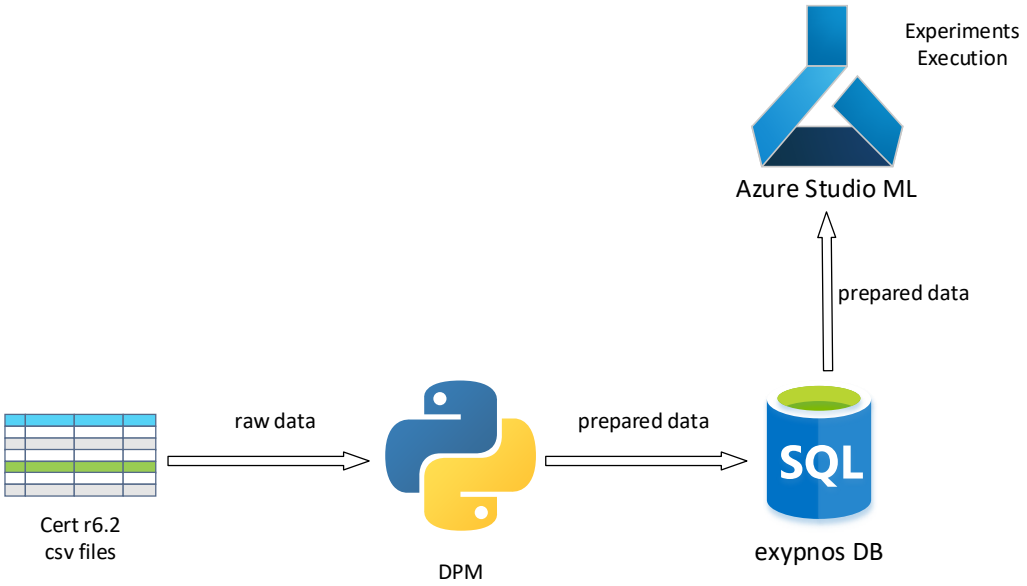
"decoy_filename","pc"
"C:\LJE2413\795JW126.jpg","PC-0302"
"C:\QMU9BC38.pdf","PC-6566"
"C:\GIS1668\YPS1RSIK.jpg","PC-2606"
"C:\KD02AETE.pdf","PC-5393"
"C:\AUZTDD4J.jpg","PC-8753"
"C:\51g8y45\FP7YAZ02.doc","PC-7913"

2009-12.csv (LDAP):

employee_name,user_id,email,role,projects,business_unit,functional_unit,department,team,supervisor
Nicholas Fletcher Pruitt, NFP2441, Nicholas.Fletcher.Pruitt@dtaa.com, ITAdmin,,1,1 - Adminstration,5 - Security,8 - ElectronicSecurity, Madison Charissa Malone

4.3.2 Μετατροπή των Δεδομένων

Τα raw data που καταγράφηκαν παραπάνω, δεν είναι σε μορφή επεξεργάσιμη από ένα NN και για το λόγο αυτό, κατασκευάστηκε ένα ενδιάμεσο σύστημα DPM (Data Preparation Module) σε Python 3, το οποίο αντιγράφει τις εγγραφές από τα παραπάνω αρχεία, τις συγχωνεύει σε μια μεγαλύτερη εγγραφή και είτε δημιουργεί ένα csv στο οποίο την αποθηκεύει είτε την εισάγει απευθείας στην Azure SQL βάση μας (εικόνα 20).



Εικόνα 20: Γενικό Διάγραμμα του Συστήματος

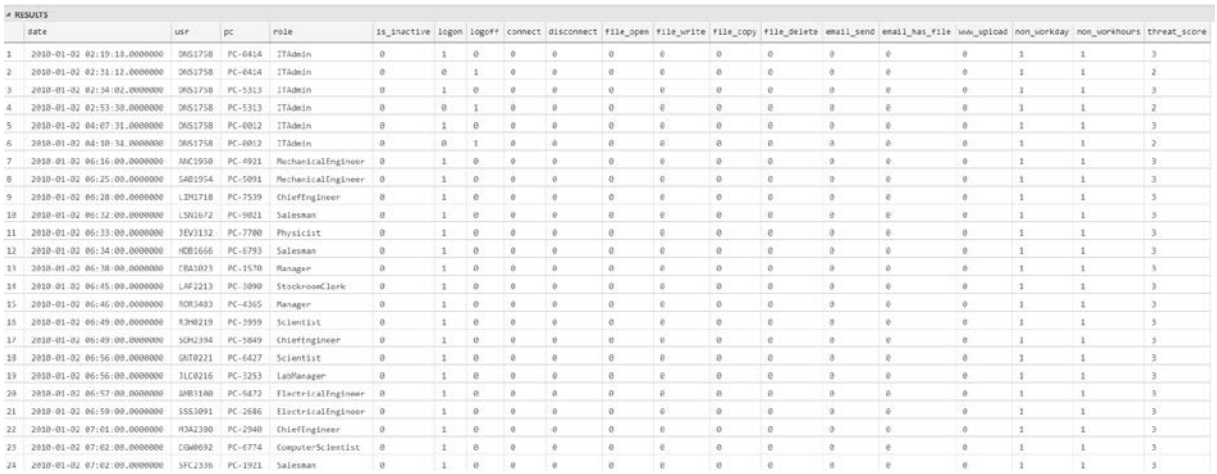
Η γραμμογράφηση της τελικής εγγραφής περιγράφεται παρακάτω:

Νέα πεδία:

```
date, usr, pc, role, is_inactive, logon, logoff, connect, disconnect,
file_open, file_write, file_copy, file_delete, email_send, email_has_file,
www_upload, non_workday, non_workhours, threat_score
```

Τα date, usr, pc αντιγράφονται ως έχουν από όλα τα πηγαία csv αρχεία. Το role είναι ένα πεδίο που εξάγεται από τα ldap αρχεία και όλα τα υπόλοιπα είναι πεδία του ενός bit, και χρησιμοποιούνται όπως σε ένα one-hot vector: εάν ο χρήστης έκανε logon τότε παίρνει την τιμή 1 και όλες οι άλλες έχουν μηδέν. Ονομάζονται και features του αλγόριθμου, διότι θα χρησιμοποιηθούν για να δημιουργηθεί ένα διάνυσμα της μορφής [0,1,0,1,0,0,0,0,0,1].

Ο πίνακας στη βάση μοιάζει όπως στην παρακάτω εικόνα 21:



	date	usr	pc	role	is_inactive	logon	logoff	connect	disconnect	file_open	file_write	file_copy	file_delete	email_send	email_has_file	www_upload	non_workday	non_workhours	threat_score
1	2010-01-02 02:19:18.0000000	DNS1758	PC-0414	ITAdmin	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
2	2010-01-02 02:31:12.0000000	DNS1758	PC-0414	ITAdmin	0	0	1	0	0	0	0	0	0	0	0	0	1	1	2
3	2010-01-02 02:34:02.0000000	DNS1758	PC-5313	ITAdmin	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
4	2010-01-02 02:53:30.0000000	DNS1758	PC-5313	ITAdmin	0	0	1	0	0	0	0	0	0	0	0	0	1	1	2
5	2010-01-02 04:07:31.0000000	DNS1758	PC-0012	ITAdmin	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
6	2010-01-02 04:10:34.0000000	DNS1758	PC-0012	ITAdmin	0	0	1	0	0	0	0	0	0	0	0	0	1	1	2
7	2010-01-02 06:16:00.0000000	ANC1950	PC-4921	MechanicalEngineer	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
8	2010-01-02 06:25:00.0000000	SAB1954	PC-5091	MechanicalEngineer	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
9	2010-01-02 06:28:00.0000000	LIM1718	PC-7539	ChiefEngineer	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
10	2010-01-02 06:32:00.0000000	LSN1672	PC-5021	Salesman	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
11	2010-01-02 06:33:00.0000000	JEV3132	PC-7700	Physicist	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
12	2010-01-02 06:34:00.0000000	HDB1666	PC-6793	Salesman	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
13	2010-01-02 06:38:00.0000000	CBA1023	PC-1570	Manager	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
14	2010-01-02 06:45:00.0000000	LAF2213	PC-3090	StockroomClerk	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
15	2010-01-02 06:46:00.0000000	RCR3403	PC-4365	Manager	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
16	2010-01-02 06:49:00.0000000	RJH0219	PC-3959	Scientist	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
17	2010-01-02 06:49:00.0000000	SCH2394	PC-5849	ChiefEngineer	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
18	2010-01-02 06:56:00.0000000	GAT0221	PC-6427	Scientist	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
19	2010-01-02 06:56:00.0000000	JLC0216	PC-3253	LabManager	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
20	2010-01-02 06:57:00.0000000	AMB3100	PC-5472	ElectricalEngineer	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
21	2010-01-02 06:59:00.0000000	SSS3091	PC-2686	ElectricalEngineer	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
22	2010-01-02 07:01:00.0000000	MJA2380	PC-2940	ChiefEngineer	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
23	2010-01-02 07:02:00.0000000	CGW0692	PC-6774	ComputerScientist	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3
24	2010-01-02 07:02:00.0000000	SFC2336	PC-1921	Salesman	0	1	0	0	0	0	0	0	0	0	0	0	1	1	3

Εικόνα 216: Δείγμα από εγγραφές στη βάση

Η γραμμογράφηση του περιγράφεται στον παρακάτω πίνακα:

Α/Α πεδίου	**Όνομα πεδίου**	**Ρόλος**
0	date	Η καταγεγραμμένη ημερομηνία του γεγονότος
1	usr	Ο χρήστης που πραγματοποίησε μια ενέργεια
2	pc	Το σύστημα που χρησιμοποιήθηκε
3	role	Ο ρόλος του χρήστη, δικαιολογεί αν ο χρήστης μπορεί να συνδεθεί μη εργάσιμες ημέρες και ώρες
4	is_inactive	Αν ο χρήστης έχει αποχωρήσει από την εταιρία, παίρνει την τιμή 1
5	logon	Ενέργεια εισόδου σε σύστημα
6	logoff	Ενέργεια εξόδου από σύστημα
7	connect	Ενέργεια σύνδεσης USB stick

8	disconnect	Ενέργεια αποσύνδεσης USB stick
9	file_open	Άνοιγμα αρχείου
10	file_write	Αποθήκευση αρχείου
11	file_copy	Αντιγραφή αρχείου
12	file_delete	Διαγραφή αρχείου
13	email_send	Αποστολή email
14	email_has_file	Αποστολή email με επισυναπτόμενο αρχείο
15	www_upload	Ανέβασμα αρχείου στον ιστό
16	non_workday	Μη εργάσιμη ημέρα (Σάββατο, Κυριακή)
17	non_workhours	Μη εργάσιμη ώρα (από 5μμ έως 8πμ)
18	threat_score	Παίρνει τιμές από 0 έως 40, και εκφράζει το βαθμό απειλής της ενέργειας του χρήστη

Πίνακας 1: Γραμμογράφηση αρχείου συμβάντων

Στη συνέχεια, στα πειράματα που πραγματοποιούνται, έχουμε τη δυνατότητα να επιλέγουν ποιες από τα πεδία αυτά θα συμμετέχουν.

4.3.4 Υλοποίηση Πολιτικής Ασφάλειας Α

Η πολιτική ασφάλειας Α, είναι τυπική και μπορεί να γίνει περιγραφή της με τον παρακάτω τρόπο:

«*Ο οργανισμός επιτρέπει στους εργαζόμενους να χρησιμοποιούν τους πόρους του για να παράγουν το απαιτούμενο έργο. Οι χρήστες, πρέπει να συνδέονται τις εργάσιμες ώρες και ημέρες, ενώ επιτρέπεται η χρήση φορητών συσκευών αποθήκευσης (USB), ή να γίνεται ανταλλαγή πληροφοριών μέσω email ή ιστοτόπων που διακινούν αρχεία.*

Οι λογαριασμοί χρηστών πρέπει να απενεργοποιούνται αμέσως μετά την αποχώρηση τους από τον οργανισμό».

Από τα νέα πεδία, η πολιτική Α μας επιβάλλει να χρησιμοποιήσουμε κάποια από αυτά για τον υπολογισμό του βαθμού απειλής (threat_score) της κάθε ενέργειας. Έτσι, από τα πεδία που περιγράψαμε προηγουμένως, σημειώνουμε ποια θα συμμετέχουν στην προγραμματιστική υλοποίηση της πολιτικής Α:

`date, usr, pc, role, **is_inactive**, **logon**, logoff, connect, disconnect, file_open, file_write, file_copy, file_delete, email_send, email_has_file, www_upload, **non_workday, non_workhours**`

Τα πεδία **non_workday και non_workhours** έχουν εξ ορισμού την τιμή 0, εάν η ενέργεια έχει πραγματοποιηθεί εντός των εργάσιμων ημερών και ωρών, αλλιώς έχει την τιμή 1 αν συμβεί το αντίθετο. Στην περίπτωση μας, το threat_score, υπολογίζεται με την εξής συνάρτηση:

*threat_score = (is_inactive + logon) * (non_working_day +1) * (non_workhours+1)*

Παραδείγματα:

- Ανενεργός χρήστης συνδέθηκε μη-εργάσιμη ημέρα σε ένα σύστημα:
    - (1 + 1) * 2 * 1 = 4,
- Χρήστης συνδέθηκε εκτός εργασίμων ωρών:
    - (0 + 1) * 1 *2 = 2,
- Αντέγραψε αρχείο, εκτός εργασίμων ωρών:
    - (0+0)*1*2 = 0

Συνοπτικά, αν το threat_score <2 θεωρείται κανονική ενέργεια, αλλιώς είναι πιθανή απειλή και πρέπει να διερευνηθεί.

4.3.5 Υλοποίηση Πολιτικής Ασφάλειας Β (PCI-DSS)

Ως γνωστόν, το PCI-DSS είναι ένα certification το οποίο περιέχει μια σειρά από κανόνες με σκοπό την προστασία των συστημάτων που διαχειρίζονται πιστωτικές κάρτες (είτε τις αποθηκεύουν είτε όχι). Σε μια από τις πολιτικές του, περιγράφεται η απαίτηση παρακολούθησης και ελέγχου των συστημάτων που ανήκουν στο CDE (Card Holder Environment), είτε είναι on-scope είτε όχι, για τυχόν διακίνηση αρχείων με πιστωτικές κάρτες με μη ασφαλή τρόπο ή σε μη εξουσιοδοτημένα άτομα. Σημειώνουμε ότι δεν υλοποιούμε ένα σύστημα Data Loss Prevention (DLP), το οποίο σαρώνει τον Η/Υ του χρήστη σε πραγματικό χρόνο και καταγράφει αν τα αρχεία που διαχειρίζεται, περιέχουν ευαίσθητα δεδομένα (unmasked PAN), αλλά ένα ενδιάμεσο το οποίο μπορεί πιο γρήγορα από το DLP, να ενεργοποιήσει alert σε περίπτωση που εκτελεστεί ενέργεια διαρροής.

Η πολιτική ασφάλειας Β, θα μπορούσε να περιγραφεί με τον παρακάτω τρόπο:

«*Ο οργανισμός επιτρέπει στους εργαζόμενους να χρησιμοποιούν τους πόρους του για να παράγουν το απαιτούμενο έργο. Οι χρήστες, πρέπει να συνδέονται τις εργάσιμες ώρες και ημέρες, ενώ εξαιρέσεις γίνονται στις περιπτώσεις που ο ρόλος τους το επιβάλει.*

Τα συστήματα των χρηστών που ανήκουν στο CDE, δεν μπορούν να χρησιμοποιούν φορητές συσκευές αποθήκευσης (USB), να αποστέλλουν email ή να διακινούν με μη ασφαλή τρόπο, αρχεία που περιέχουν ευαίσθητα δεδομένα χωρίς προστασία, όπως unmasked PAN.

Οι λογαριασμοί χρηστών πρέπει να απενεργοποιούνται αμέσως μετά την αποχώρηση τους από τον οργανισμό».

Από τα νέα πεδία, η πολιτική Β μας επιβάλλει να χρησιμοποιήσουμε κάποια από αυτά για τον υπολογισμό του βαθμού απειλής (threat_score) της κάθε ενέργειας. Έτσι, από τα πεδία που περιγράψαμε προηγουμένως, σημειώνουμε ποια θα συμμετέχουν στην προγραμματιστική υλοποίηση της πολιτικής Β:

`date, usr, pc, role, **is_inactive**, **logon**, logoff, **connect**, disconnect, file_open, file_write, **file_copy**, file_delete, **email_send**, **email_has_file**, **www_upload**, **non_workday, non_workhours**`

Τα πεδία αυτά έχουν και την ανάλογη βαθμολογία:

Όνομα Πεδίου	Βαθμός	Λεπτομέρειες
is_inactive	5	Ενέργεια υψηλού κινδύνου. Ο χρήστης θα έπρεπε να είναι ανενεργός.
logon	1	Ενέργεια χαμηλού κινδύνου
connect	2	Η μη εξουσιοδοτημένη σύνδεση USB μπορεί να οδηγήσει σε υλοποίηση εσωτερικής απειλής
file_copy	2	Η μη εξουσιοδοτημένη αντιγραφή αρχείων μπορεί οδηγήσει σε υλοποίηση εσωτερικής απειλής
email_send	2	Η αποστολή email μπορεί οδηγήσει σε υλοποίηση εσωτερικής απειλής
email_has_file	3	Η μη εξουσιοδοτημένη αποστολή αρχείων με email μπορεί οδηγήσει σε υλοποίηση εσωτερικής απειλής
www_upload	4	Η μη εξουσιοδοτημένη αποστολή αρχείων σε ιστότοπο μπορεί οδηγήσει σε υλοποίηση εσωτερικής απειλής
non_workday	2	Βαθμολόγηση των παραπάνω ενεργειών, διπλάσιος βαθμός αν πραγματοποιείται εκτός ημερών εργασίας, αλλιώς είναι 1
non_workhours	2	Βαθμολόγηση των παραπάνω ενεργειών, διπλάσιος βαθμός αν πραγματοποιείται εκτός ωρών εργασίας, αλλιώς είναι 1

Τα πεδία **non_workday και non_workhours** έχουν εξορισμού την τιμή 1, εάν η ενέργεια έχει πραγματοποιηθεί εντός των εργάσιμων ημερών και ωρών, αλλιώς έχει την τιμή 2 αν συμβεί το αντίθετο

Στην περίπτωση μας, το threat_score, υπολογίζεται με την εξής συνάρτηση:

*threat_score = (is_inactive*5 + logon*1 + connect*2 + file_copy*2 + email_send*2 + email_has_file*3 + www_upload*4) * (non_working_day+1) * (non_workhours+1)*

Παραδείγματα:

- Ανενεργός χρήστης συνδέθηκε μη-εργάσιμη ώρα σε ένα σύστημα:
    - (5 + 1 + 0 + 0 + 0 + 0) * 1 * 2 = 12,
- Χρήστης συνδέθηκε εντός εργάσιμων ωρών:
    - (0 + 1 + 0 + 0 + 0 + 0) * 1 *1 = 1,
- Χρήστης αντέγραψε ένα αρχείο, εντός εργάσιμων ωρών:
    - (0 + 0 + 0 + 2 + 0 + 0) * 1 *1 = 2,
- Χρήστης ανέβασε αρχείο σε ιστότοπο εντός εργάσιμων ωρών:
    - (0 + 0 + 0 + 0 + 0 + 4) * 1 *1 = 4,
- Χρήστης ανέβασε αρχείο σε ιστότοπο εκτός εργάσιμων ωρών:
    - (0 + 0 + 0 + 0 + 0 + 4) * 1 *2 = 8,

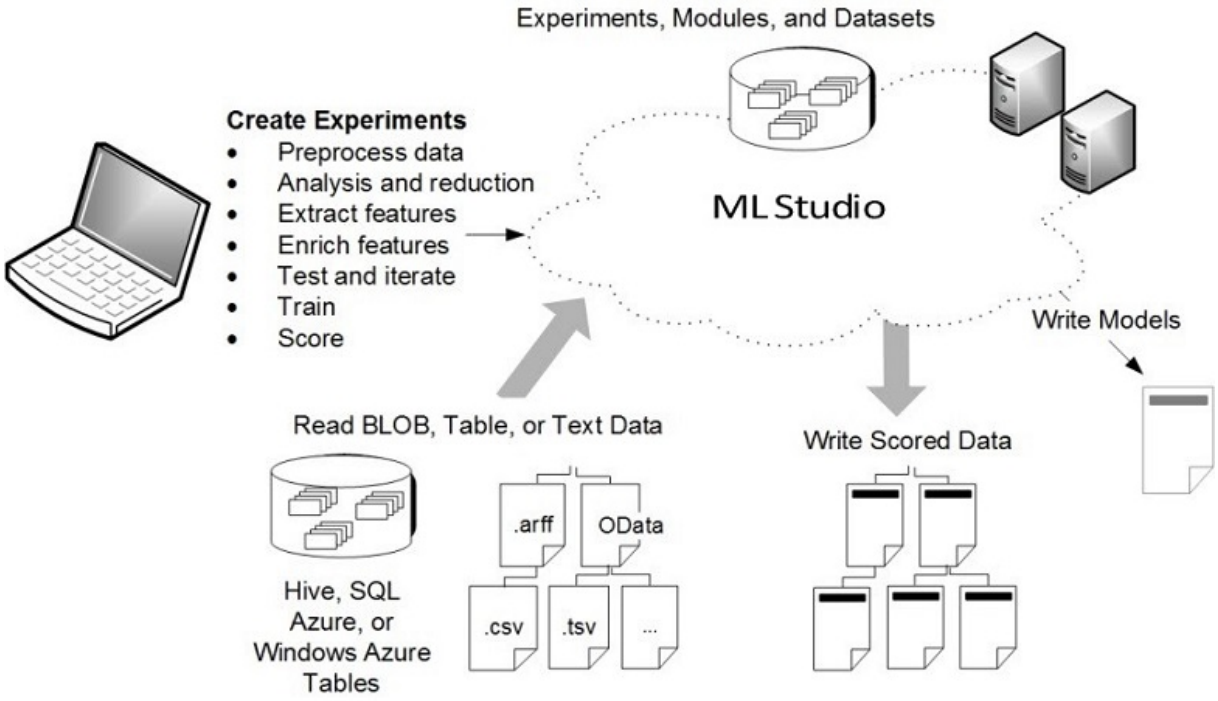
Συνοπτικά, threat_score <2 θεωρείται κανονική ενέργεια, αλλιώς είναι πιθανή απειλή και πρέπει να διερευνηθεί.

Στην παρούσα έκδοση, το exypnos δεν παρέχει User Interface για την εύκολη προσαρμογή των ρυθμίσεων και των πολιτικών, αλλά είναι κάτι που έχει σχεδιαστεί για το μέλλον.

4.4 Παρουσίαση Azure ML Studio

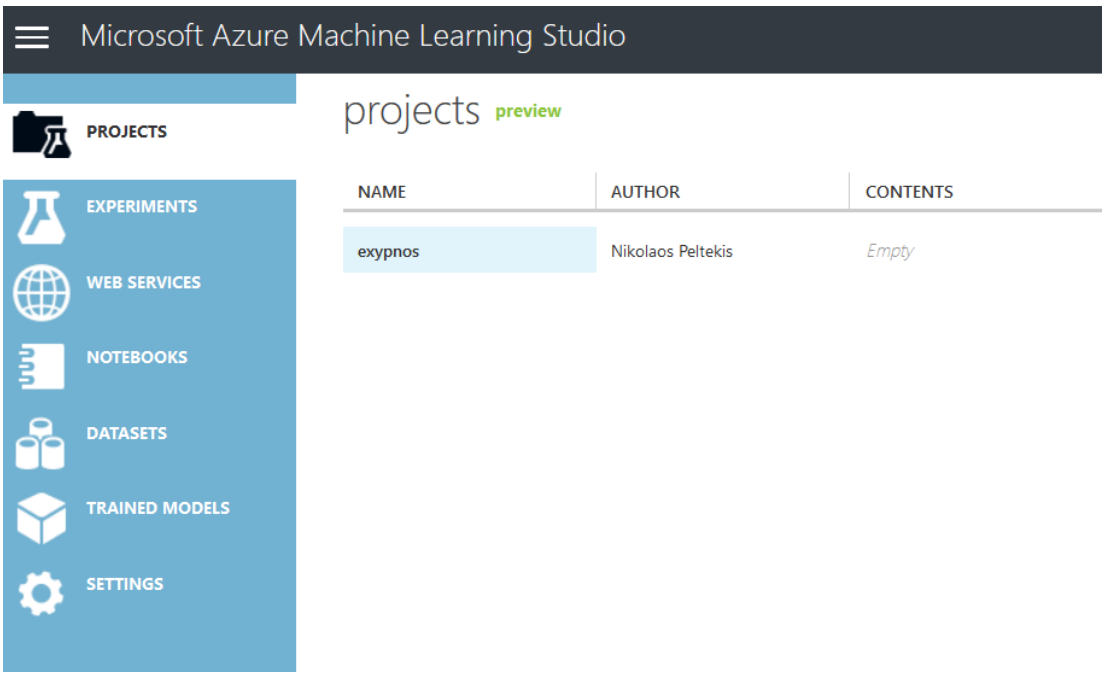
Η πλήρης ονομασία του είναι Azure Machine Learning Studio και προσφέρεται από το Azure είτε δωρεάν (και μάλιστα με επιλογή για Guest) είτε με κάποια συνδρομή που υπάρχει στο Azure. Το portal σύνδεσης βρίσκεται στην διεύθυνση: https://studio.azureml.net/ είτε μέσω του Azure portal (https://portal.azure.com)

Σύμφωνα με τη Microsoft, το Azure ML Studio είναι ένα εργαλείο συνεργασίας, με δυνατότητα drag-and-drop και με το οποίο μπορούμε να χτίσουμε, δοκιμάσουμε και εκτελέσουμε λύσεις επεξεργασίας και ανάλυσης των δεδομένων μας. Επίσης, παρέχει την δυνατότητα να δημιουργήσουμε web services από τα μοντέλα μας ή να εξάγουμε τα αποτελέσματα σε Business Intelligence εφαρμογές (Power BI, Excel) [17]. Συνοπτικά, βλέπουμε πως λειτουργεί, με το παρακάτω σχήμα (εικόνα 22):



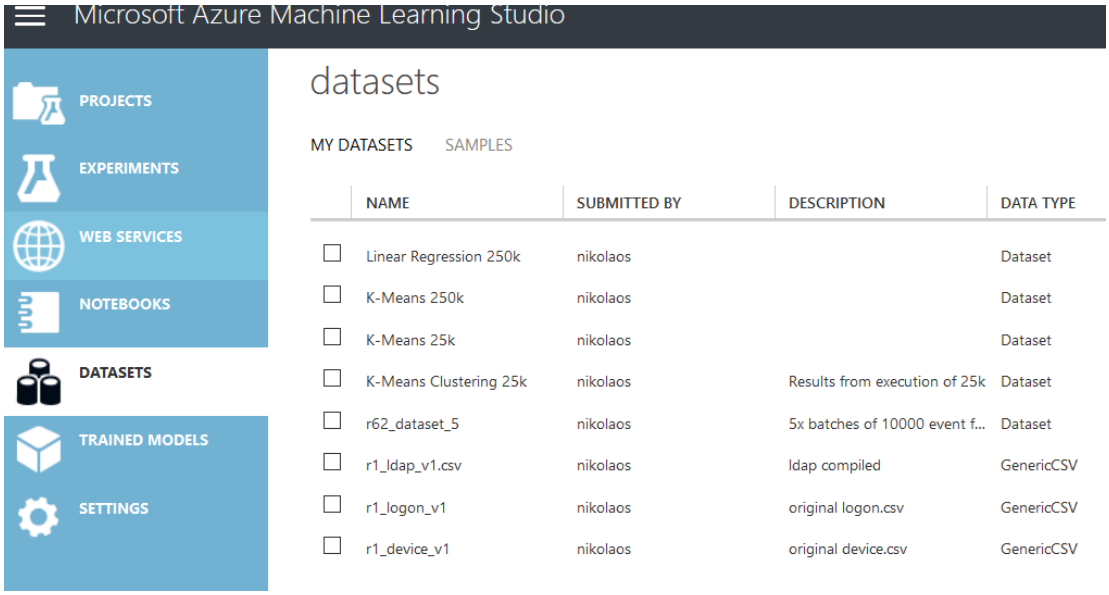
Εικόνα 22: Διάγραμμα υπηρεσιών του Azure ML Studio

Στο κεντρικό μενού, βλέπουμε ότι έχουμε projects, πειράματα, web services, jupyter notebooks, datasets και trained models (εικόνα 23):



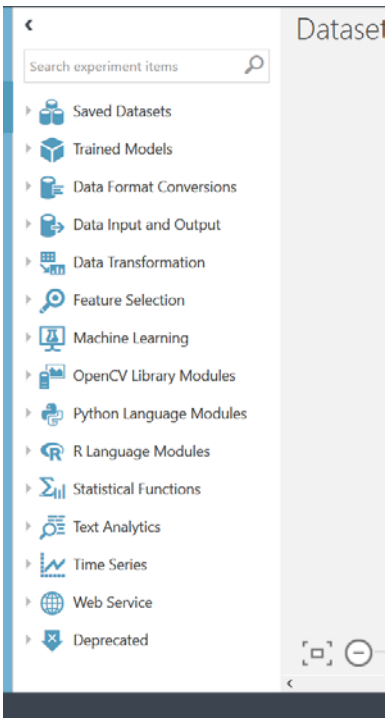
Εικόνα 23: Azure ML Studio: κατάλογος με τα projects

Μπορούμε να εισάγουμε τα δεδομένα, είτε ανεβάζοντας τα στην πλατφόρμα είτε σαν csv αρχεία, είτε με τη χρήση SQL Server ή Azure Storage (εικόνα 24).



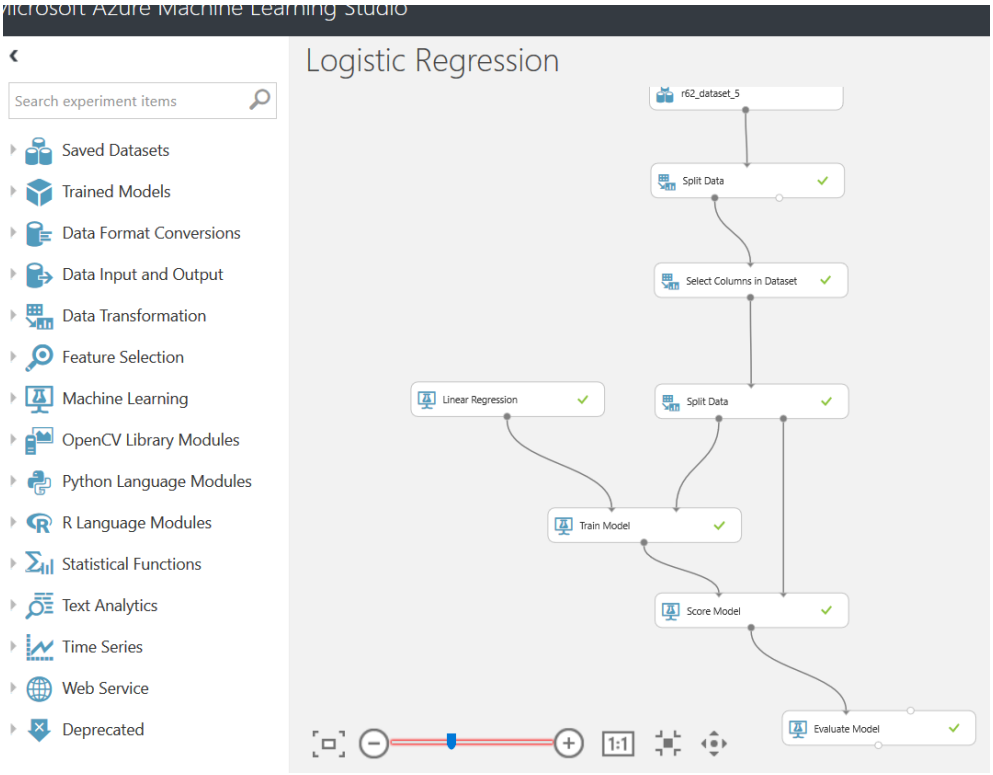
Εικόνα 24: Azure ML Studio: κατάλογος των datasets

Στη συνέχεια, δημιουργούμε ένα πείραμα, από τα modules που είναι οργανωμένα σε κατηγορίες (εικόνα 25):



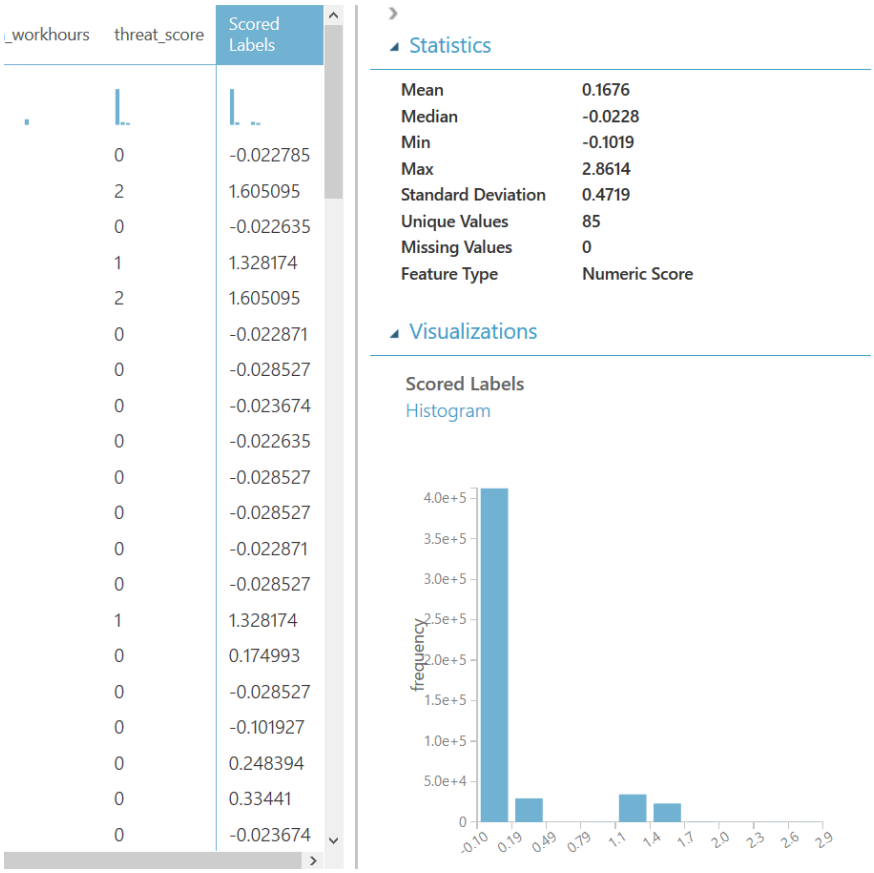
Εικόνα 25: Azure ML Studio: κατάλογος με τα modules

Επιλέγοντας τα κατάλληλα, μπορούμε να δημιουργήσουμε ένα μοντέλο χωρίς να γράψουμε ούτε μία γραμμή κώδικα (εικόνα 26):



Εικόνα 26: Azure ML Studio: δημιουργία του μοντέλου του Linear Regression

Εξηγούμε την παραπάνω εικόνα 26, χωρίς να εισέλθουμε σε λεπτομέρειες, ώστε να αντιληφθούμε τις δυνατότητες αυτού του συστήματος: Παρουσιάζουμε ένα πείραμα, όπου χρησιμοποιήσαμε τα δεδομένα από το r62_dataset_5, επιλέξαμε ποια πεδία θα συμμετέχουν στο πείραμα (Select Columns in Dataset), τα διαχωρίσαμε με το Split Data module με ένα ratio 0.75 (το 75% χρησιμοποιείται για εκπαίδευση του μοντέλου και το 25 για αξιολόγηση). Εδώ επιλέχθηκε ο αλγόριθμος Linear Regression για την εκπαίδευση. Στη συνέχεια, χρησιμοποιούμε το Score Model για να αξιολογήσουμε το υπόλοιπο των δεδομένων (25%) και επιλέγοντας την οπτικοποίηση των αποτελεσμάτων (δεξί κλικ -> Visualize), βλέπουμε τον παρακάτω πίνακα (εικόνα 27):



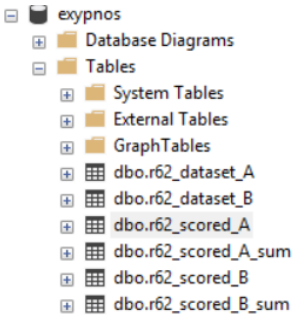
Εικόνα 27: Azure ML Studio: Αποτελέσματα βαθμολόγησης Linear Regression

Η στήλη scored labels είναι το αποτέλεσμα που υπολόγισε ο αλγόριθμος, και βλέπουμε τις τιμές του σε αντιπαραβολή με τον δικό μας υπολογισμό του threat_score.

4.4.1 Cert r6.2 Dataset στο Azure ML Studio

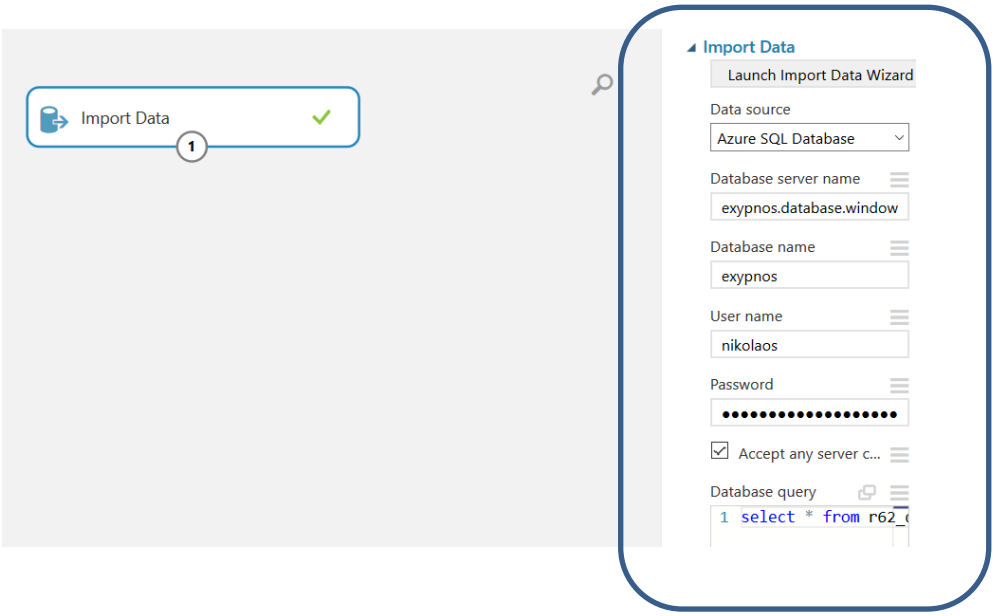
Το Cert r6.2 dataset, όπως είδαμε, αποτελείται από αρχεία csv μεγάλου μεγέθους που καταλαμβάνουν χώρο στο δίσκο περίπου 96Gb ενώ το αντίστοιχο επεξεργασμένο αρχείο εξόδου που θα δημιουργούσαμε, το events.csv, υπολογίζεται περίπου στα 23Gb.

Χρησιμοποιώντας το DPM σε Python, βαθμολογήσαμε τα δεδομένα σύμφωνα με την Πολιτική Α και τα εισάγαμε στην Azure SQL, στον πίνακα r2_dataset_A. Χρησιμοποιώντας το πλεονέκτημα του Azure Studio ML να μπορεί να λαμβάνει δεδομένα και από άλλες υπηρεσίες του Azure (Storage, SQL), εισάγαμε τα δεδομένα στο Azure SQL, στη βάση exypnos και στο table r62_dataset_A:



και στη συνέχεια ρυθμίσαμε το Import Module να τα εισάγει απευθείας, εκτελώντας το query (εικόνα 28):

```
SELECT * FROM r62_dataset_A
```



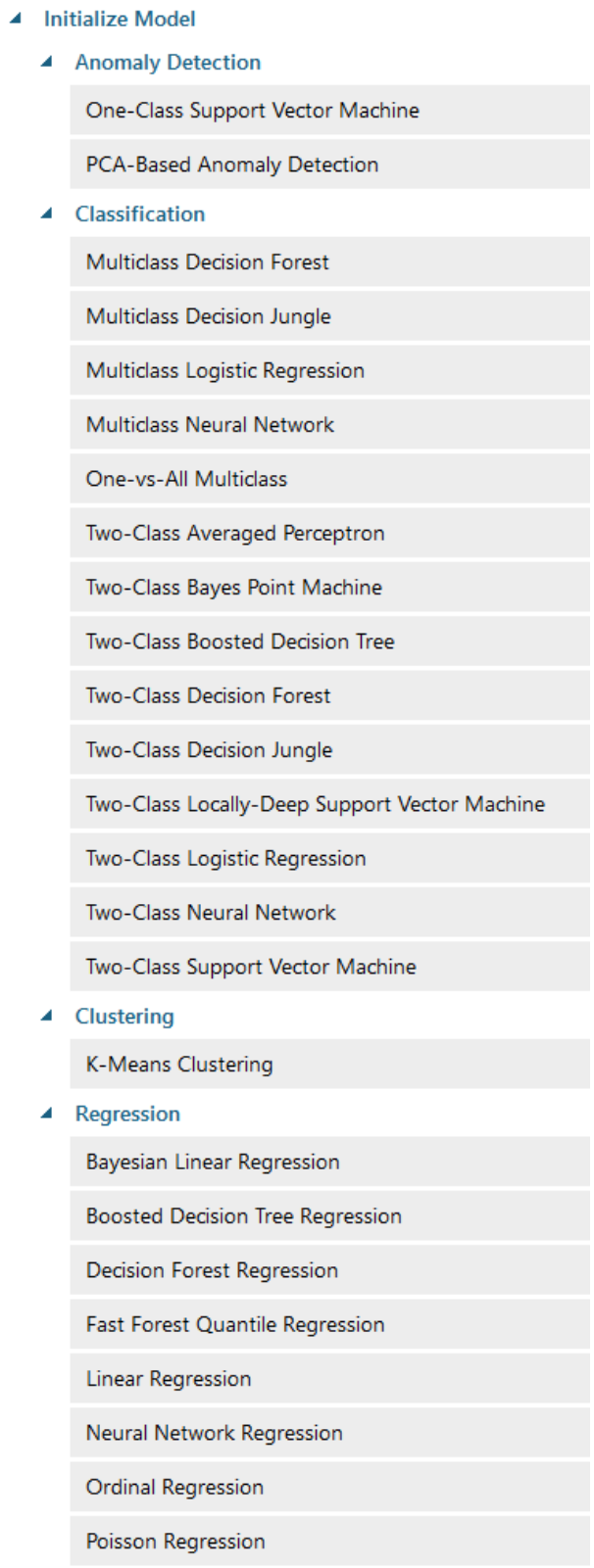
Εικόνα 28: Azure ML Studio: Εισαγωγή δεδομένων από SQL Server

Το αποτέλεσμα το αποθηκεύουμε σαν Dataset με όνομα r62_dataset_A. Στην συνέχεια κάνουμε το ίδιο και για την Πολιτική Β και το dataset θα ονομάζεται r62_dataset_B και έχουμε τη δυνατότητα να τα χρησιμοποιήσουμε σε όποιο ή όποια πειράματα θέλουμε, χωρίς περιορισμό.

4.5 Αλγόριθμοι ΤΝ στο Azure ML Studio

Προηγουμένως αναφερθήκαμε στο γεγονός ότι μπορούμε να φτιάξουμε μοντέλα Machine Learning και Deep Learning στο Azure ML Studio χωρίς να γράψουμε ούτε μία γραμμή κώδικα. Η πιο ολοκληρωμένη απάντηση είναι ότι δεν είναι απαραίτητο να γνωρίζουμε κάποια γλώσσα προγραμματισμού αλλά αν συμβαίνει αυτό και θέλουμε να δώσουμε επιπλέον λειτουργικότητα στο μοντέλο μας, μπορούμε να χρησιμοποιήσουμε είτε Python είτε R. Παρόλα αυτά, κάποιος για να δημιουργήσει το σωστό μοντέλο, πρέπει να μπορεί να γνωρίζει τους αλγόριθμους που υπάρχουν στον τομέα της ΤΝ και να αντιλαμβάνεται πως λειτουργούν ώστε να μπορέσει να τους παραμετροποιήσει κατά τον πειραματισμό του.

Η πλατφόρμα διαθέτει αρκετά μεγάλη ποικιλία αλγορίθμων ΤΝ που μπορούν να χρησιμοποιηθούν στα πειράματα μας (εικόνα 29):



Εικόνα 29: Azure ML Studio: Κατάλογος με τους υλοποιημένους αλγόριθμους

4.6 Υλοποίηση πειραμάτων στο Azure ML Studio

4.6.1 Linear Regression

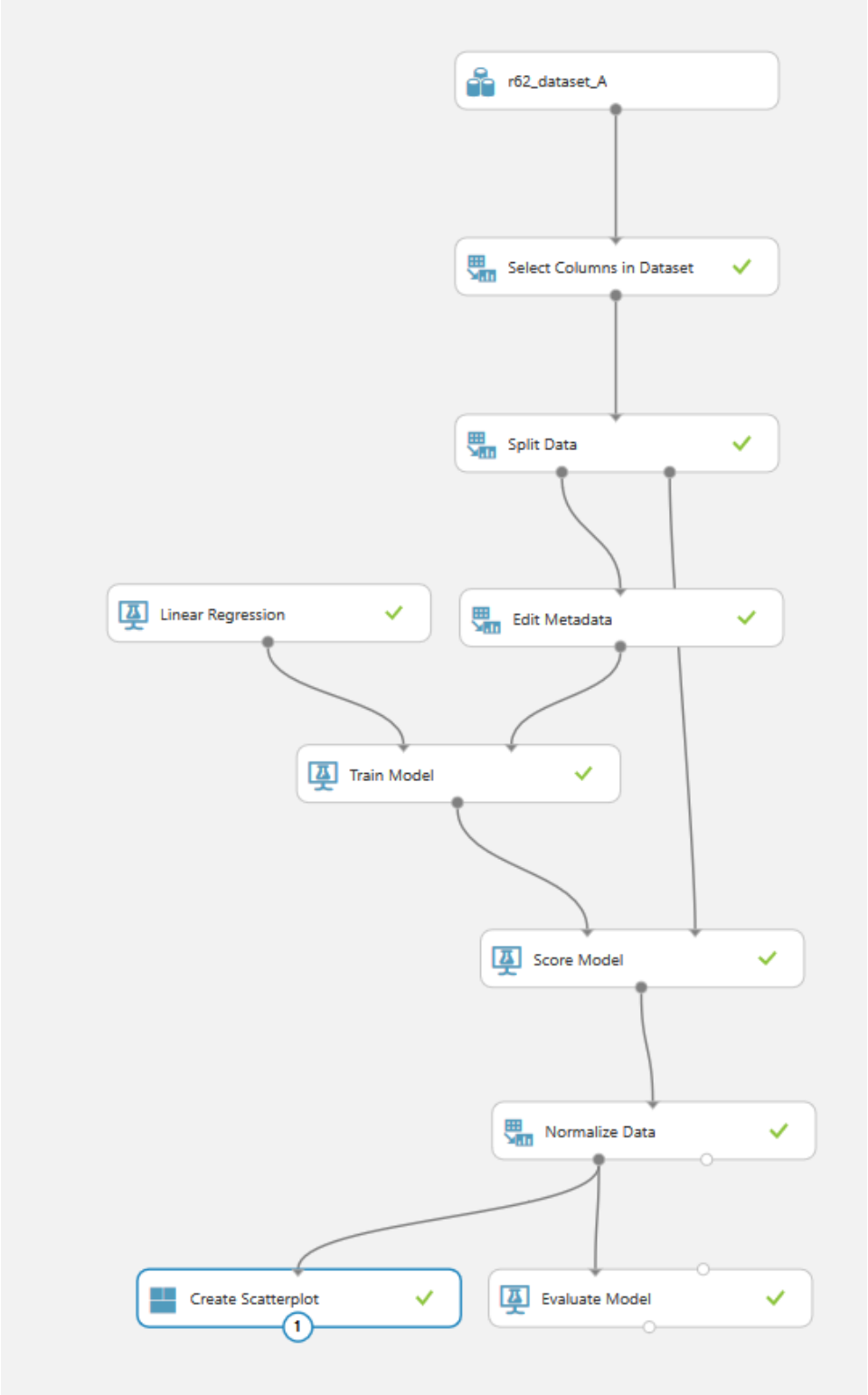
Όπως είδαμε, ο Linear Regression είναι αλγόριθμος που ασχολείται με δεδομένα γραμμικά και επειδή ανήκει στην κατηγορία των Regression αλγορίθμων, προσπαθεί να προβλέψει μια αριθμητική τιμή.

Σκοπός του πειράματος είναι το μοντέλο που θα δημιουργηθεί, να υπολογίζει τη στήλη total_threat_score χωρίς να χρειαστεί να δηλώσουμε κάπου τον μαθηματικό τύπο. Θα χρησιμοποιήσουμε το 75% του dataset μας για να εκπαιδεύσουμε το μοντέλο μας (train_data) και στη συνέχεια θα ζητήσουμε να υπολογίζει το threat_score στο υπόλοιπο 25% (test_data).

Πρώτα, θα εφαρμόσουμε τον αλγόριθμο, στο dataset το οποίο είναι βαθμολογημένο για την Πολιτική Ασφάλειας Α. Τα modules που θα χρειαστούμε είναι:

1. r62_dataset_A
2. Select Columns in Dataset
3. Split Data,
4. Edit Metadata,
5. Linear Regression,
6. Train Model,
7. Score Model,
8. Normalize Data,
9. Evaluate Model,
10. Create Scatterplot,

Το σχεδιάγραμμα θα είναι όπως παρακάτω (εικόνα 30):



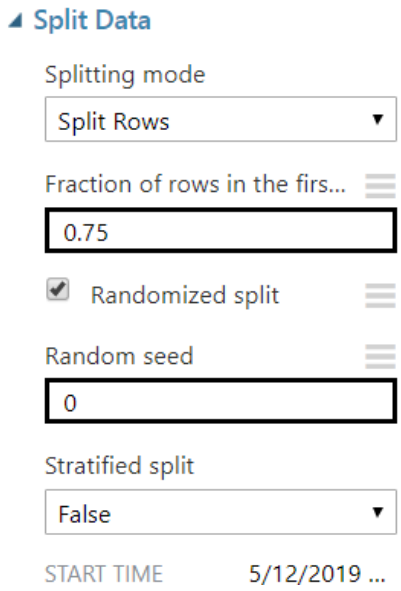
Εικόνα 30: Azure ML Studio: Ο Linear Regression στο πείραμα βαθμολόγησης

Τα βήματα, επεξηγούνται παρακάτω:

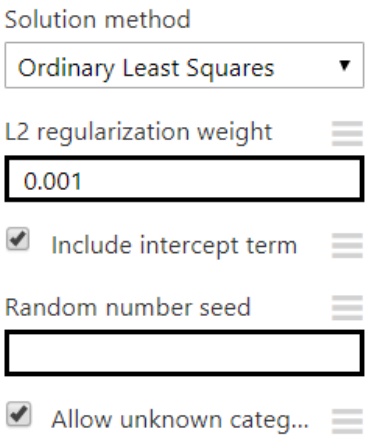
1. Εισαγωγή του r62_dataset_A, το οποίο είναι βαθμολογημένο σύμφωνα με τη Πολιτική ασφαλείας A,

2. Με το Select Columns in Dataset, επιλέγουμε τα πεδία τα οποία θα είναι και τα features του αλγόριθμου,

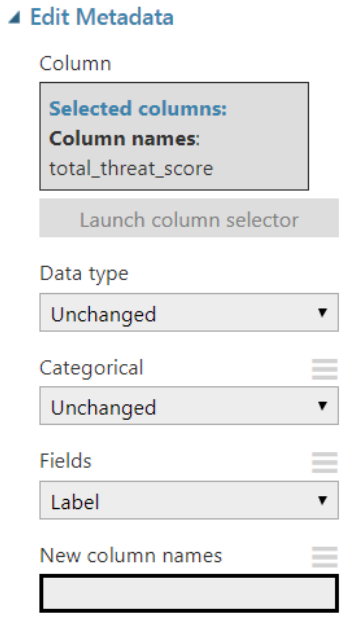
3. Με το Split Data, κάνουμε το διαχωρισμό του Dataset, όπου το 75% θα διατεθεί για εκπαίδευση και το 25% για test data,



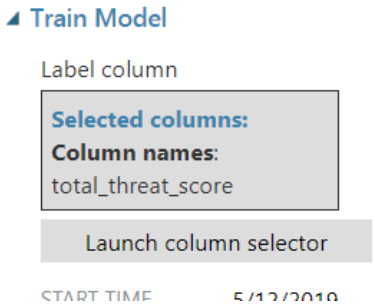
4. Το Linear Regression module, περιέχει τις παραμέτρους που μπορούμε να μεταβάλλουμε ώστε να πλησιάσουμε όσο πιο κοντά γίνεται, και κατά προσέγγιση, σε ένα ολοκληρωμένο μοντέλο. Έχει δύο επιλογές για error calculation: Ordinary Least Squares method και Gradient Descent. Εμείς θα επιλέξουμε το Ordinary Least Squares method,



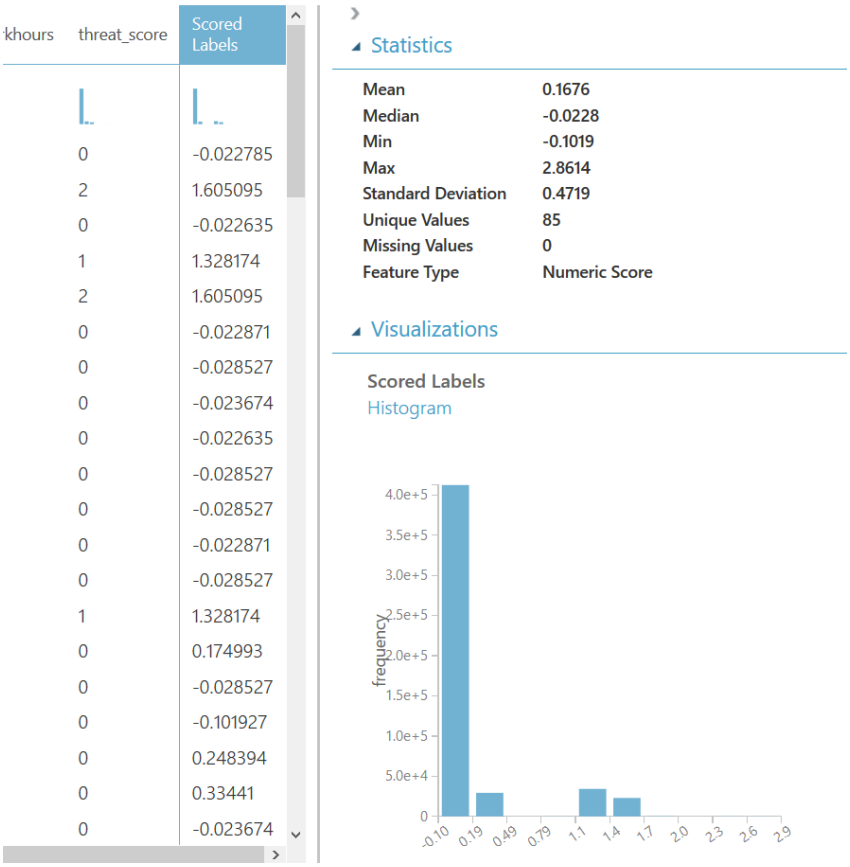
5. Με το Edit Metadata, ορίζουμε τη στήλη total_threat_score σαν label, ώστε να χρησιμοποιηθεί για την εκπαίδευση του αλγόριθμου (είναι το πεδίο στόχος για τον αλγόριθμο),



6. Το Train Model, δέχεται σαν παράμετρο την στήλη που είναι label. Δέχεται τα δεδομένα από το Edit Metadata και συνδέεται με το Linear Regression module.



7. Το Score Model συνδέει το εκπαιδευμένο μας μοντέλο και τα test data. Επιλέγουμε δεξί κλικ -> Visualize για να ελέγξουμε τα αποτελέσματα του πειράματος (εικόνα 31).

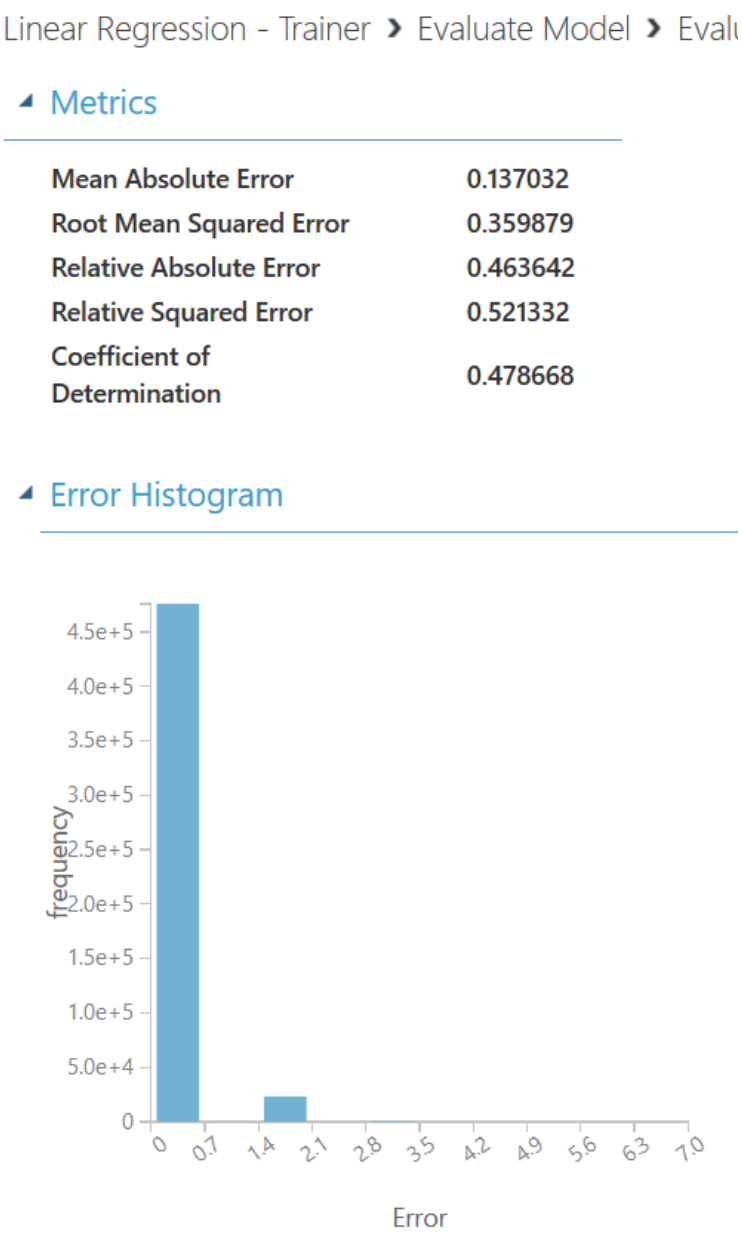


Εικόνα 31: Azure ML Studio: Αποτελέσματα βαθμολόγησης Linear Regression

Βλέπουμε ότι προστίθεται η στήλη Scored Labels η οποία περιέχει τους υπολογισμούς του μοντέλου μας με τα δεδομένα που του δώσαμε. Σημειώνουμε ότι δεν του δόθηκε ο τύπος υπολογισμού του threat_score, ούτε κατά την εκπαίδευση αλλά ούτε και κατά την επαλήθευση.

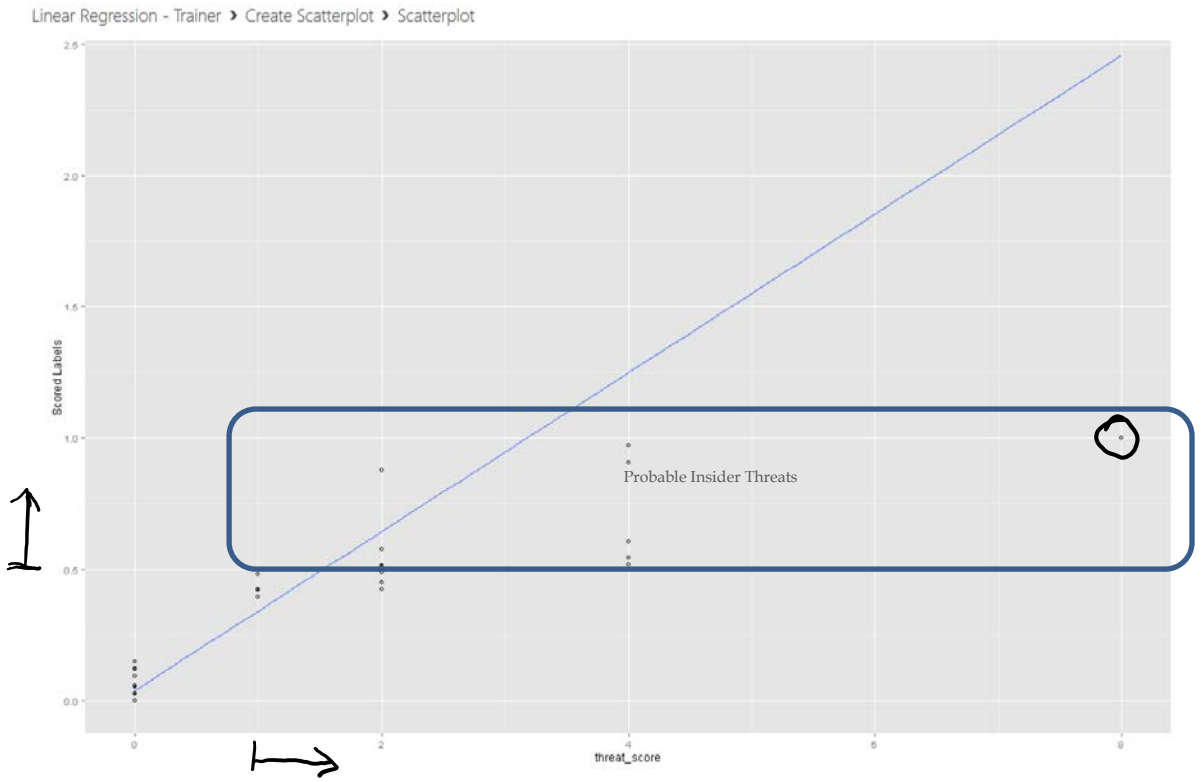
8. Το Normalize Data, χρησιμοποιεί τον αλγόριθμο MinMax ή Sigmoid function ώστε να περιορίσει το εύρος των τιμών μεταξύ [0,1]. Εμείς, επιλέξαμε τον MinMax.

9. Το module Evaluate Model, μας εμφανίζει κάποια στατιστικά (εικόνα 32) από το πείραμα (θα αναλυθούν παρακάτω):



Εικόνα 32: Azure ML Studio: Αποτίμηση του πειράματος με τον Linear Regression για την πολιτική A

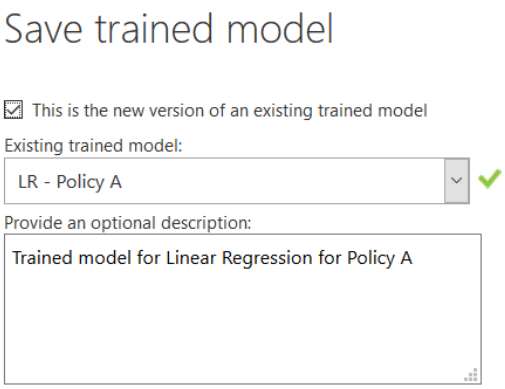
10. Το Create Scatter Plot, είναι ένα εξωτερικό πρόσθετο που χρησιμοποιείται για την προβολή της σύγκρισης των πεδίων threat_score (οι δικοί μας υπολογισμοί) και Scored_Labels (οι υπολογισμοί του μοντέλου) (εικόνα 33).



Εικόνα 33: Γράφημα αποτελεσμάτων πειράματος βαθμολόγησης με Linear Regression για την πολιτική A

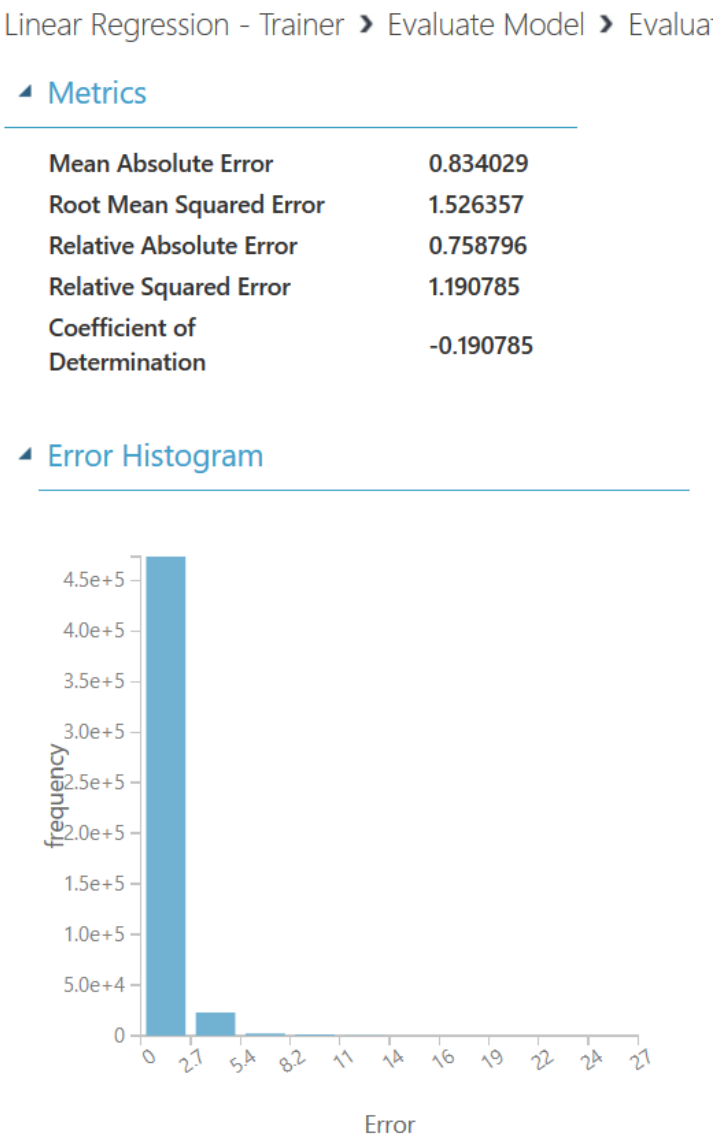
Και τα δύο πεδία, συμφωνούν, ότι ένα γεγονός που εμείς το βαθμολογήσαμε με 8, το σύστημα το βαθμολόγησε με 1 (το μέγιστο, εφόσον τα δεδομένα είναι κανονικοποιημένα). Παρατηρούμε ότι τα αποτελέσματα μας, ακολουθούν τη γραμμή Linear Regression του γραφήματος μας, ενώ ξεχωρίζουν τα data τα οποία θα μπορούσαν να χαρακτηριστούν ως anomalies.

Στο σημείο αυτό, μπορούμε να αλλάξουμε τις παραμέτρους του module Linear Regression (στο βήμα 4) είτε να αποθηκεύσουμε το μοντέλο για χρήση σε άλλα πειράματα, ως εξής (δεξί κλικ στο Train Model):



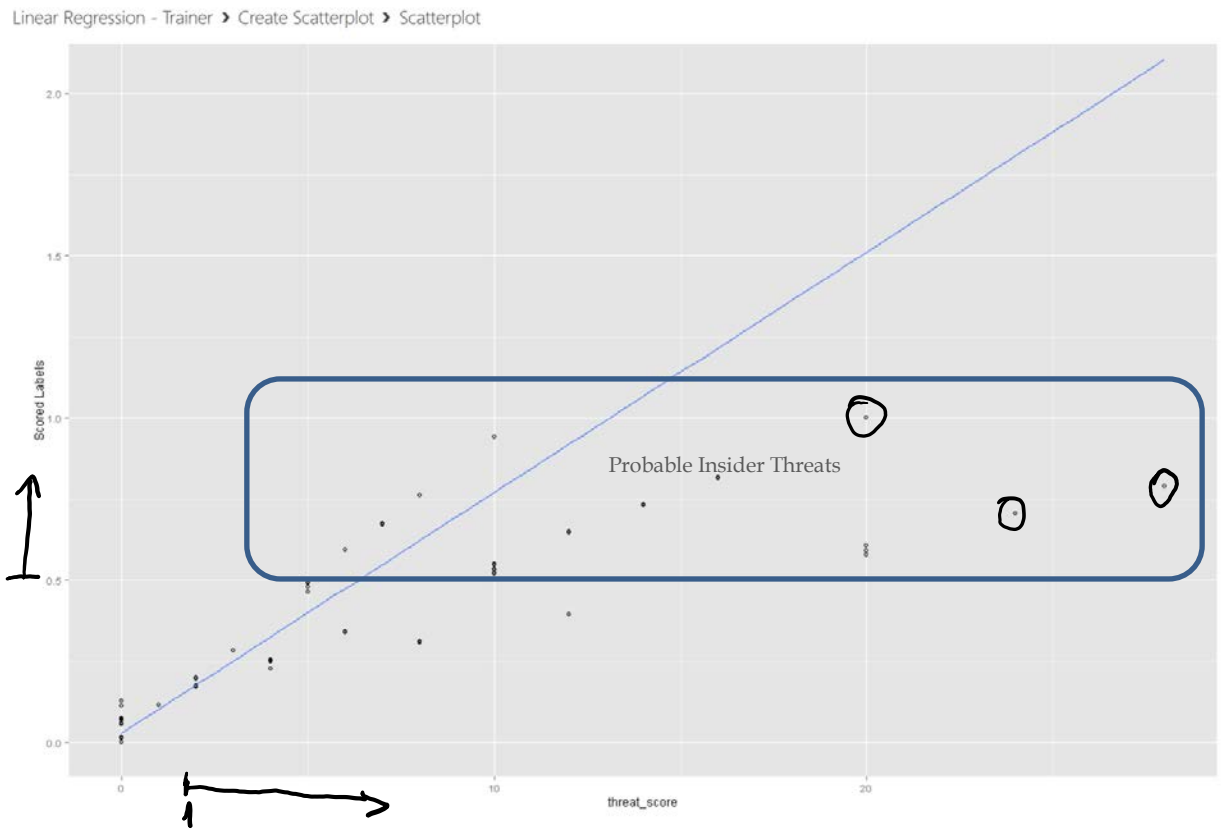
Για την πολιτική Β, διατηρούμε όλα τα modules και αλλάζουμε το module με το r62_dataset_B, το οποίο είναι βαθμολογημένο από το DPM μας για την Πολιτική Β. Σημειώνουμε ότι, οι πληροφορίες από το προηγούμενο πείραμα δεν απομνημονεύονται από τη στιγμή που θα ξεκινήσει το νέο πείραμα (Όταν εκτελούμε την εντολή Run).

Τα αποτελέσματα του Evaluation Module είναι τα εξής (εικόνα 34):



Εικόνα 34: Azure ML Studio: Αποτίμηση του πειράματος με τον Linear Regression για την πολιτική Β

Και το αντίστοιχο Scatter Plot είναι το παρακάτω (εικόνα 35):



Εικόνα 35: Γράφημα αποτελεσμάτων πειράματος βαθμολόγησης με Linear Regression για την πολιτική Β

Παρατηρούμε ότι ενώ υπάρχουν τιμές του threat_score μεγαλύτερες του 20 (πιθανές εσωτερικές απειλές), ο αλγόριθμος σε μερικές από αυτές συμφώνησε μαζί μας (και έδωσε την τιμή 1) ενώ στις υπόλοιπες έδωσε τιμές από 0.6 έως 0.8. Βέβαια, στις τιμές που εμείς βαθμολογήσαμε από 2 έως 10, το μοντέλο έδωσε αντίστοιχα τιμές από 0.1 έως 0.5, δηλαδή δεν τις θεώρησε απειλές.

Συγκρίνοντας τα αποτελέσματα του Evaluate module, βλέπουμε τα Mean Absolute Error, Root Mean Squared Error, Relative Absolute Error, Relative Squared Error και Coefficient of Determination. Την απόδοση του ίδιου αλγόριθμου και για τις δύο πολιτικές, μπορούμε να την δούμε στον παρακάτω πίνακα:

	Policy A	**Policy B**
Mean Absolute Error	0.137032	0.834029
Root Mean Squared Error	0.329879	1.526357
Relative Absolute Error	0.463642	0.758796
Relative Squared Error	0.521332	1.190785
Coefficient of Determination	0.478668	-0.190785

Με τον όρο error, εκφράζεται η απόκλιση και εδώ μας ενδιαφέρουν πάρα πολύ τα Relative Absolute Error και Relative Squared Error, τα οποία όσο πιο μικρότερες είναι οι τιμές τους, τόσο πιο ακριβές το μοντέλο στην πρόβλεψη του. Το Coefficient of Determination εξηγείται πόσο καλά το μοντέλο έχει ταιριάξει με τα δεδομένα. Το 1 είναι το ιδανικό. Επομένως, ο Linear Regression για την πολιτική Α είχε πιο καλή απόδοση απ' ότι για την πολιτική Β, αλλά κανένας από τους δύο δεν πλησίασε αρκετά στο perfect fit ενώ και οι αποκλίσεις είναι αρκετά υψηλές (οι αναμενόμενες τιμές πρέπει να είναι κάτω του 0.20). [18]

4.6.2 Ενδιάμεσο Βήμα Επεξεργασίας

Τα αποτελέσματα των εκτελέσεων των δύο Linear Regression, εξάγονται στη βάση Azure SQL, σε δύο νέους πίνακες r62_scored_A και r62_scored_B. Περιλαμβάνουν τα δεδομένα με τη δική μας βαθμολόγηση και επιπλέον, με τη βαθμολόγηση από τα μοντέλα.

Στη βάση, εκτελούνται τα παρακάτω queries:

1. SELECT DATEname(dd,date) AS event_day, DATEname(mm,date) as event_month, DATEname(yyyy,date) AS event_year, usr, COUNT(*) AS NumEvents, sum(threat_score) as total_policy_threat_score_per_day, sum(Scored_Labels) as total_calculated_score_per_day, sum(Scored_Labels)/COUNT(*) as mean_of_calculated_threat_score into r62_scored_A_sum FROM r62_scored_A
2. SELECT DATEname(dd,date) AS event_day, DATEname(mm,date) as event_month, DATEname(yyyy,date) AS event_year, usr, COUNT(*) AS NumEvents, sum(threat_score) as total_policy_threat_score_per_day, sum(Scored_Labels) as total_calculated_score_per_day, sum(Scored_Labels)/COUNT(*) as mean_of_calculated_threat_score into r62_scored_B_sum FROM r62_scored_B

Οι πίνακες που παράγονται, έχουν την παρακάτω μορφή (εικόνα 36):

Results Messages

	event_day	event_month	event_year	usr	NumEvents	total_policy_threat_score_per_day	total_calculated_score_per_day	mean_of_calculated_threat_score
1	1	April	2010	AAC0610	9	0	0,333487272906212	0,0370541414340236
2	1	April	2010	AAD3030	1	0	0,0266781470844753	0,0266781470844753
3	1	April	2010	AAF0819	26	0	0,694245573496356	0,0267017528267829
4	1	April	2010	AAP0352	4	0	0,106329335042963	0,0265823337607408
5	1	April	2010	AAP1919	23	0	0,705611029400177	0,0306787404087033
6	1	April	2010	AAP1942	2	0	0,0533562941689506	0,0266781470844753
7	1	April	2010	AAS2987	1	0	0,0266781470844753	0,0266781470844753
8	1	April	2010	AAS3428	1	0	0,0266781470844753	0,0266781470844753
9	1	April	2010	ABB0167	1	0	0,0266781470844753	0,0266781470844753
10	1	April	2010	ABD3426	43	0	1,1481551782467	0,0267012832150396
11	1	April	2010	ABH0349	30	0	0,802524055328559	0,0267508018442853
12	1	April	2010	ABK3081	8	0	0,213725515416561	0,0267156894270702
13	1	April	2010	ABM3687	29	0	0,775668254856506	0,0267471812019485
14	1	April	2010	ABM3772	6	0	0,347323942635281	0,0578873237725468
15	1	April	2010	ABO1173	7	0	0,186844824994134	0,0266921178563048
16	1	April	2010	ABP2003	1	0	0,0266781470844753	0,0266781470844753

Εικόνα 36: Δείγμα από τον πίνακα με τα βαθμολογημένα δεδομένα στον SQL Server

Δημιουργούνται τα πεδία NumEvents, total_policy_threat_score_per_day, total_calculated_score_per_day, mean_of_calculated_threat_score, εκ των οποίων:

NumEvents	Είναι το πλήθος των ενεργειών ενός χρήστη, ανά ημέρα
total_policy_threat_score_per_day	Είναι το άθροισμα των total_score των παραπάνω ενεργειών
total_calculated_score_per_day	Είναι το άθροισμα των Scored_Labels (η βαθμολογία των μοντέλων μας)
mean_of_calculated_threat_score	Είναι ο μέσος όρος του threat_score των μοντέλων μας, υπολογιζόμενος ως: $$(\text{mean of calculated threat score}) = \frac{\text{total calculated score per day}}{NumEvents}$$

Η λογική για τον υπολογισμό του mean_of_calculated_threat_score είναι η εξής:

Σε ένα σύνολο *n* ενεργειών, με άθροισμα της βαθμολογίας τους *m*, ο λόγος $\frac{m}{n}$ εκφράζει τη μέση βαθμολογία της ημέρας στο σύνολο των ενεργειών. Αν το n=1 και m=1, τότε $\frac{m}{n} = 1$, τότε η ενέργεια θεωρείται κανονική. Γενικά, κάθε ενέργεια με βαθμολογία =< 1 (πολιτική Α και Β),

θεωρείται ως κανονική, αλλιώς πιθανή απειλή. Αν *n=25* και *m=30*, τότε $\frac{30}{25} = 1.2 > 1$ και θεωρείται πιθανή απειλή. Με αυτόν τον τρόπο, το σύστημα διακρίνει εάν ένας χρήστης με πολλές ενέργειες σε μια ημέρα και με υψηλή συνολική βαθμολογία (πχ ITAdmin), είναι πιθανή απειλή ή όχι. Στο συγκεκριμένο υπολογισμό, δεν θα χρησιμοποιήσουμε τη δική μας βαθμολόγηση των ενεργειών αλλά των δύο μοντέλων (total_calculated_score_per_day).

Στο σημείο αυτό, πρέπει να τονίσουμε ότι τα πεδία total_calculated_score_per_day, mean_of_calculated_threat_score είναι στην κανονικοποιημένη τους μορφή, γεγονός που οδηγεί στην ύπαρξη δεκαδικών αριθμών. Πλέον, ο λόγος $\frac{m}{n}$ πρέπει να χρησιμοποιηθεί και αυτός στην κανονικοποιημένη του μορφή και έτσι αν η βαθμολόγηση είναι >0.5, τότε έχουμε την ύπαρξη πιθανής εσωτερικής απειλής. Αυτό είναι ένα στοιχείο που θα χρησιμοποιήσουμε στη συνέχεια.

4.6.3 One-Class Support Vector Machine

Ο One-Class Support Vector Machine ανήκει στην κατηγορία των Anomaly Detection αλγορίθμων, και εδώ θα ορίσουμε κάποιο μέρος των δεδομένων ως normal data και τα υπόλοιπα θα χρησιμοποιηθούν για anomaly detection:

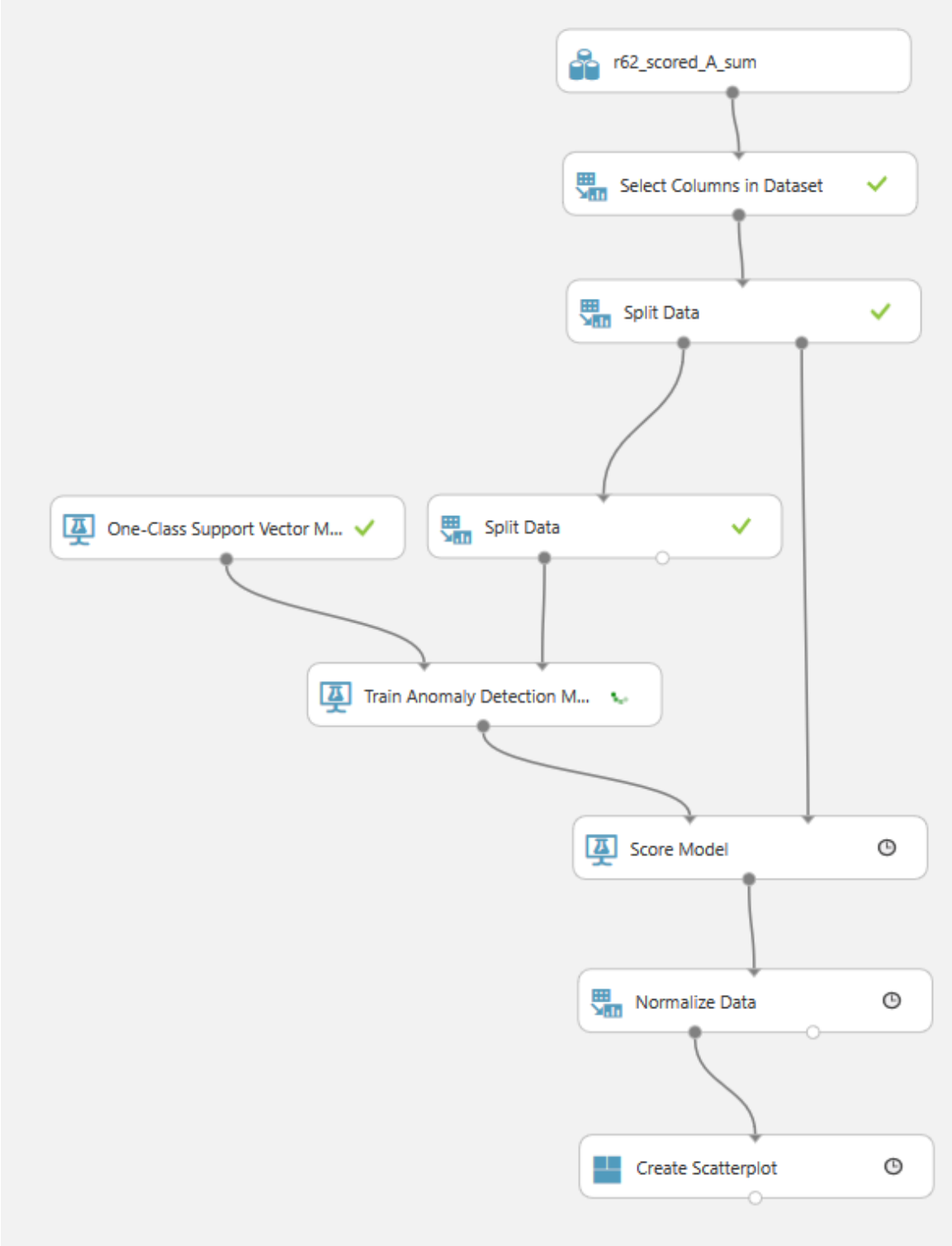
Τα modules που θα χρειαστούμε είναι (εδώ θα παρουσιάσουμε το μοντέλο για την πολιτική Α, αλλά τα βήματα είναι τα ίδια και για την πολιτική Β):

1. r62_scored_A_sum,
2. Select Columns in Dataset,
3. Split Data,
4. Split Data,
5. One-Class Support Vector Machine,
6. Train Anomaly Detection Module,
7. Score Model,

8. Normalize Data,

9. Create Scatterplot,

Το σχεδιάγραμμα θα είναι όπως παρακάτω (εικόνα 37):



Εικόνα 37: Azure ML Studio: Ο OC-SVM στο πείραμα βαθμολόγησης

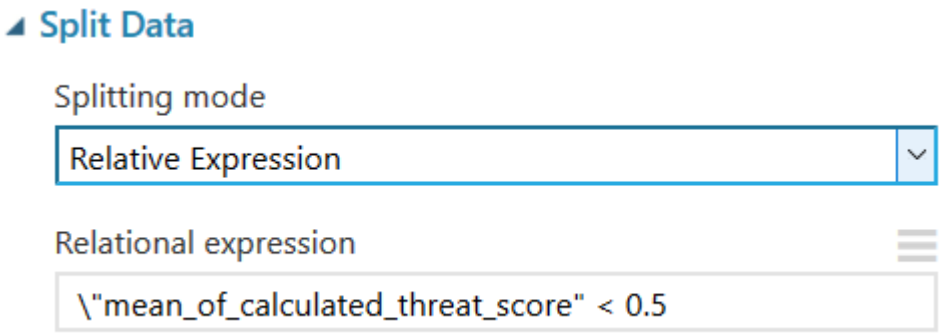
Τα βήματα, επεξηγούνται παρακάτω:

1. Εισαγωγή του r62_dataset_A_sum,

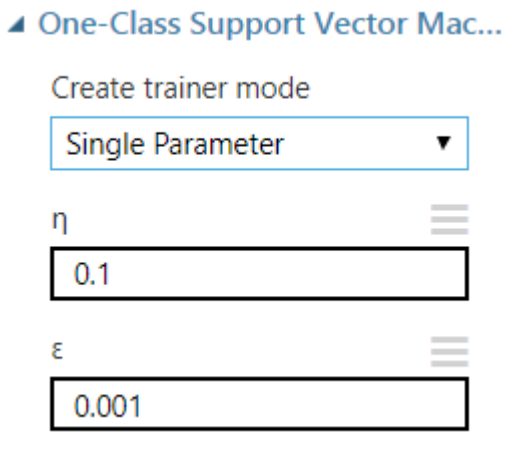
2. Με το Select Columns in Dataset, επιλέγουμε τα πεδία τα οποία θα είναι και τα features του αλγόριθμου,

3. Με το Split Data, κάνουμε το διαχωρισμό του Dataset, όπου το 75% θα διατεθεί για εκπαίδευση και το 25% για test data,

4. Με το 2ο Split Data, εφαρμόζουμε ένα relative expression ώστε να διαχωρίσουμε τα normal data από τα υπόλοιπα (αναφερθήκαμε γι' αυτό στο τέλος της προηγούμενης υποενότητας). Στην ουσία ορίζουμε στον αλγόριθμο την κλάση με τα normal data. Εδώ επιλέγουμε το mean_of_calculated_threat_score < 0.5,

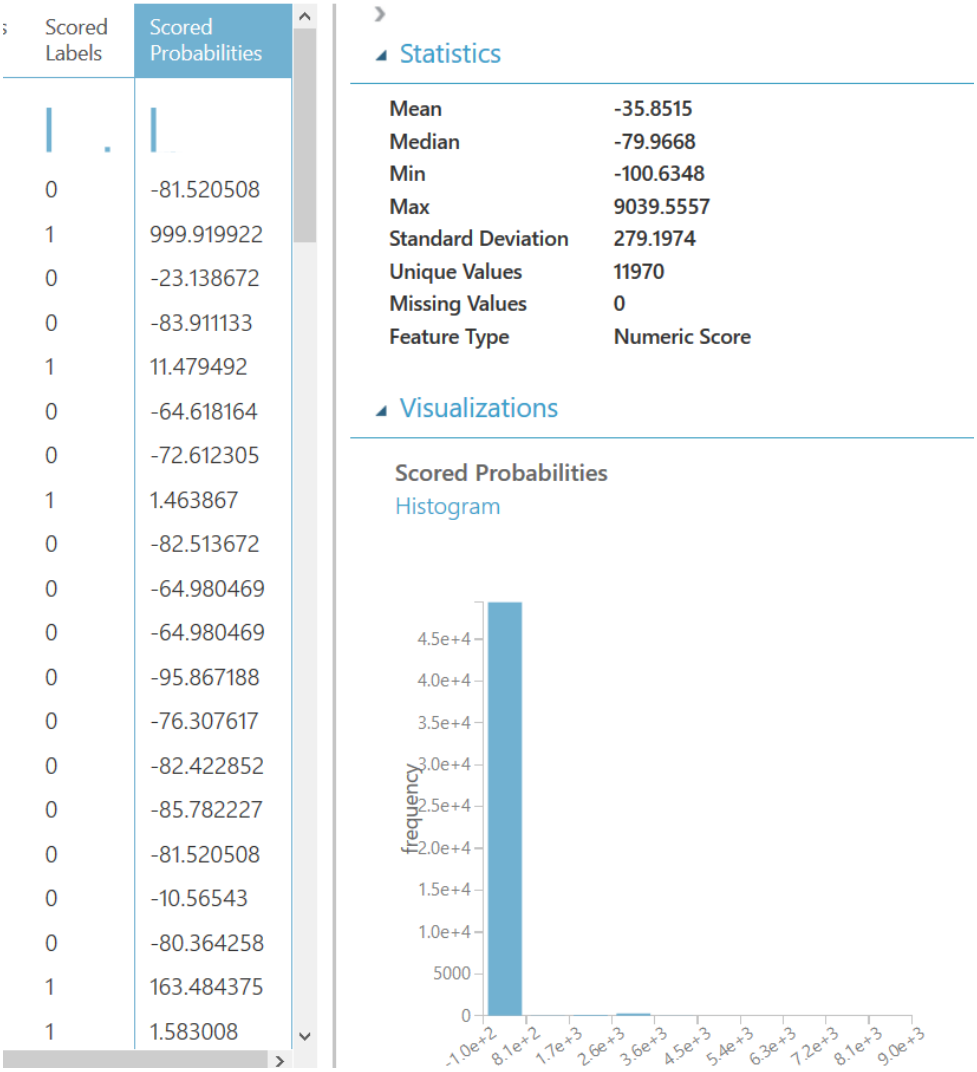


5. Το One-Class Support Vector Machine module, περιέχει τις παραμέτρους που μπορούμε να μεταβάλλουμε ώστε να πλησιάσουμε όσο πιο κοντά γίνεται, και κατά προσέγγιση, σε ένα ολοκληρωμένο μοντέλο. Τα προκαθορισμένα είναι επαρκή,



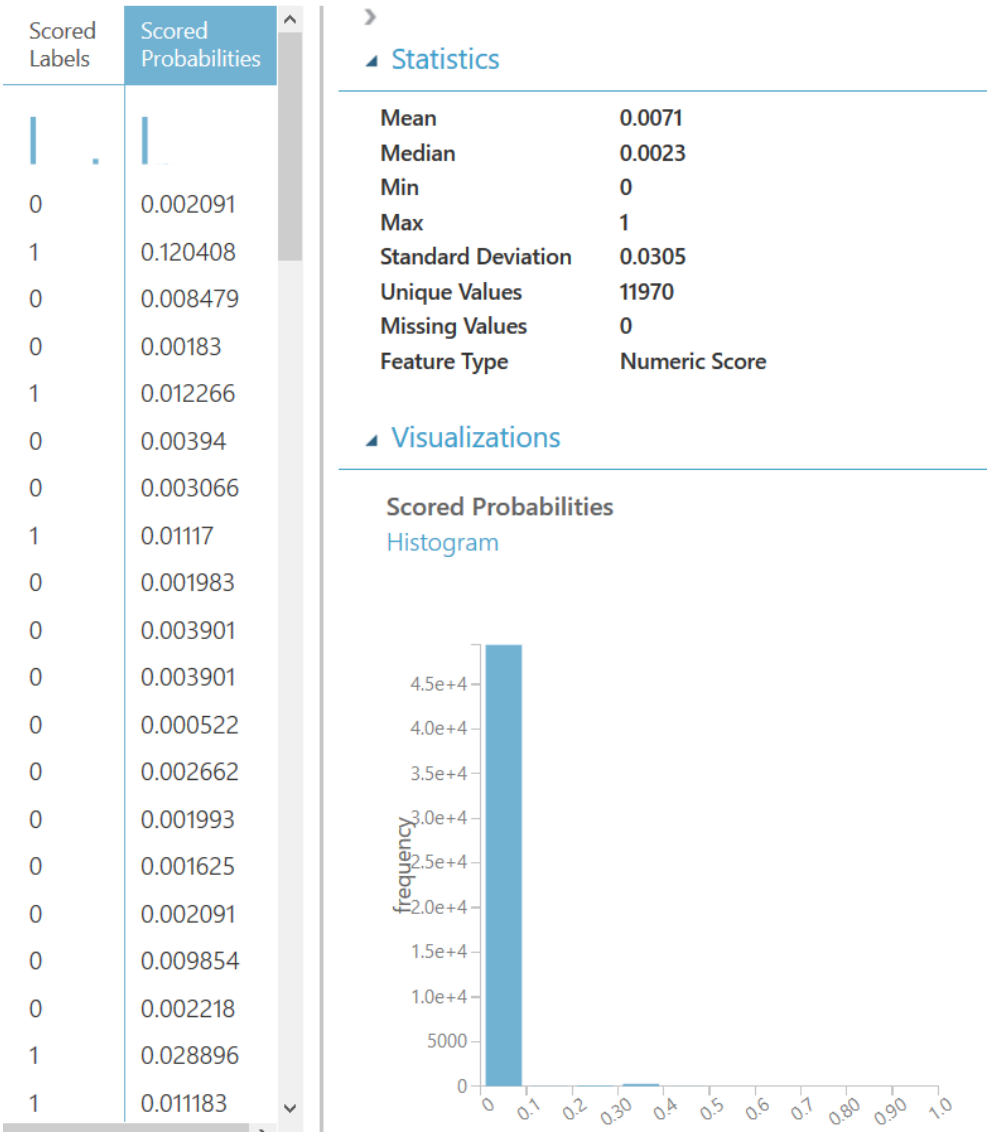
6. Το Train module δεν δέχεται κάποια παράμετρο, απλά εκπαιδεύει το μοντέλο,

7. Με το Score Model, βλέπουμε τα αποτελέσματα του πειράματος (εικόνα 38),



Εικόνα 38: Azure ML Studio: Αποτελέσματα βαθμολόγησης OC-SVM

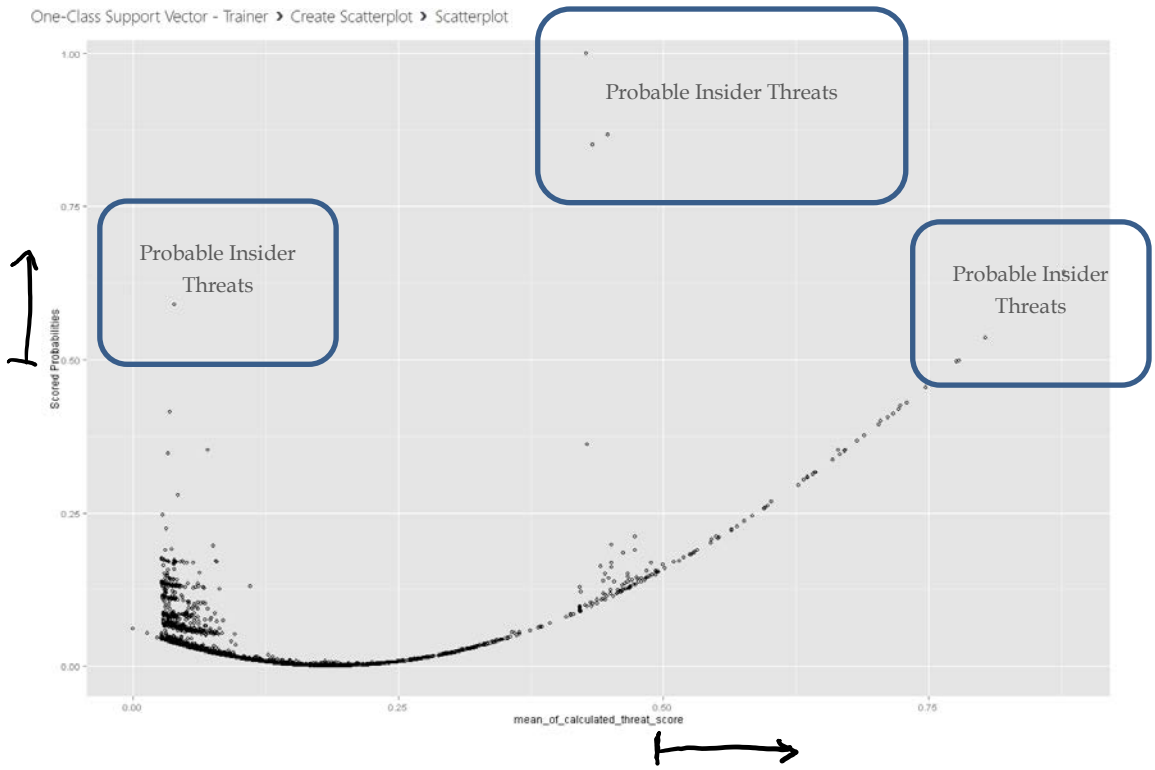
8. Με το Normalize Data, αλλάζουμε το εύρος των αποτελεσμάτων ώστε να βρίσκεται μεταξύ 0 και 1 (εικόνα 39),



Εικόνα 39: Azure ML Studio: Κανονικοποιημένα αποτελέσματα βαθμολόγησης OC-SVM

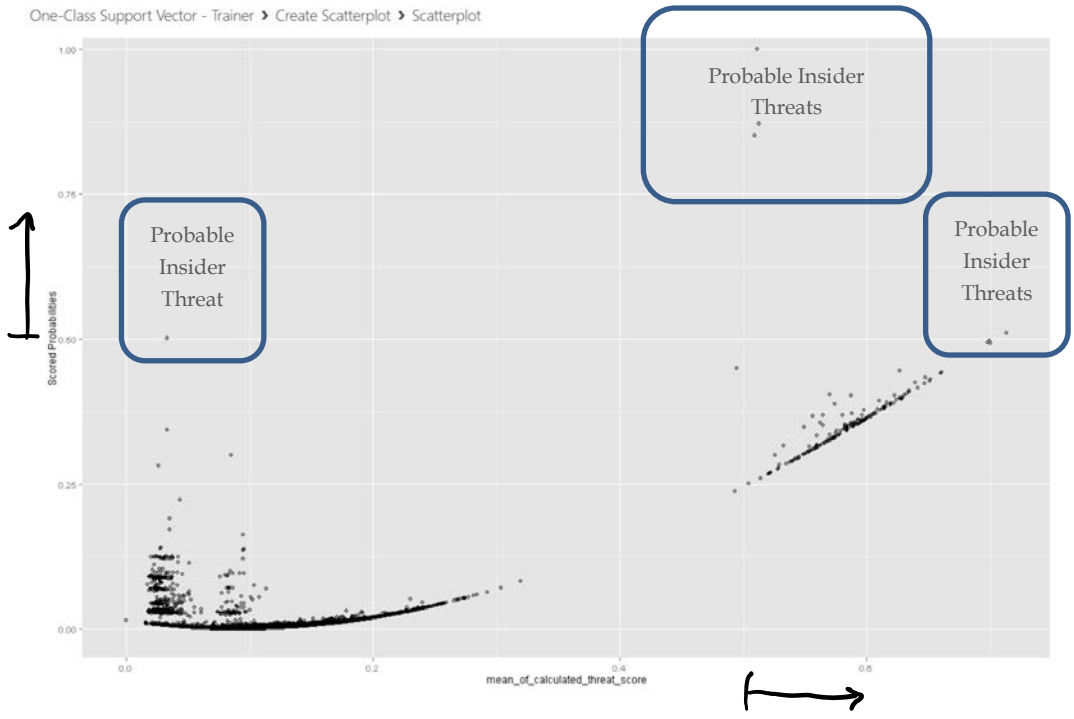
Η στήλη Scored labels μας δείχνει σε ποια κλάση ανήκει το συγκεκριμένο event και το Scored Probabilities, ποια πιθανότητα να είναι της κλάσης που αναφέρεται. Η κλάση 0 είναι τα normal δεδομένα και 1 είναι τα χαρακτηρισμένα ως anomaly,

9. Το Create Scatterplot module, παράγει το παρακάτω γράφημα (εικόνα 40):



Εικόνα 40: Γράφημα αποτελεσμάτων πειράματος βαθμολόγησης με OC-SVM για την πολιτική A

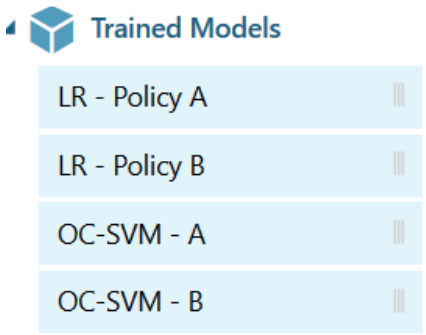
όπου οι δύο κλάσεις κατηγοριοποίησης, με threshold διαχωρισμού να είναι το 0.5. Το αντίστοιχο γράφημα για την Πολιτική B είναι (εικόνα 41):



Εικόνα 41: Γράφημα αποτελεσμάτων πειράματος βαθμολόγησης με OC-SVM για την πολιτική B

όπου και εδώ είναι ξεκάθαρες οι δύο κλάσεις κατηγοριοποίησης, με threshold το 0.5.

Στη συνέχεια, αποθηκεύουμε τα εκπαιδευμένα μοντέλα στη βιβλιοθήκη μας:



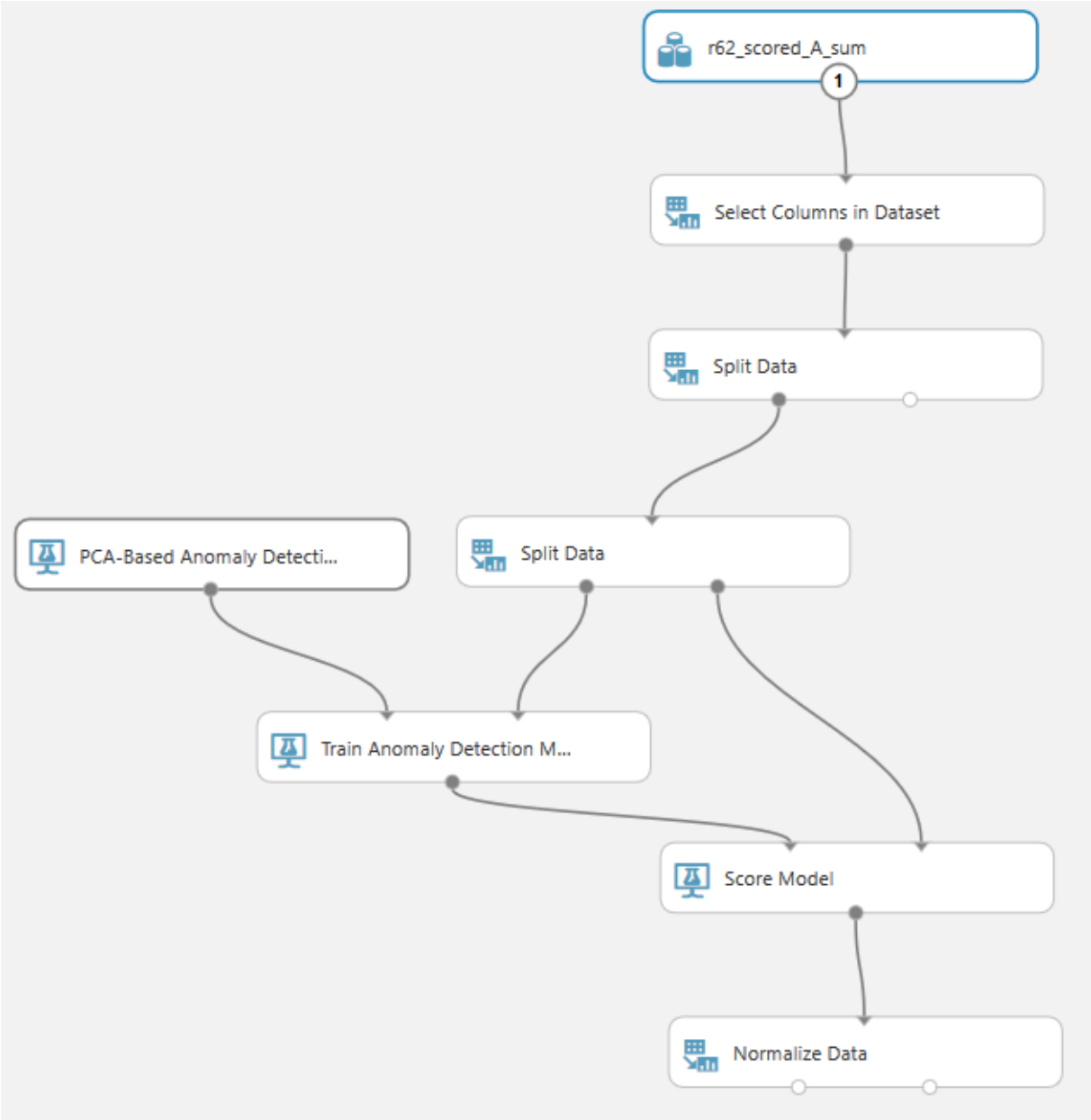
4.6.4 PCA-Based Anomaly Detection

Ο PCA-Based Anomaly Detection ανήκει και αυτός στην κατηγορία των Anomaly Detection αλγορίθμων, και εδώ θα δώσουμε τα δεδομένα χωρίς να τα χωρίσουμε πρωτύτερα σε κλάσεις [19].

Τα modules που θα χρειαστούμε είναι:

1. r62_dataset_A_sum,
2. Select Columns in Dataset,
3. Split Data,
4. Split Data,
5. PCA-Based Anomaly Detection,
6. Train Anomaly Detection,
7. Score Model,
8. Normalize Data

Το σχεδιάγραμμα θα είναι όπως παρακάτω (εικόνα 42):



Εικόνα 42: Azure ML Studio: Ο PCA στο πείραμα βαθμολόγησης

Τα βήματα, επεξηγούνται παρακάτω:

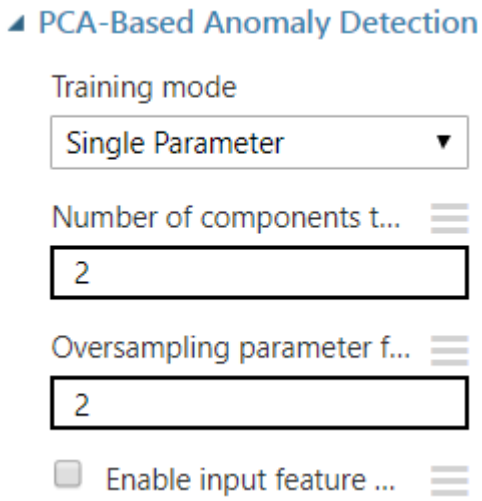
1. Εισαγωγή του r62_dataset_A_sum,

2. Με το Select Columns in Dataset, επιλέγουμε τα πεδία τα οποία θα είναι και τα features του αλγόριθμου

3. Με το Split Data, κάνουμε το διαχωρισμό του Dataset, όπου το 75% θα διατεθεί για εκπαίδευση και το 25% για test data.

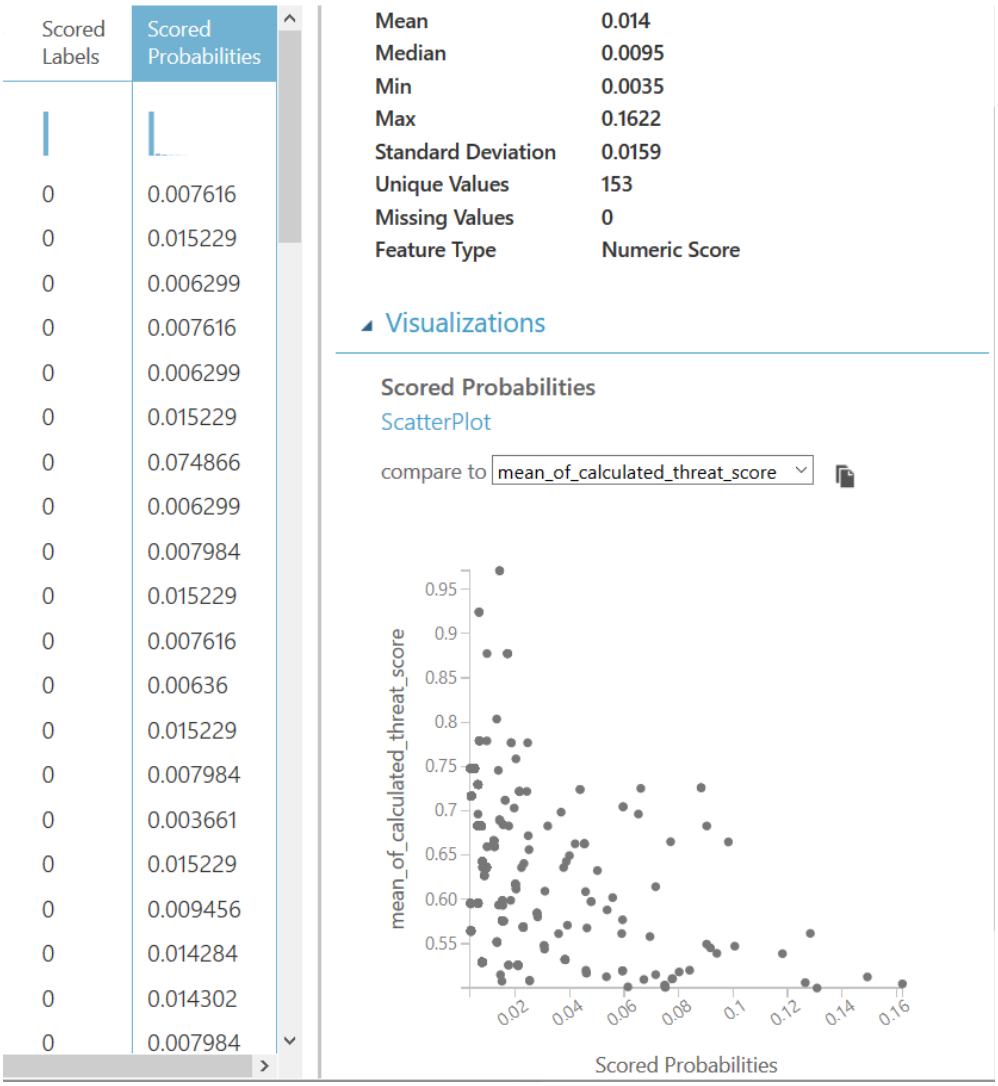
4. Με το 2ο Split Data, εφαρμόζουμε ένα relative expression ώστε να διαχωρίσουμε τα normal data από τα υπόλοιπα. Στην ουσία ορίζουμε στον αλγόριθμο την κλάση με τα normal data. Εδώ επιλέγουμε το mean_of_calculated_threat_score < 0.5,

5. Το PCA-Based Anomaly Detection module, περιέχει τις παραμέτρους που μπορούμε να μεταβάλλουμε ώστε να πλησιάσουμε όσο πιο κοντά γίνεται, και κατά προσέγγιση, σε ένα ολοκληρωμένο μοντέλο. Η παράμετρος Number of Components αφορά στο πλήθος των διαστάσεων στο οποίο θέλουμε να καταλήξουμε, και πρέπει να είναι μικρότερος από τον αριθμό των features. Στην περίπτωση μας, πρέπει να είναι μικρότερο του 4. Εμείς δοκιμάσαμε με 3, 2, 1 και καταλήξαμε ότι το 2 είναι ιδανικό:



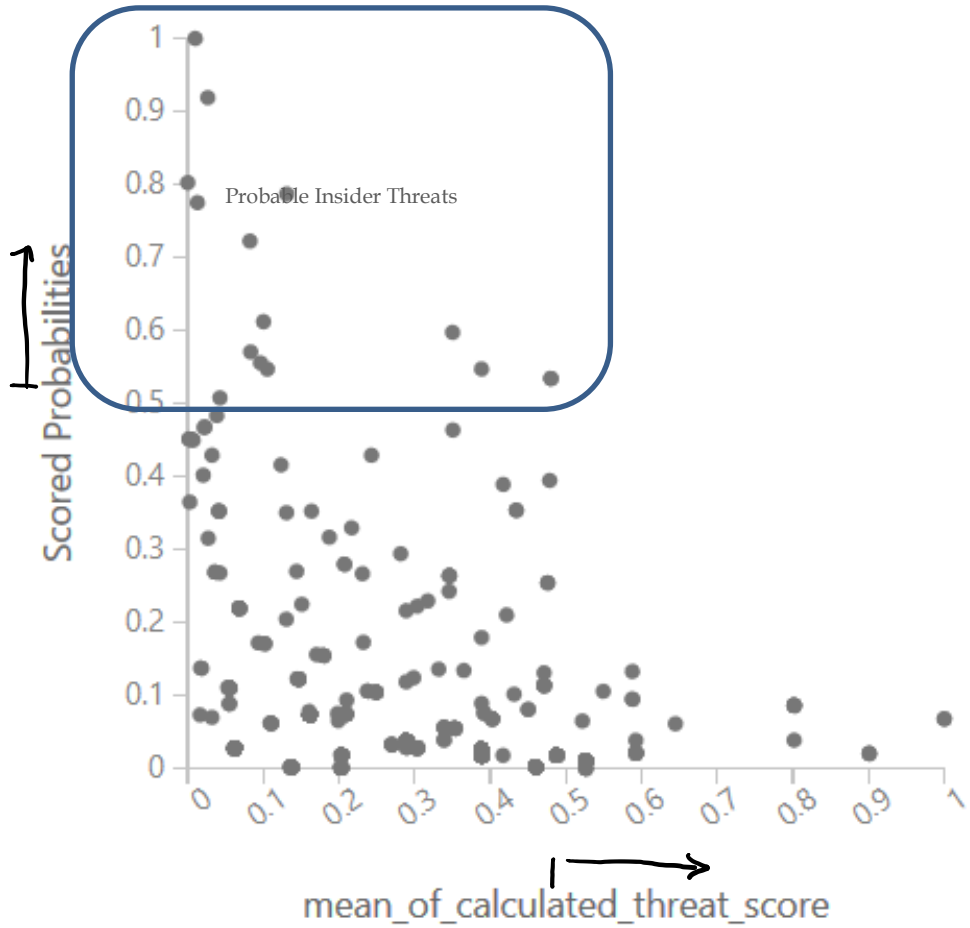
6. Το Train Anomaly Detection δεν δέχεται καμία παράμετρο, απλά εκπαιδεύει το μοντέλο,

7. Με το Score Model, βλέπουμε τα αποτελέσματα (εικόνα 43):



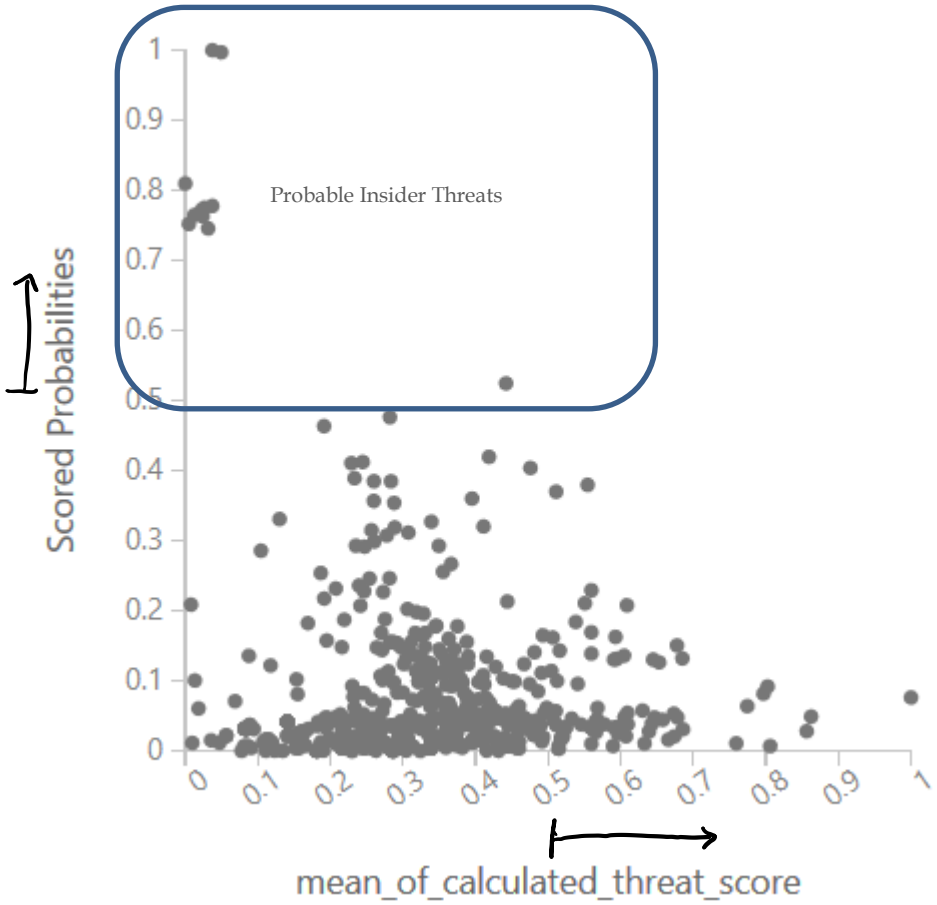
Εικόνα 43: Azure ML Studio: Αποτελέσματα βαθμολόγησης PCA

8. Με το Normalize Data, αλλάζουμε το εύρος των αποτελεσμάτων μεταξύ 0 και 1. Το γράφημα διασποράς μας βοηθάει ώστε να καταλάβουμε τις πιθανές εσωτερικές απειλές (εικόνα 44):



Εικόνα 44: Γράφημα αποτελεσμάτων πειράματος βαθμολόγησης με PCA για την πολιτική Α

Το αντίστοιχο γράφημα για την Πολιτική Β είναι (εικόνα 45):



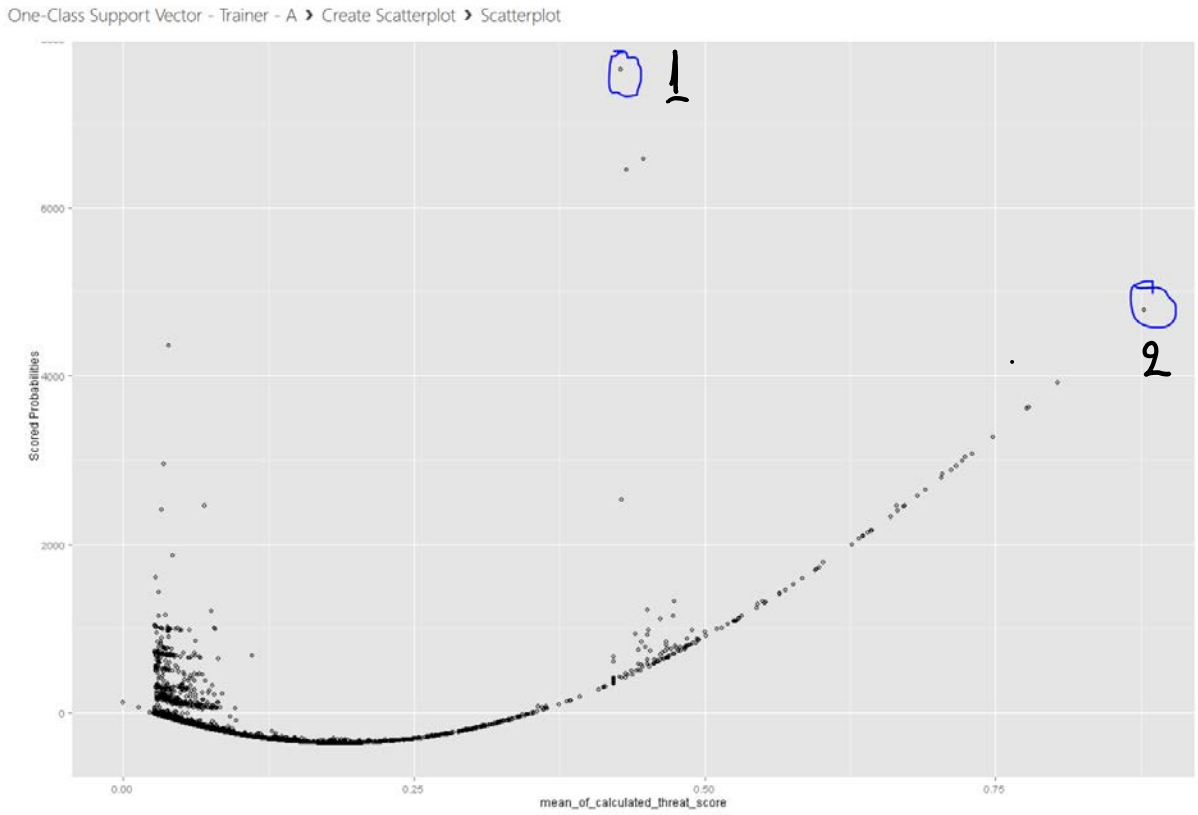
Εικόνα 45: Γράφημα αποτελεσμάτων πειράματος βαθμολόγησης με PCA για την πολιτική B

Σε αυτό είναι αρκετά ξεκάθαρο, τι έχει βαθμολογηθεί σαν πιθανή εσωτερική απειλή.

4.6.5 Επιβεβαίωση Αποτελεσμάτων

Στο σημείο αυτό, αφού το σύστημα μας παρείχε τις σχετικές πληροφορίες, το τμήμα IT-Security θα πρέπει να επιβεβαιώσει την ορθότητα αυτών των στοιχείων. Θα πάρουμε δειγματοληπτικά 2 σημεία από κάθε γράφημα των SVM και PCA, και για τις δύο πολιτικές και θα προσπαθήσουμε να βρούμε την αντίστοιχη εγγραφή στη βάση μας ώστε να εκτιμήσουμε την ορθότητα της κατάταξης της ενέργειας (η αναγνώριση θα γίνει χειροκίνητα). Στη συνέχεια θα παραθέσουμε τα διαγράμματα διασποράς (στην μη-κανονικοποιημένη τους μορφή, ώστε να γίνει ευκολότερα η αναζήτηση), από τα οποία θα επιλέξουμε κάποια σημεία:

SVM – Policy A



Εικόνα 46: Γράφημα αποτελεσμάτων πειράματος βαθμολόγησης με SVM για την πολιτική A

Σημείο 1: Το σημείο έχει τεταγμένη περίπου 0.42 και τετμημένη λίγο μικρότερη από το 8000. Αναζητώντας στη βάση μας, βλέπουμε ότι είναι το (0.42750096, 7635.80127),

Results | Messages

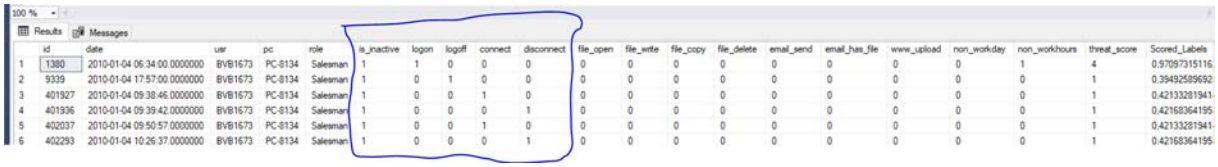
	event_day	event_month	event_year	usr	NumEvents	total_policy_threat_score_per_day	total_calculated_score_per_day	mean_of_calculated_threat_score
1	4	January	2010	BVB1673	208	222	88,9201996660705	0,427500959933031
2	4	January	2010	QWL3474	208	4	9,76521765738984	0,0469481618143742
3	4	January	2010	HGD1664	208	2	8,083948460035	0,0388651368270913
4	4	January	2010	HHD2393	208	2	7,86147616908682	0,0377955585052251

και περιγράφονται γεγονότα του χρήστη BVB1673 στις 4 Ιανουαρίου 2010. Οι ενέργειες του εκείνη την ημέρα είναι:

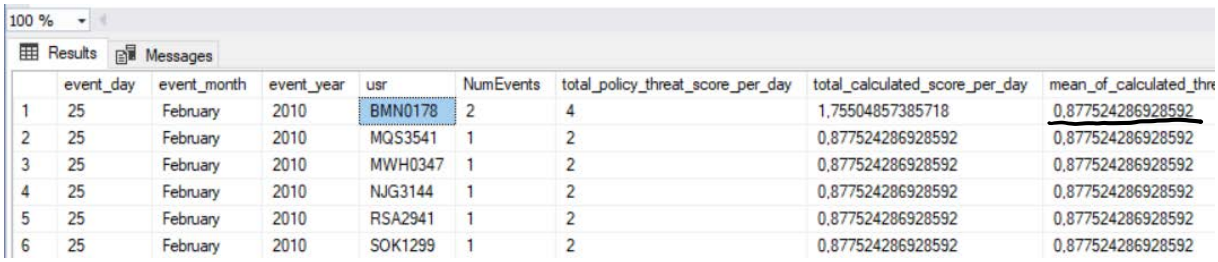
Results | Messages

id	date	usr	pc	role	is_inactive	logon	logoff	connect	disconnect	file_open	file_write	file_copy	file_delete	email_send	email_has_file	www_upload	non_workday	non_workhours	threat_score	Scored_Labels
1380	2010-01-04 06:34:00.0000000	BVB1673	PC-8134	Salesman	1	1	0	0	0	0	0	0	0	0	0	0	0	1	4	0.97097315116
9339	2010-01-04 17:57:00.0000000	BVB1673	PC-8134	Salesman	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.39492589692
401927	2010-01-04 09:38:46.0000000	BVB1673	PC-8134	Salesman	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.42133281947
401936	2010-01-04 09:39:42.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0.42168364195

Ο χρήστης BVB1673 είναι ο Brody Vernon Bonner ο οποίος είναι πωλητής και στις 4 Ιανουαρίου 2010 έπρεπε να είναι ανενεργός:



Σημείο 2: Το σημείο έχει τεταγμένη περίπου 0.8 και τετμημένη λίγο μεγαλύτερη από το 4000. Αναζητώντας στη βάση μας, βλέπουμε ότι είναι το (0.877524287, 477.675781),

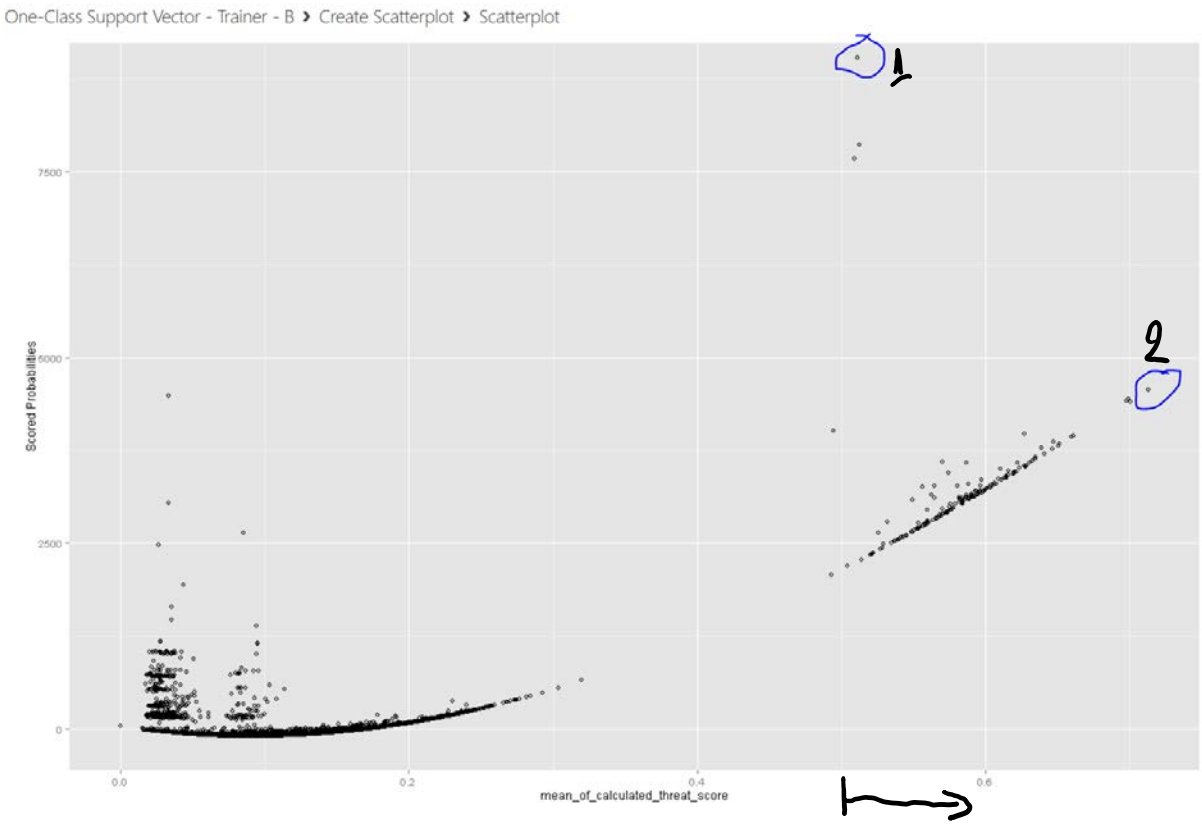


	event_day	event_month	event_year	usr	NumEvents	total_policy_threat_score_per_day	total_calculated_score_per_day	mean_of_calculated_thre
1	25	February	2010	BMN0178	2	4	1,75504857385718	0,877524286928592
2	25	February	2010	MQS3541	1	2	0,877524286928592	0,877524286928592
3	25	February	2010	MWH0347	1	2	0,877524286928592	0,877524286928592
4	25	February	2010	NJG3144	1	2	0,877524286928592	0,877524286928592
5	25	February	2010	RSA2941	1	2	0,877524286928592	0,877524286928592
6	25	February	2010	SOK1299	1	2	0,877524286928592	0,877524286928592

και περιγράφονται γεγονότα του χρήστη BMN0178 στις 25 Φεβρουαρίου 2010. Οι ενέργειες του εκείνη την ημέρα είναι:

Ο χρήστης BMN0178 είναι η Bo Mechelle Nelson η οποία είναι Technical Writer και στις 25 Φεβρουαρίου 2010 έπρεπε να είναι ανενεργή:

SVM – Policy B



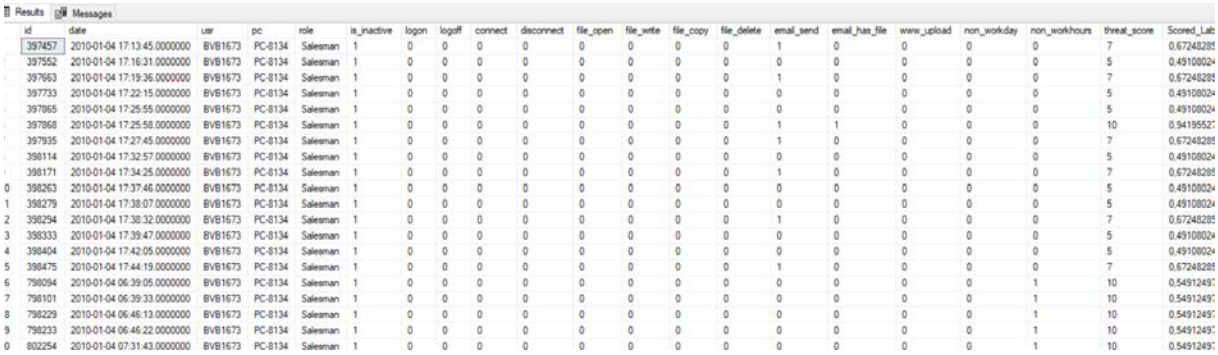
Εικόνα 47: Γράφημα αποτελεσμάτων πειράματος βαθμολόγησης με SVM για την πολιτική B

Σημείο 1: Το σημείο έχει τεταγμένη περίπου 0.42 και τετμημένη λίγο μικρότερη από το 8000. Αναζητώντας στη βάση μας, βλέπουμε ότι είναι το (0.42750096, 7635.80127),

Results | Messages

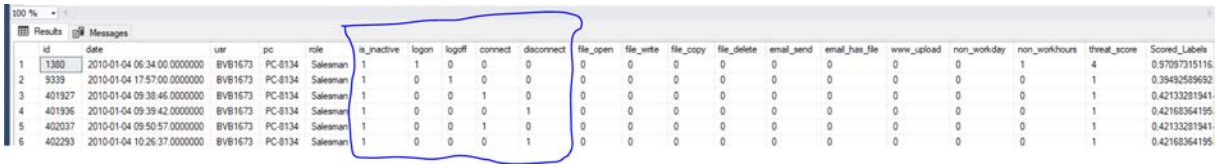
	event_day	event_month	event_year	usr	NumEvents	total_policy_threat_score_per_day	total_calculated_score_per_day	mean_of_calculated_thre
1	4	January	2010	BVB1673	208	1139	106,34505862953	0,511274320334281
2	4	January	2010	HGD1664	208	40	7,36311426135488	0,0353995877949754
3	4	January	2010	HHD2393	208	29	6,74935086133479	0,0324488022179557
4	4	January	2010	OWL3474	208	20	6,57664014666266	0,0316184622435705

και περιγράφονται γεγονότα του χρήστη BVB1673 στις 4 Ιανουαρίου 2010. Οι ενέργειες του εκείνη την ημέρα είναι:



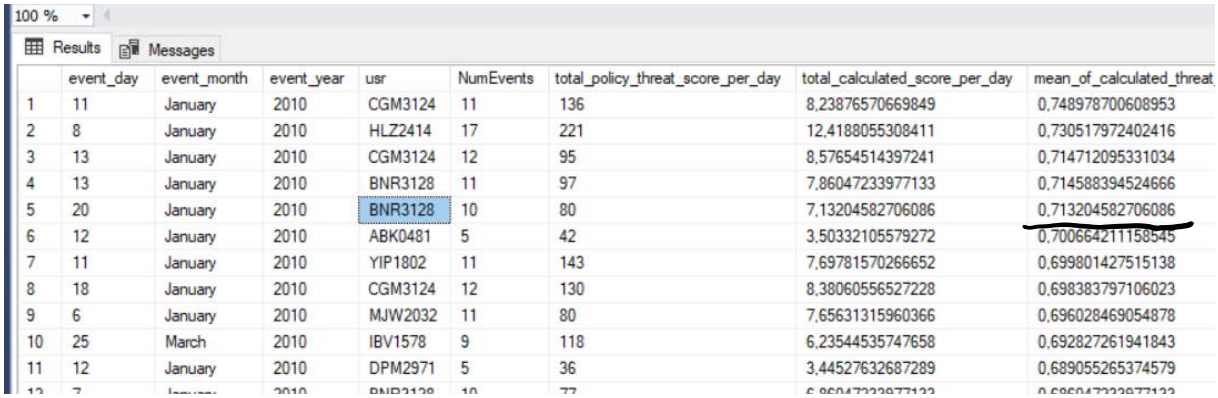
	id	date	usr	pc	role	is_inactive	logon	logoff	connect	disconnect	file_open	file_write	file_copy	file_delete	email_send	email_has_file	www_upload	non_workday	non_workhours	threat_score	Scored_Lab
	397457	2010-01-04 17:13:45.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	1	0	0	0	0	7	0.67248285
	397552	2010-01-04 17:16:31.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0.49108024
	397663	2010-01-04 17:19:36.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	1	0	0	0	0	7	0.67248285
	397733	2010-01-04 17:22:15.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0.49108024
	397865	2010-01-04 17:25:55.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0.49108024
	397868	2010-01-04 17:25:58.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	1	1	0	0	0	10	0.94195527
	397935	2010-01-04 17:27:45.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	1	0	0	0	0	7	0.67248285
	398114	2010-01-04 17:32:57.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0.49108024
	398171	2010-01-04 17:34:25.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	1	0	0	0	0	7	0.67248285
0	398263	2010-01-04 17:37:46.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0.49108024
1	398279	2010-01-04 17:38:07.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0.49108024
2	398294	2010-01-04 17:38:32.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	1	0	0	0	0	7	0.67248285
3	398333	2010-01-04 17:39:47.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0.49108024
4	398404	2010-01-04 17:42:05.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0.49108024
5	398475	2010-01-04 17:44:19.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	1	0	0	0	0	7	0.67248285
6	798094	2010-01-04 06:39:05.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	0	0	0	0	1	10	0.54912497
7	798101	2010-01-04 06:39:33.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	0	0	0	0	1	10	0.54912497
8	798229	2010-01-04 06:46:13.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	0	0	0	0	1	10	0.54912497
9	798233	2010-01-04 06:46:22.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	0	0	0	0	1	10	0.54912497
0	802254	2010-01-04 07:31:43.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	0	0	0	0	0	0	0	0	0	1	10	0.54912497

Ο χρήστης BVB1673 είναι ο Brody Vernon Bonner ο οποίος είναι πωλητής και στις 4 Ιανουαρίου 2010 έπρεπε να είναι ανενεργός:



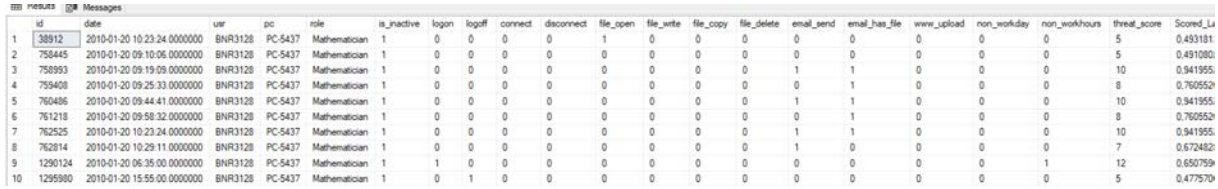
	id	date	usr	pc	role	is_inactive	logon	logoff	connect	disconnect	file_open	file_write	file_copy	file_delete	email_send	email_has_file	www_upload	non_workday	non_workhours	threat_score	Scored_Labels
1	1380	2010-01-04 06:34:00.0000000	BVB1673	PC-8134	Salesman	1	1	0	0	0	0	0	0	0	0	0	0	1	4	0.97097315116
2	9339	2010-01-04 17:57:00.0000000	BVB1673	PC-8134	Salesman	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0.39492589692
3	401927	2010-01-04 09:38:46.0000000	BVB1673	PC-8134	Salesman	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0.42133281941
4	401936	2010-01-04 09:39:42.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0.42168364195
5	402037	2010-01-04 09:50:57.0000000	BVB1673	PC-8134	Salesman	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0.42133281941
6	402293	2010-01-04 10:26:37.0000000	BVB1673	PC-8134	Salesman	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0.42168364195

<u>Σημείο 2</u>: Το σημείο έχει τεταγμένη περίπου 0.7 και τετμημένη λίγο μικρότερη από το 5000. Αναζητώντας στη βάση μας, βλέπουμε ότι είναι το (0.713204, 4566.443359),



	event_day	event_month	event_year	usr	NumEvents	total_policy_threat_score_per_day	total_calculated_score_per_day	mean_of_calculated_threat
1	11	January	2010	CGM3124	11	136	8,23876570669849	0,748978700608953
2	8	January	2010	HLZ2414	17	221	12,4188055308411	0,730517972402416
3	13	January	2010	CGM3124	12	95	8,57654514397241	0,714712095331034
4	13	January	2010	BNR3128	11	97	7,86047233977133	0,714588394524666
5	20	January	2010	BNR3128	10	80	7,13204582706086	0,713204582706086
6	12	January	2010	ABK0481	5	42	3,50332105579272	0,700664211158545
7	11	January	2010	YIP1802	11	143	7,69781570266652	0,699801427515138
8	18	January	2010	CGM3124	12	130	8,38060556527228	0,698383797106023
9	6	January	2010	MJW2032	11	80	7,65631315960366	0,696028469054878
10	25	March	2010	IBV1578	9	118	6,23544535747658	0,692827261941843
11	12	January	2010	DPM2971	5	36	3,44527632687289	0,689055265374579

και περιγράφονται γεγονότα του χρήστη BNR3128 στις 20 Ιανουαρίου 2010. Οι ενέργειες του εκείνη την ημέρα είναι:

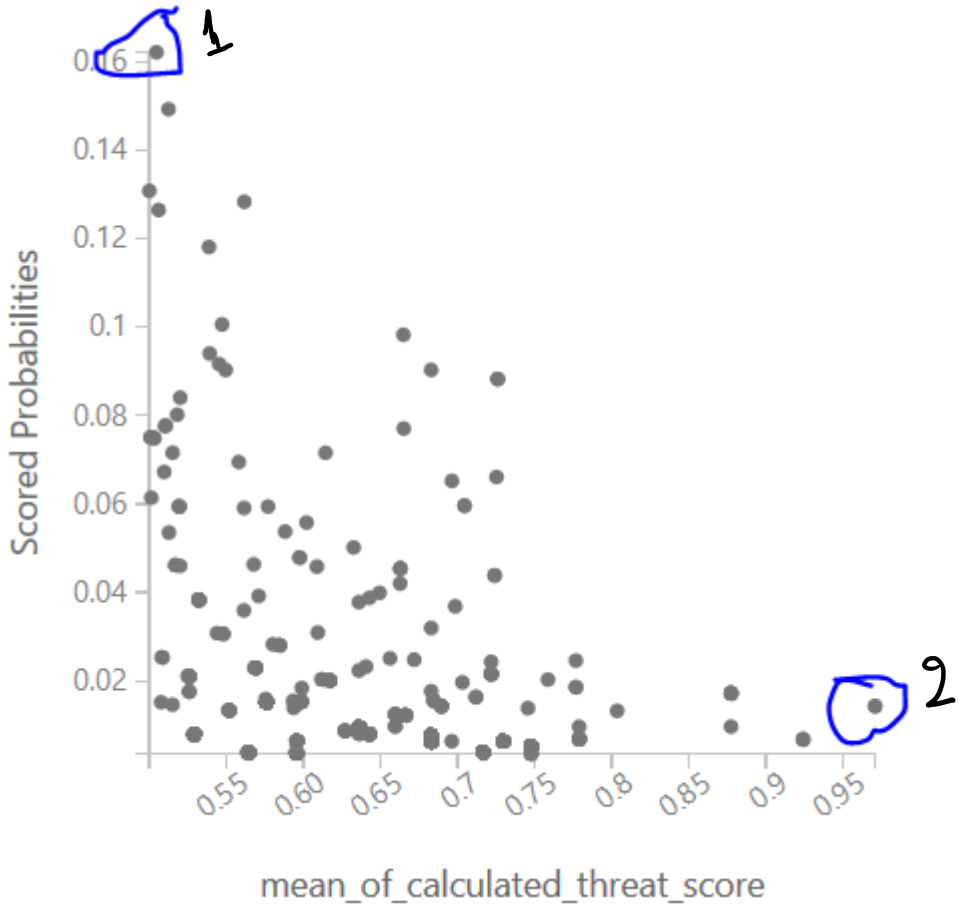


	id	date	usr	pc	role	is_inactive	logon	logoff	connect	disconnect	file_open	file_write	file_copy	file_delete	email_send	email_has_file	www_upload	non_workday	non_workhours	threat_score	Scored_La
1	38912	2010-01-20 10:23:24.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	1	0	0	0	0	0	0	0	5	0.49318
2	758445	2010-01-20 09:10:06.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	0	0	0	0	0	0	0	0	5	0.49108
3	758993	2010-01-20 09:19:09.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	0	0	0	0	1	1	0	0	10	0.94195
4	759408	2010-01-20 09:25:33.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	0	0	0	0	0	1	0	0	8	0.76055
5	760486	2010-01-20 09:44:41.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	0	0	0	0	1	1	0	0	10	0.94195
6	761218	2010-01-20 09:58:32.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	0	0	0	0	0	1	0	0	8	0.76055
7	762525	2010-01-20 10:23:24.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	0	0	0	0	1	1	0	0	10	0.94195
8	762814	2010-01-20 10:29:11.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	0	0	0	0	1	0	0	0	7	0.67248
9	1290124	2010-01-20 06:35:00.0000000	BNR3128	PC-5437	Mathematician	1	1	0	0	0	0	0	0	0	0	0	0	1	12	0.65075
10	1295980	2010-01-20 15:55:00.0000000	BNR3128	PC-5437	Mathematician	1	0	1	0	0	0	0	0	0	0	0	0	0	5	0.47757

Ο χρήστης BNR3128 είναι ο Benjamin Nicolas Ryan ο οποίος είναι Μαθηματικός και στις 20 Ιανουαρίου 2010 έπρεπε να είναι ανενεργός, ενώ έστειλε email με αρχείο επισυναπτόμενο:

	id	date	usr	pc	role	is_inactive	logon	logoff	connect	disconnect	file_open	file_write	file_copy	file_delete	email_send	email_has_file	www_upload	non_workday	non_workhours	threat_score	Scored_Lal
1	38912	2010-01-20 10:23:24.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	1	0	0	0	0	0	0	0	0	5	0,4931813
2	758445	2010-01-20 09:10:06.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0,4910802
3	758993	2010-01-20 09:19:09.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	0	0	0	0	1	1	0	0	0	10	0,9419552
4	759408	2010-01-20 09:25:33.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	0	0	0	0	0	1	0	0	0	8	0,7605526
5	760486	2010-01-20 09:44:41.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	0	0	0	0	1	1	0	0	0	10	0,9419552
6	761218	2010-01-20 09:58:32.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	0	0	0	0	0	1	0	0	0	8	0,7605526
7	762525	2010-01-20 10:23:24.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	0	0	0	0	1	1	0	0	0	10	0,9419552
8	762814	2010-01-20 10:29:11.0000000	BNR3128	PC-5437	Mathematician	1	0	0	0	0	0	0	0	0	1	0	0	0	0	7	0,6724828
9	1290124	2010-01-20 06:35:00.0000000	BNR3128	PC-5437	Mathematician	1	1	0	0	0	0	0	0	0	0	0	0	0	1	12	0,6507596
10	1295980	2010-01-20 15:55:00.0000000	BNR3128	PC-5437	Mathematician	1	0	1	0	0	0	0	0	0	0	0	0	0	0	5	0,4775706

PCA – Policy A



Εικόνα 48: Γράφημα αποτελεσμάτων πειράματος βαθμολόγησης με PCA για την πολιτική A

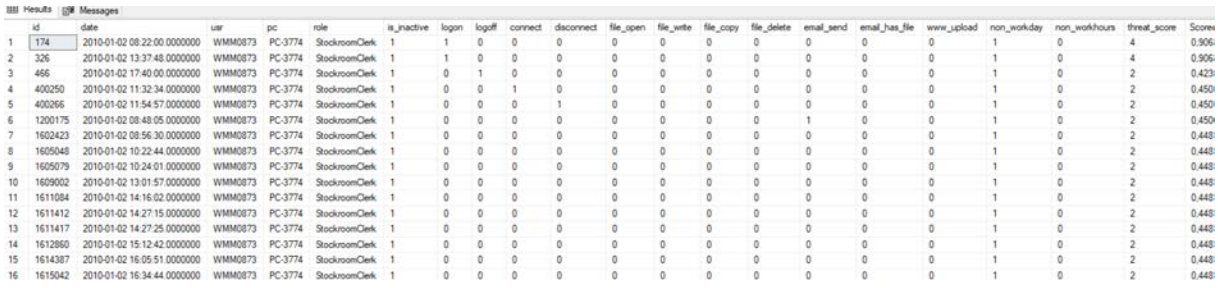
Σημείο 1: Το σημείο έχει τεταγμένη περίπου 0.5 και τετμημένη λίγο μεγαλύτερη από το 0.16. Αναζητώντας στη βάση μας, βλέπουμε ότι είναι το (0.50475 , 0.16216),

100 %

Results | Messages

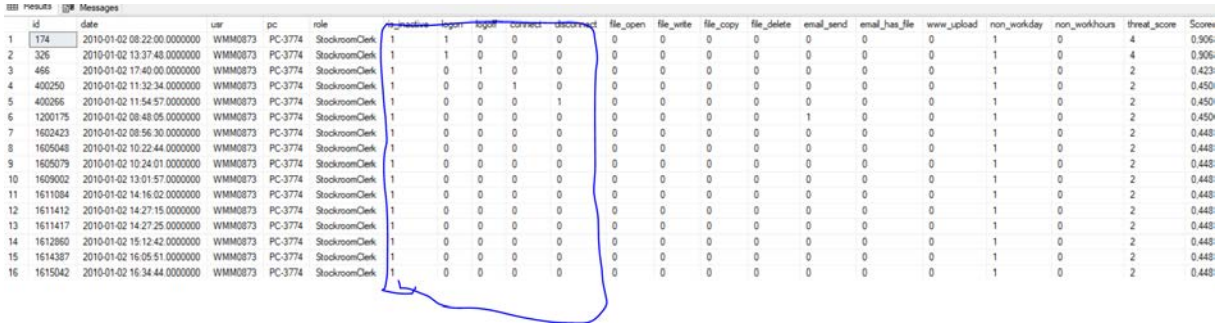
	event_day	event_month	event_year	usr	NumEvents	total_policy_threat_score_per_day	total_calculated_score_per_day	mean_of_calculated_threat_scor
1	2	January	2010	WMM0873	16	36	8,07600108203503	0,504750067627189

και περιγράφονται γεγονότα του χρήστη WMM0873 στις 2 Ιανουαρίου 2010. Οι ενέργειες του εκείνη την ημέρα είναι:



	id	date	usr	pc	role	is_inactive	logon	logoff	connect	disconnect	file_open	file_write	file_copy	file_delete	email_send	email_has_file	www_upload	non_workday	non_workhours	threat_score	Score
1	174	2010-01-02 08:22:00.0000000	WMM0873	PC-3774	StockroomClerk	1	1	0	0	0	0	0	0	0	0	0	0	1	0	4	0.906
2	326	2010-01-02 13:37:48.0000000	WMM0873	PC-3774	StockroomClerk	1	1	0	0	0	0	0	0	0	0	0	0	1	0	4	0.906
3	466	2010-01-02 17:40:00.0000000	WMM0873	PC-3774	StockroomClerk	1	0	1	0	0	0	0	0	0	0	0	0	1	0	2	0.423
4	400250	2010-01-02 11:32:34.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	1	0	0	0	0	0	0	0	0	1	0	2	0.450
5	400266	2010-01-02 11:54:57.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	1	0	0	0	0	0	0	0	1	0	2	0.450
6	1200175	2010-01-02 08:48:05.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	1	0	0	1	0	2	0.450
7	1602423	2010-01-02 08:56:30.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
8	1605048	2010-01-02 10:22:44.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
9	1605079	2010-01-02 10:24:01.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
10	1609002	2010-01-02 13:01:57.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
11	1611084	2010-01-02 14:16:02.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
12	1611412	2010-01-02 14:27:15.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
13	1611417	2010-01-02 14:27:25.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
14	1612860	2010-01-02 15:12:42.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
15	1614387	2010-01-02 16:05:51.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
16	1615042	2010-01-02 16:34:44.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448

Ο χρήστης WMM0873 είναι ο Whoopi Maite Maxwel ο οποίος είναι αποθηκάριος και στις 2 Ιανουαρίου 2010 έπρεπε να είναι ανενεργός:



	id	date	usr	pc	role	is_inactive	logon	logoff	connect	disconnect	file_open	file_write	file_copy	file_delete	email_send	email_has_file	www_upload	non_workday	non_workhours	threat_score	Score
1	174	2010-01-02 08:22:00.0000000	WMM0873	PC-3774	StockroomClerk	1	1	0	0	0	0	0	0	0	0	0	0	1	0	4	0.906
2	326	2010-01-02 13:37:48.0000000	WMM0873	PC-3774	StockroomClerk	1	1	0	0	0	0	0	0	0	0	0	0	1	0	4	0.906
3	466	2010-01-02 17:40:00.0000000	WMM0873	PC-3774	StockroomClerk	1	0	1	0	0	0	0	0	0	0	0	0	1	0	2	0.423
4	400250	2010-01-02 11:32:34.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	1	0	0	0	0	0	0	0	0	1	0	2	0.450
5	400266	2010-01-02 11:54:57.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	1	0	0	0	0	0	0	0	1	0	2	0.450
6	1200175	2010-01-02 08:48:05.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	1	0	0	1	0	2	0.450
7	1602423	2010-01-02 08:56:30.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
8	1605048	2010-01-02 10:22:44.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
9	1605079	2010-01-02 10:24:01.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
10	1609002	2010-01-02 13:01:57.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
11	1611084	2010-01-02 14:16:02.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
12	1611412	2010-01-02 14:27:15.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
13	1611417	2010-01-02 14:27:25.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
14	1612860	2010-01-02 15:12:42.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
15	1614387	2010-01-02 16:05:51.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448
16	1615042	2010-01-02 16:34:44.0000000	WMM0873	PC-3774	StockroomClerk	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0.448

Σημείο 2: Το σημείο έχει τεταγμένη περίπου 0.95 και τετμημένη λίγο μεγαλύτερη από το 0.01. Αναζητώντας στη βάση μας, βλέπουμε ότι είναι το (0.9709731, 0.014248),

	event_day	event_month	event_year	usr	NumEvents	total_policy_threat_score_per_day	total_calculated_score_per_day	mean_of_calculated_thre
1	25	February	2010	RNB2679	1	4	0,970973151162781	0,970973151162781
2	25	February	2010	WPJ1400	1	4	0,970973151162781	0,970973151162781

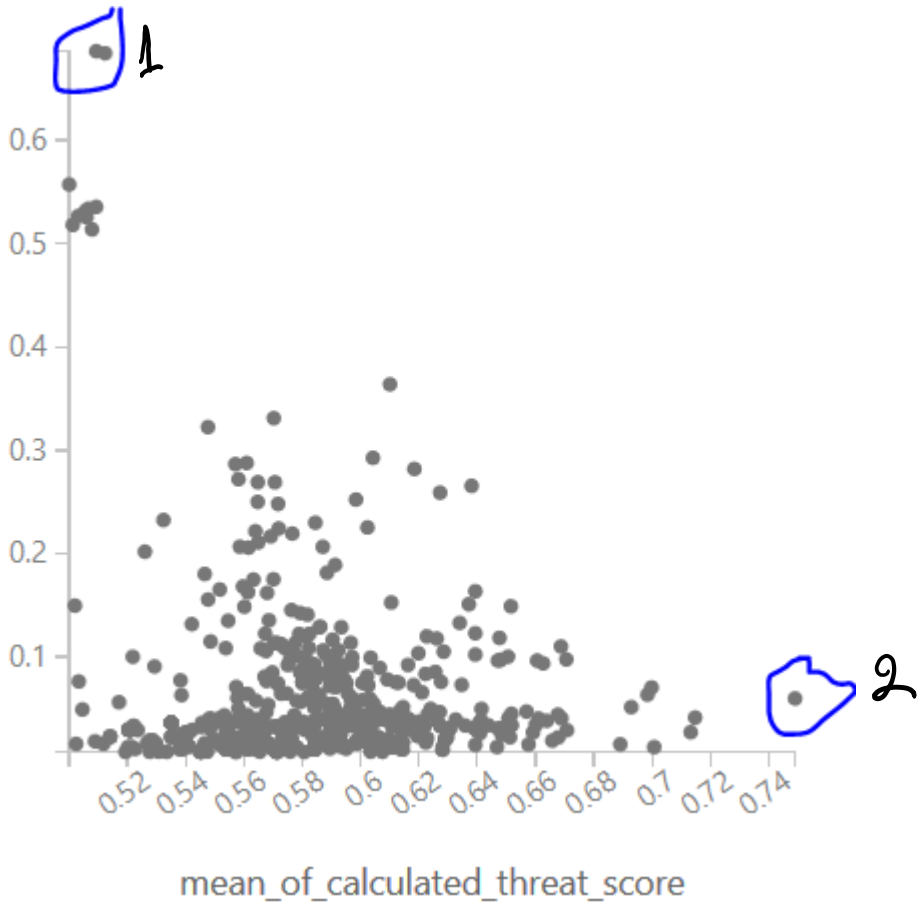
και περιγράφονται γεγονότα του χρήστη RNB2679 στις 25 Φεβρουαρίου 2010. Οι ενέργειες του εκείνη την ημέρα είναι:

	id	date	usr	pc	role	is_inactive	logon	logoff	connect	disconnect	file_open	file_write	file_copy	file_delete	email_send	email_has_file	www_upload	non_workday	non_workhours	threat
1	393235	2010-02-25 07:03:00.0000000	RNB2679	PC-3486	ComputerProgrammer	1	1	0	0	0	0	0	0	0	0	0	0	0	1	4

Ο χρήστης BNR3128 είναι ο Reese Neil Blackburn ο οποίος είναι Computer Programmer και στις 25 Φεβρουαρίου 2010 θα έπρεπε να είναι ανενεργός:

	id	date	usr	pc	role	is_inactive	logon	logoff	connect	disconnect	file_open	file_write	file_copy	file_delete	email_send	email_has_file	www_upload	non_workday	non_workhours	threat
1	393235	2010-02-25 07:03:00.0000000	RNB2679	PC-3486	ComputerProgrammer	1	1	0	0	0	0	0	0	0	0	0	0	0	1	4

PCA – Policy B



Εικόνα 49: Γράφημα αποτελεσμάτων πειράματος βαθμολόγησης με PCA για την πολιτική B

Σημείο 1: Το σημείο έχει τεταγμένη περίπου 0.5 και τετμημένη λίγο μεγαλύτερη από το 0.6. Αναζητώντας στη βάση μας, βλέπουμε ότι είναι το (0.50938 , 0.68631),

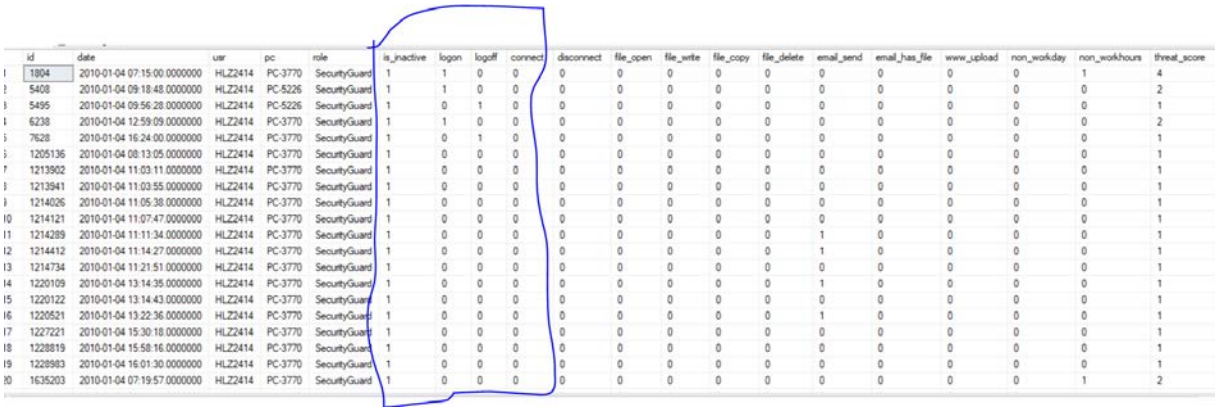
Results | Messages

	event_day	event_month	event_year	usr	NumEvents	total_policy_threat_score_per_day	total_calculated_score_per_day	mean_of_calculated_threat_
1	4	January	2010	HLZ2414	169	1037	86,085667354065	0,509382647065474
2	2	January	2010	NDK3436	169	32	13,0638862055039	0,0773011018077153
3	4	January	2010	DKD2409	169	47	7,57491131212203	0,0448219604267576
4	4	January	2010	JSN0214	169	51	7,09780520487732	0,041998847366138
5	4	January	2010	CSM3502	169	26	5,88411360248592	0,034817240251396
6	4	January	2010	CIY1654	169	14	4,62055660992312	0,0273405716563498

και περιγράφονται γεγονότα του χρήστη HLZ2414 στις 4 Ιανουαρίου 2010. Οι ενέργειες του εκείνη την ημέρα είναι:

	id	date	usr	pc	role	is_inactive	logon	logoff	connect	disconnect	file_open	file_write	file_copy	file_delete	email_send	email_has_file	www_upload	non_workday	non_workhours	threat_score
1	1804	2010-01-04 07:15:00.0000000	HLZ2414	PC-3770	SecurityGuard	1	1	0	0	0	0	0	0	0	0	0	0	0	1	4
2	5408	2010-01-04 09:18:48.0000000	HLZ2414	PC-5226	SecurityGuard	1	1	0	0	0	0	0	0	0	0	0	0	0	0	2
3	5495	2010-01-04 09:56:28.0000000	HLZ2414	PC-5226	SecurityGuard	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1
4	6238	2010-01-04 12:59:09.0000000	HLZ2414	PC-3770	SecurityGuard	1	1	0	0	0	0	0	0	0	0	0	0	0	0	2
5	7628	2010-01-04 16:24:00.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1
6	1205136	2010-01-04 08:13:05.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
7	1213902	2010-01-04 11:03:11.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	1213941	2010-01-04 11:03:55.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
9	1214026	2010-01-04 11:05:38.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
10	1214121	2010-01-04 11:07:47.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
11	1214289	2010-01-04 11:11:34.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
12	1214412	2010-01-04 11:14:27.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
13	1214734	2010-01-04 11:21:51.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
14	1220109	2010-01-04 13:14:35.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
15	1220122	2010-01-04 13:14:43.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
16	1220521	2010-01-04 13:22:36.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
17	1227221	2010-01-04 15:30:18.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
18	1228819	2010-01-04 15:58:16.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
19	1228983	2010-01-04 16:01:30.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
20	1635203	2010-01-04 07:19:57.0000000	HLZ2414	PC-3770	SecurityGuard	1	0	0	0	0	0	0	0	0	0	0	0	0	1	2

Ο χρήστης HLZ2414 είναι ο Hall Leonard Zimmerman ο οποίος είναι φύλακας και στις 4 Ιανουαρίου 2010 έπρεπε να είναι ανενεργός:



<u>Σημείο 2</u>: Το σημείο έχει τεταγμένη περίπου 0.74 και τετμημένη λίγο μικρότερη από το 0,1. Αναζητώντας στη βάση μας, βλέπουμε ότι είναι το (0.74897, 0.059830),

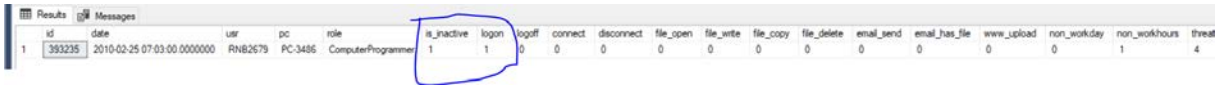
Results Messages

	event_day	event_month	event_year	usr	NumEvents	total_policy_threat_score_per_day	total_calculated_score_per_day	mean_of_calculated_threat_s
1	11	January	2010	CGM3124	11	136	8,23876570669849	0,748978700608953
2	13	January	2010	BNR3128	11	97	7,86047233977133	0,714588394524666
3	11	January	2010	YIP1802	11	143	7,69781570266652	0,699801427515138
4	6	January	2010	MJW2032	11	80	7,65631315960366	0,696028469054878
5	15	January	2010	WCR2136	11	99	7,34428412306387	0,667662193005806
6	12	January	2010	JCF0800	11	102	7,22618940732767	0,656926309757061
7	12	January	2010	MQS3541	11	75	7,20543813579624	0,655039830526931

και περιγράφονται γεγονότα του χρήστη CGM3124 στις 11 Ιανουαρίου 2010. Οι ενέργειες του εκείνη την ημέρα είναι:

	id	date	usr	pc	role	is_inactive	logon	logoff	connect	disconnect	file_open	file_write	file_copy	file_delete	email_send	email_has_file	www_upload	non_workday	non_workhours	threat_score	Scored_Labe
	52838	2010-01-11 06:31:00.0000000	CGM3124	PC-6002	Physicist	1	1	0	0	0	0	0	0	0	0	0	0	1	4	0.97097315
	58551	2010-01-11 15:52:00.0000000	CGM3124	PC-6002	Physicist	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0.39492589
	1364176	2010-01-11 07:11:54.0000000	CGM3124	PC-6002	Physicist	1	0	0	0	0	0	0	0	0	0	0	0	1	2	0.51314416
	1364177	2010-01-11 07:11:54.0000000	CGM3124	PC-6002	Physicist	1	0	0	0	0	0	0	0	0	1	1	0	1	2	0.51667828
	1364182	2010-01-11 07:12:48.0000000	CGM3124	PC-6002	Physicist	1	0	0	0	0	0	0	0	0	1	1	0	1	2	0.51667828
	1364185	2010-01-11 07:13:04.0000000	CGM3124	PC-6002	Physicist	1	0	0	0	0	0	0	0	0	0	1	0	1	2	0.51474455
	1364282	2010-01-11 07:22:01.0000000	CGM3124	PC-6002	Physicist	1	0	0	0	0	0	0	0	0	1	1	0	1	2	0.51667828
	1364289	2010-01-11 07:22:13.0000000	CGM3124	PC-6002	Physicist	1	0	0	0	0	0	0	0	0	0	0	0	1	2	0.51314416
	1377622	2010-01-11 11:58:57.0000000	CGM3124	PC-6002	Physicist	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0.41969530
0	1378903	2010-01-11 12:24:16.0000000	CGM3124	PC-6002	Physicist	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0.42129569
1	1389323	2010-01-11 15:50:17.0000000	CGM3124	PC-6002	Physicist	1	0	0	0	0	0	0	0	0	1	1	0	0	1	0.42322795

Ο χρήστης CGM3124 είναι ο Casey Graiden Mann ο οποίος είναι Φυσικός και στις 11 Ιανουαρίου 2010 θα έπρεπε να είναι ανενεργός:



Results Messages

	id	date	usr	pc	role	is_inactive	logon	logoff	connect	disconnect	file_open	file_write	file_copy	file_delete	email_send	email_has_file	www_upload	non_workday	non_workhours	threat
1	393235	2010-02-25 07:03:00.0000000	RNB2679	PC-3486	ComputerProgrammer	1	1	0	0	0	0	0	0	0	0	0	0	0	1	4

Συμπερασματικά, όλα τα σημεία «1» τα οποία κατηγοριοποιήθηκαν ως πιθανές εσωτερικές απειλές, επιβεβαιώνεται ότι ισχύει και για τους δυο αλγόριθμους. Τα σημεία «2», ο SVM επιβεβαίωσε ότι ισχύει ενώ ο PCA όχι. Βλέπουμε λοιπόν, ότι ο SVM σε αυτή τη δοκιμή, απέδωσε καλύτερα από ό,τι ο PCA.

Κεφάλαιο 5
Συμπεράσματα

5.1 Συμπεράσματα

Στόχος της παρούσας διατριβής ήταν να διερευνήσουμε κατά πόσο οι πολιτικές ασφαλείας των οργανισμών, όταν υλοποιηθούν με κάποιο σύστημα ανίχνευσης, είναι χρηστικές προς αυτόν και μπορούν να αποφέρουν κάποια θετικά αποτελέσματα στην επιβεβαίωση μιας πιθανής εσωτερικής απειλής. Πιο συγκεκριμένα, στο dataset που χρησιμοποιήθηκε, οι αλγόριθμοι μας ανίχνευσαν πιθανές εσωτερικές απειλές και επιβεβαιώσαμε μερικές από αυτές.

Τα στάδια που ακολουθήσαμε περιλάμβαναν: 1) την υλοποίηση δύο πολιτικών ασφαλείας προγραμματιστικά (απάντηση στο 1ο ερευνητικό ερώτημα), 2) προεπεξεργασία των δεδομένων και εισαγωγή σε μια βάση, 3) εκπαίδευση του αλγόριθμου Linear Regression για βαθμολόγηση των ενεργειών των χρηστών και έπειτα βαθμολόγηση των ενεργειών από τα δεδομένα του dataset, 4) εκπαίδευση των SVM και PCA αλγορίθμων με χρήση των βαθμολογημένων δεδομένων από τον LR και έπειτα βαθμολόγηση των ενεργειών από τα δεδομένα του dataset, 5) σύγκριση

των αποτελεσμάτων τους και 6) επιβεβαίωση της ύπαρξης εσωτερικών απειλών, δειγματοληπτικά από τα αποτελέσματα και των δύο.

Όπως είδαμε, οι πολιτικές ασφαλείας που περιγράψαμε ήταν είτε περιοριστικές είτε όχι, και το σύστημα ΤΝ το οποίο κατασκευάσαμε, κατάφερε να εντοπίσει πιθανές εσωτερικές απειλές, (απαντώντας έτσι στο 2ο ερευνητικό ερώτημα), οι οποίες επιβεβαιωνόταν και από τις δύο πολιτικές. Το σύστημα μας, σύμφωνα με την Πολιτική Β, μπορεί να ανιχνεύσει πιθανές απειλές πολλών κατηγοριών ταυτόχρονα και να τις απεικονίσει ανάλογα στις γραφικές παραστάσεις, απ' όπου και επιβεβαιώνουμε ότι το πλήθος τους είναι μεγαλύτερο από το αντίστοιχο της Πολιτικής Α (απαντώντας στο 3ο ερευνητικό ερώτημα) (εικόνες 33-35, 40-41, 44-45). Επίσης, είδαμε ότι και αρκετοί χρήστες συνδεόταν στα συστήματα ενώ έπρεπε να είναι ανενεργοί. Επομένως, ανακαλύψαμε μια ευπάθεια στο ΠΣ και απαντήσαμε στο 4ο ερευνητικό ερώτημα.

5.2 Σύγκριση μεθοδολογιών

Η βασική διαφορά αυτού του συστήματος σε σύγκριση με άλλα, είναι ότι στηρίχθηκε σε συγκεκριμένες πολιτικές ασφαλείας, οι οποίες όριζαν τι θεωρείται εσωτερική απειλή και με γνώμονα αυτή την γνώση, κατηγοριοποιούσε τις ενέργειες των χρηστών, σε σύγκριση με άλλες έρευνες οι οποίες, είτε τυχαία επέλεγαν κάποιο γεγονός από την καθημερινότητα ενός χρήστη, είτε άφηναν το σύστημα που υλοποιούσαν να επιλέξει μόνο του τι είναι εσωτερική απειλή, ενώ δεν κάλυπταν παρά μόνο λίγες περιπτώσεις. Σε μερικές μάλιστα, ενώ αναφέρεται η ύπαρξη πολιτικών, δεν περιγράφεται πως υλοποιούνται.

Το σύστημα ΤΝ που υλοποιήθηκε, επεξεργάστηκε ένα dataset με στοιχεία, αλλά όχι σε πραγματικό χρόνο, σε σύγκριση με άλλες μελέτες που χρησιμοποίησαν αυτή τη μέθοδο. Το σύστημα μας έχει τη δυνατότητα να το κάνει, εάν μετατραπεί σε Web Service (με τη χρήση του Azure ML Studio) χωρίς μεταβολή στον κώδικα του, και έτσι να χρησιμοποιηθεί από άλλες εφαρμογές για την ανίχνευση απειλών ή την δημιουργία alerts σε πραγματικό χρόνο.

Σε άλλες μελέτες, χρησιμοποιήθηκαν και παραλλαγές νευρωνικών δικτύων ώστε να είναι δυνατό να «μάθουν» τη συμπεριφορά του χρήστη και ει δυνατό, σε πραγματικό χρόνο ενώ υπήρχε η δυνατότητα αναπροσαρμογής του μοντέλου στα δεδομένα, ώστε να γίνει εφικτή η αναγνώριση της συμπεριφοράς ενός κακόβουλου χρήστη ο οποίος προσπαθεί να μιμηθεί την κανονική συμπεριφορά. Το σύστημα μας, είναι ανθεκτικό σε τέτοιου είδους προσπάθειες, διότι η πολιτική

ασφάλειας είναι συγκεκριμένη και έτσι δεν μεταβάλλεται το μοντέλο κατά τη διάρκεια της λειτουργίας του.

Επίσης, καινοτόμες μελέτες χρησιμοποίησαν οπτικοποίηση των ενεργειών του χρήστη και αναγνώριση της εσωτερικής απειλής με γραφικό τρόπο. Κι όμως, αυτές οι μελέτες κάλυπταν μια συγκεκριμένη περίπτωση κακόβουλης δραστηριότητας. Σε περίπτωση που χρειαστεί να καλυφθούν περισσότερες κατηγορίες, όπως στη δική μας, είναι αναμενόμενο ότι η επεξεργαστική ισχύς και ο χρόνος που χρειάζεται για την αναγνώριση τους, είναι πολλαπλάσιος του δικού μας.

5.3 Επόμενα βήματα

Το σύστημα μας, χρησιμοποίησε τη βάση δεδομένων, μόνο κατά τη διάρκεια της εκπαίδευσης των μοντέλων. Η εξέλιξη του ***exypnos***, προβλέπει τη μετατροπή του σε Web Service ώστε να δημιουργηθεί ένα ολοκληρωμένο σύστημα ανίχνευσης που θα περιέχει:

- UI για την δημιουργία, καταχώρηση και μεταβολή κανόνων πολιτικής ασφαλείας,
- Επεξεργασία των γεγονότων σε πραγματικό χρόνο,
- UI για την ειδοποίηση των υπευθύνων ασφαλείας και δυνατότητα δημιουργίας alerts,
- Δυνατότητα εκμάθησης από τις απαντήσεις των υπευθύνων ασφαλείας,
- Διερεύνηση χρήσης άλλων αλγόριθμων και σύγκριση της απόδοσης τους,
- Διερεύνηση της πιθανότητας να ανιχνεύονται και άλλες ευπάθειες σε ένα ΠΣ ενός οργανισμό, ως αποτέλεσμα της εντατικής ανίχνευσης.

Βιβλιογραφία

[1] "2018 Cost of Insider Threats: Global." Ponemon Institute, Apr-2018.

[2] "Common Sense Guide to Mitigating Insider Threats, Fifth Edition." [Online]. Available: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=484738. [Accessed: 18-Oct-2018].

[3] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining and Knowledge Discovery, vol. 2, pp. 121–167, 1998.

[4] J. Breier and J. Branišová, "Anomaly Detection from Log Files Using Data Mining Techniques," in Information Science and Applications, Berlin, Heidelberg, 2015, pp. 449–457.

[5] J. Breier and J. Branišová, "A Dynamic Rule Creation Based Anomaly Detection Method for Identifying Security Breaches in Log Records," Wireless Personal Communications, vol. 94, no. 3, pp. 497–511, Jun. 2017.

[6] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, "Deep Learning for Unsupervised Insider Threat Detection in Structured Cybersecurity Data Streams," arXiv:1710.00811 [cs, stat], Oct. 2017.

[7] F. Yuan, Y. Cao, Y. Shang, Y. Liu, J. Tan, and B. Fang, "Insider Threat Detection with Deep Neural Network," in Computational Science – ICCS 2018, vol. 10860, Y. Shi, H. Fu, Y. Tian, V. V. Krzhizhanovskaya, M. H. Lees, J. Dongarra, and P. M. A. Sloot, Eds. Cham: Springer International Publishing, 2018, pp. 43–54.

[8] F. T. Liu, K. Ming Ting, and Z.-H. Zhou, "Isolation-Based Anomaly Detection," ACM Transactions on Knowledge Discovery From Data - TKDD, vol. 6, pp. 1–39, Mar. 2012.

[9] P. A. Legg, O. Buckley, M. Goldsmith, and S. Creese, "Caught in the act of an insider attack: detection and assessment of insider threat," in 2015 IEEE International Symposium on Technologies for Homeland Security (HST), Waltham, MA, 2015, pp. 1–6.

[10] O. Lo, W. J. Buchanan, P. Griffiths, and R. Macfarlane, "Distance Measurement Methods for Improved Insider Threat Detection," Security and Communication Networks, 2018. [Online]. Available: https://www.hindawi.com/journals/scn/2018/5906368/. [Accessed: 27-Mar-2019].

[11] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection," arXiv:1607.00148 [cs, stat], Jul. 2016.

[12] Ε. Κεραυνού, Τεχνητή Νοημοσύνη και Έμπειρα Συστήματα. ΠΑΤΡΑ: ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ, 2000.

[13] Π. Αργυράκης, Νευρωνικά Δίκτυα και Εφαρμογές. ΠΑΤΡΑ: ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ, 2001.

[14] "Sabbatini, R.M.E.: Neurons and Synapses: The History." [Online]. Available: http://www.cerebromente.org.br/n17/history/neurons1_i.htm. [Accessed: 06-May-2019].

[15] Jolliffe, Ian. Principal component analysis. New York: Springer Verlag, 2002. .

[16] "Insider Threat Test Dataset." [Online]. Available: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099. [Accessed: 18-Oct-2018].

[17] garyericson, "What is - Azure Machine Learning Studio." [Online]. Available: https://docs.microsoft.com/en-us/azure/machine-learning/studio/what-is-ml-studio. [Accessed: 10-May-2019].

[18] xiaoharper, "Evaluate model performance - Azure Machine Learning Studio." [Online]. Available: https://docs.microsoft.com/en-us/azure/machine-learning/studio/evaluate-model-performance. [Accessed: 26-May-2019].

[19] "Principal Component Analysis - Azure Machine Learning Studio | Microsoft Docs." [Online]. Available: https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/principal-component-analysis. [Accessed: 27-May-2019].

[20] "[1404.1100] A Tutorial on Principal Component Analysis." [Online]. Available: https://arxiv.org/abs/1404.1100. [Accessed: 21-May-2019].

[21] J. Platt, B. Schölkopf, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution," Nov. 1999.

[22] S. Kobayashi, A. Piegat, J. Pejaś, I. El Fray, and J. Kacprzyk, Eds., Hard and Soft Computing for Artificial Intelligence, Multimedia and Security, vol. 534. Cham: Springer International Publishing, 2017.

[23] M. Pietras, "Hidden Markov Models with Affix Based Observation in the Field of Syntactic Analysis," in Hard and Soft Computing for Artificial Intelligence, Multimedia and Security, 2017, pp. 17–26.

[24] V. Rokhlin, A. Szlam, and M. Tygert, "A randomized algorithm for principal component analysis," arXiv:0809.2274 [stat], Sep. 2008.

[25] J. Feit, "Machine Learning 101: Is Predictive Analytics Possible in Your Facility? Learn how to gather data and use machine learning to your advantage," Buildings, vol. 112, no. 7, p. 22, Jul. 2018.

[26] "Method and System for Improving Security Threats Detection in Communication Networks," 2014.

[27] "Need to Know: Ai and Machine Learning," Tech & Learning, vol. 39, no. 2, pp. 25–27, Sep. 2018.

[28] "Proactive Insider Threat Detection through Graph Learning and Psychological Context - IEEE Conference Publication." [Online]. Available: https://ieeexplore.ieee.org/document/6227698. [Accessed: 27-May-2019].

[29] "Use of Domain Knowledge to Detect Insider Threats in Computer Activities - IEEE Conference Publication." [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6565230. [Accessed: 27-May-2019].

[30] "Αντιμετώπιση εσωτερικών απειλών με χρήση τεχνητής νοημοσύνης σε ευρέως διαδεδομένο dataset." [Online]. Available: https://kypseli.ouc.ac.cy/handle/11128/3703. [Accessed: 27-May-2019].

ΠΑΡΑΡΤΗΜΑ Α

Κώδικας

A.1 main.py

```
from common import merge_file, load_ldap_data
from prep_ldap import *
import csv
import os

dataset_version = "r6.2"
csv_path = "../csv_files/"+dataset_version+"/"

# Prepare and Load ldap data in list
prep_ldap()
ldap_list = load_ldap_data()

# define the CSV files for merging
files_list = ['logon.csv', 'device.csv', 'file.csv', 'email.csv', 'http.csv']
# merge
[merge_file(open(os.path.join(csv_path, files)), ldap_list) for files in
files_list]
```

A.2 prep_ldap.py

```
def prep_ldap():
    dataset_version = "r6.2"
    csv_path = "../csv_files/" + dataset_version + "/LDAP/"

    files_list = ["2009-12.csv", "2010-01.csv", "2010-02.csv", "2010-03.csv",
"2010-04.csv", "2010-05.csv",
                  "2010-06.csv", "2010-07.csv", "2010-08.csv", "2010-09.csv",
"2010-10.csv", "2010-11.csv",
                  "2010-12.csv", "2011-01.csv", "2011-02.csv", "2011-03.csv",
"2011-04.csv", "2011-05.csv"]

    merged_list = []


    def check_if_exists(row):
        i=0
        while i<len(merged_list):
            if row[1] in merged_list[i][1]:
                merged_list[i][4]=row[4]
                merged_list[i][5]=row[5]
                return True
            i+=1
        return False


    for f in files_list:
        with open(csv_path+f) as csv_in:
            original_line = csv.reader(csv_in, delimiter=',')
            print("Reading from " + f, ": OK")
            line_count = 0
            row1 = []
            last_active_month = f[5:7]
            last_active_year = f[:4]

            for row in original_line:
                #Prepare the headers
                if line_count == 0:
                    #Convert 'user_id' field name to 'user'
                    row1.append(row[0])
                    row1.append('user')
                    row1.append(row[2])
                    row1.append(row[3])
                    # row1.append(row[4])
                    row1.append('last_active_month')
                    row1.append('last_active_year')

                    if row1 not in merged_list:
```

```
                        merged_list.append(row1)
                    line_count += 1
                    print("Read the headers: OK")
                    row1=[]
                else:
                    #Copy the rest of the lines
                    row1.append(row[0])
                    row1.append(row[1])
                    row1.append(row[2])
                    row1.append(row[3])
                    row1.append(last_active_month)
                    row1.append(last_active_year)

                    if not check_if_exists(row1):
                        merged_list.append(row1)
                    line_count += 1
                    row1=[]
                    # print("Written "+str(line_count)+" rows")

    print("Total length: ", len(merged_list))

    #Save everything in CSV file
    with open('../csv_files/'+dataset_version+'/ldap_'+dataset_version+'.csv',
mode='w') as csv_out:
        updated_line = csv.writer(csv_out, delimiter=',')
        for row in merged_list:
            updated_line.writerow(row)
```

A.3 common.py

```
from datetime import datetime
import csv
import os
import pyodbc
import time

# SQL Server params
server = 'tcp:*******.database.windows.net'
database = '********'
username = '********'
password = '********'
cnxn = pyodbc.connect('DRIVER={ODBC Driver 17 for SQL
Server};SERVER='+server+';DATABASE='+database+';UID='+username+';PWD='+
password)
cursor = cnxn.cursor()
cnxn.autocommit = False
```

```
# File manipulation data
dataset_version = "r6.2"
csv_path = "../csv_files/"+dataset_version+"/"


def check_weekday(date_to_check):
    weekday = datetime.strptime(date_to_check, '%m/%d/%Y %H:%M:%S').weekday()
    if weekday < 5:
        return 0
    else:
        return 1


def check_workhours(time_to_check):
    worktime = datetime.strptime(time_to_check, '%m/%d/%Y %H:%M:%S').hour
    if worktime < 8 or worktime > 17:
        return 1
    else:
        return 0


def check_date(event_date, last_user_date):
    event_month_year = datetime.strptime(event_date, '%m/%d/%Y %H:%M:%S')
    user_month_year = datetime.strptime(last_user_date, "%m%Y")
    if event_month_year<user_month_year:
        return 0
    elif event_month_year >= user_month_year:
        return 1


def calc_threat_score(row):
    # Policy A
    # return (row[4] + row[5]) * (row[16]+1) * (row[17]+1)

    # Policy B
    return
(row[4]*5+row[5]*1+row[7]*2+row[11]*2+row[13]*2+row[14]*3+row[15]*4)*(row[16]+
1)*(row[17]+1)


def load_ldap_data():
    # open the ldap file to retrieve data
    csv_ldap = open(csv_path + 'ldap_' + dataset_version + '.csv')
    read_ldap = csv.reader(csv_ldap, delimiter=',')
    ldap_list = []
    for ldap_row in read_ldap:
        ldap_list.append(ldap_row)
    return ldap_list


def check_ldap_user(usr, ldap_list):
```

```
    i=0
    while i <len(ldap_list):
        if usr ==  ldap_list[i][1]:
            return ldap_list[i][3], ldap_list[i][4], ldap_list[i][5]
        i +=1

# TODO: Investigate if check_ldap_user can be optimized

def process_batch(chunk_for_sql, file_name, ldap_list, last_chunk):
    # Batch Params
    batch_sql = []
    batch_sql_size = 10000


    # create the list with zeros
    row1 = [0] * 19

    for row in chunk_for_sql:
        # Get the first four fields as is
        row1[0] = row[1]
        row1[1] = row[2]
        row1[2] = row[3]

        # role
        usr = check_ldap_user(row[2], ldap_list)
        row1[3] = usr[0]
        # is_inactive
        row1[4]=check_date(row[1],usr[1]+usr[2])

        # Calculate Logon/Logoff activity, return 0,1
        if file_name == "logon":
            # logon
            if row[4] == "Logon":
                row1[5] = 1
            # logoff
            elif row[4] == "Logoff":
                row1[6] = 1

        # Calculate Device Connection, return 0,1
        if file_name == "device":
            # connect
            if row[5] == "Connect":
                row1[7] = 1
            # disconnect
            elif row[5] == "Disconnect":
                row1[8] = 1

        if file_name == "file":
            # file_open
            if row[5] == "File Open":
                row1[9] = 1
```

```
            # file_write
            elif row[5] == "File Write":
                row1[10] = 1
            # file_copy
            elif row[5] == "File Copy":
                row1[11] = 1
            # file_delete
            elif row[5] == "File Delete":
                row1[12] = 1

        # Calculate email sending action with attachments, return 0,1
        if file_name == "email":
            # email_send
            if row[8] == "Send":
                row1[13] = 1
            # email_has_file
            if row[10] != "":
                row1[14] = 1

        if file_name == "http":
            # www_upload
            if row[5] == "WWW upload":
                row1[15] = 1

        if file_name != 'ldap_' + dataset_version:
            # Calculate non_workday, return 0,1
            row1[16] = check_weekday(row[1])

            # Calculate non_workhours return 0,1
            row1[17] = check_workhours(row[1])

        # threat_score
        row1[18] = calc_threat_score(row1)

        batch_sql.append(row1)
        # Start inserting in MS SQL
        if (len(batch_sql) % batch_sql_size == 0):# and len(batch_sql) > 0:
            cursor.executemany("INSERT INTO r62_dataset (date, usr, pc, role,
is_inactive, logon, logoff, connect, disconnect, file_open, file_write,
file_copy, file_delete, email_send, email_has_file, www_upload,
non_workday,non_workhours, threat_score) VALUES
(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)",batch_sql)
            cnxn.commit()
            batch_sql=[]
        elif last_chunk == 1 and len(chunk_for_sql) == len(batch_sql):
            cursor.executemany(
                "INSERT INTO r62_dataset (date, usr, pc, role, is_inactive,
logon, logoff, connect, disconnect, file_open, file_write, file_copy,
file_delete, email_send, email_has_file, www_upload,
non_workday,non_workhours, threat_score) VALUES
(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)",
```

```
                batch_sql)
            print("Last batch inserted")
            cnxn.commit()
            batch_sql = []

        row1 = [0] * 19


def merge_file(csv_in, ldap_list):
    # Open file for read
    read_line = csv.reader(csv_in, delimiter=',')
    print("Opened file: ", os.path.basename(csv_in.name))

    batch_size = 10001
    csv_chunk = []
    max_batches = 40

    # Take the name of the file without extension
    file_name = os.path.basename(os.path.splitext(csv_in.name)[0])
    line_count = 0
    start_total = time.time()
    batch_counter = 0

    for i, line in enumerate(read_line):
        if line_count == 0:
            print("Read headers: OK")
            line_count += 1
        else:
            if (i % batch_size == 0 and i >0) and (batch_counter
<max_batches):
                # TODO: there is a bug in this Boolean clause, so it is
activated for i-1 and not i. Thus, the batch_size must be +1
                # start measuring duration
                start = time.time()
                process_batch(csv_chunk, file_name, ldap_list, 0)
                del csv_chunk[:]
                batch_counter += 1
                # Display metrics
                end = time.time()
                print("Batch #" + str(batch_counter) + " merged and sent in ",
round(end - start, 2))
            elif batch_counter >= max_batches:
                break
            else:
                csv_chunk.append(line)

    # Process the remaining
    process_batch(csv_chunk, file_name, ldap_list, 1)
    print("Total duration for file is: ", round(time.time() - start_total,2))
```

A.4 user_score.py

```
from datetime import datetime
from common import load_ldap_data
import csv
import os
import pyodbc
import time

# SQL Server params
server = 'tcp:********.database.windows.net'
database = '********'
username = '********'
password = '********'
cnxn = pyodbc.connect('DRIVER={ODBC Driver 17 for SQL
Server};SERVER='+server+';DATABASE='+database+';UID='+username+';PWD='+
password)
cursor = cnxn.cursor()
cnxn.autocommit = False

# Load LDAP data
ldap_list = load_ldap_data()
source_table = "r62_scored_A"

def clean_tables(ldap_list):
    # 1st way: search table name in ldap_list. !SAFE WAY!
    i=0
    print("Started Dropping Tables")
    start_total = time.time()
    while i< len(ldap_list):
        usrname = ldap_list[i][1]
        table_name = 'r62_'+usrname+'_A'
        # cursor.execute("DROP TABLE IF EXISTS "+table_name)
        cursor.execute("IF OBJECT_ID('"+table_name+"', 'U') IS NOT NULL DROP
TABLE "+table_name)
        i += 1
        if i % 100 == 0:
            print("Dropped " + str(i) + " tables")

    cnxn.commit()
    print("Total duration for DROPs is: ", round(time.time() - start_total,
2))

def groupping():
    i=0
    while i< len(ldap_list):
        usrname =ldap_list[i][1]
        table_name = 'r62_'+usrname+'_A'
        i +=1
```

```
        cursor.execute("SELECT DATEname(dd,date) AS event_day,
DATEname(mm,date) as event_month, DATEname(yyyy,date) AS event_year, usr,
COUNT(*) AS NumEvents, sum(Scored_Labels) as threat_score_per_day,
sum(Scored_Labels)/COUNT(*) as mean_threat_score into "+table_name+" FROM
"+source_table+" WHERE usr='"+username+"' group by DATEname(dd,date),
DATEname(mm,date), DATEname(yyyy,date),usr")
        cnxn.commit()

# Clean the existing tables
clean_tables(ldap_list)
# Start grouping the events per user per day
groupping()
```