ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΠΡΟΒΛΕΠΤΙΚΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΕΠΙΧΕΙΡΗΣΙΑΚΩΝ ΔΙΕΡΓΑΣΙΩΝ ΜΕ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΜΑΚΡΑΣ ΚΑΙ ΒΡΑΧΕΙΑΣ ΜΝΗΜΗΣ

ΜΑΛΩΝΑΣ ΚΩΝΣΤΑΝΤΙΝΟΣ
Α.Μ. 71345226

Εισηγητής: Δρ Αλέξανδρος Μπουσδέκης, Καθηγητής

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΠΡΟΒΛΕΠΤΙΚΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΕΠΙΧΕΙΡΗΣΙΑΚΩΝ ΔΙΕΡΓΑΣΙΩΝ ΜΕ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΜΑΚΡΑΣ ΚΑΙ ΒΡΑΧΕΙΑΣ ΜΝΗΜΗΣ

ΜΑΛΩΝΑΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

Α.Μ. 71345226

Επιβλέπων καθηγητής:

Δρ Αλέξανδρος Μπουσδέκης

Συν-Επιβλέπων καθηγητής:

Δρ Γεώργιος Μιαούλης

Εξεταστική Επιτροπή:

Δρ Γεώργιος Μιαούλης

Δρ Αθανάσιος Βουλόδημος

Δρ Γεώργιος Μπαρδής

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/η κάτωθι υπογεγραμμένος/η
Μαλωνάς Κωνσταντίνος του Μαλωνά Εμμανουήλ , με αριθμό μητρώου 71345226 φοιτητής/τρια του Προγράμματος Μεταπτυχιακών Σπουδών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:
«Είμαι συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.
Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».
Επιθυμώ την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας μου μέχρι 23/7/21 και έπειτα από αίτηση μου στη Βιβλιοθήκη και έγκριση του επιβλέποντα καθηγητή.

Ο/Η Δηλών/ούσα

Athanasios Voulodimos
Digitally signed by Athanasios Voulodimos
Date: 2021.07.28 14:24:57 +03'00'

Georgios Bardis
Digitally signed by Georgios Bardis
Date: 2021.07.29 15:30:37 +03'00'

Georgios Miaoulis
Digitally signed by Georgios Miaoulis
DN: c=GR, l=Athens, o=University of West Attica, ou=Class B - Private Key created and stored in software CSP, sn=Miaoulis, givenName=Georgios, serialNumber=3179983822, cn=Georgios Miaoulis, email=gmiaoul@uniwa.gr
Date: 2021.07.25 20:57:39 +03'00'

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε για το τμήμα Μηχανικών Πληροφορικής και Υπολογιστών του Πανεπιστημίου Δυτικής Αττικής. Η ιδέα αυτής της εργασίας προήλθε από τον καθηγητή μου κ. Αλέξανδρο Μπουσδέκη, ο οποίος με την πολύτιμη βοήθεια και καθοδήγηση του με ώθησε στο να πετύχω το καλύτερο δυνατό αποτέλεσμα. Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη της κατά την διάρκεια των σπουδών μου και όχι μόνο.

ΠΕΡΙΛΗΨΗ

Σε αυτή την διπλωματική εργασία χρησιμοποιούμε το αναδρομικό νευρωνικό δίκτυο LSTM για να προβλέψουμε την επόμενη δραστηριότητα σε έναν τραπεζικό οργανισμό αλλά και σε ένα εργοστάσιο. Μέσω του πειραματισμού προσπαθούμε να καταλήξουμε στην κατασκευή του βέλτιστου μοντέλου με το οποίο θα μπορούμε να κάνουμε ακριβείς προβλέψεις. Επίσης γίνονται πειράματα με κλασσικούς αλγορίθμους εξόρυξης δεδομένων.

ABSTRACT

The present thesis concerns the development of a LSTM neural network to predict the next activity at a bank organization and at a factory. Through experimentation we try to construct a model that leads to predictions with highest accuracy. Also we are doing experiments at the same datasets with classic process mining algorithms.

Περιεχόμενα

Κεφάλαιο 1

1.1 Εξόρυξη διεργασιών

Το πλήθος των καλά ορισμένων διεργασιών σε μια επιχείρηση θεωρείται βασικός δείκτης της επιχειρησιακής ωριμότητας της. Λειτουργούν όμως όλες οι διεργασίες αποτελεσματικά, και τελικά προσδίδουν αξία στην επιχείρηση; Το ερώτημα αυτό

προσπαθεί να απαντήσει η Εξόρυξη Διεργασιών, μια σύγχρονη τεχνική που βασίζεται στην ανάλυση δεδομένων. Η Εξόρυξη Διεργασιών δεν περιορίζεται σε κάποιο θεωρητικό μοντέλο της διεργασίας, παρά αξιοποιεί καταγεγραμμένα γεγονότα από πολλαπλές εκτελέσεις της, τα οποία αντλεί από βάσεις δεδομένων ώστε να συνθέσει την πραγματική εικόνα της διεργασίας. Η ραγδαία εξέλιξη της τεχνολογίας, ο όγκος των δεδομένων και η απαιτητικότητα των σημερινών πελατών, έχει καταστήσει την διαχείριση έργων εξαιρετικά περίπλοκη για τις επιχειρήσεις. Οι ευέλικτες μεθοδολογίες επιχειρούν να απαντήσουν σε αυτές τις προκλήσεις διαιρώντας την εκτέλεση του έργου σε επαναληπτικούς κύκλους για την πιο αποτελεσματική διαχείριση των εργασιών, δίνοντας ιδιαίτερη βάση στην προσαρμοστικότητα διατηρώντας την ευελιξία στις αλλαγές μέχρι και την τελευταία στιγμή της υλοποίησης του έργου, και επιδιώκοντας εντονότερη συμμετοχή του πελάτη για την συνεχή αναθεώρηση των απαιτήσεων. Τα συστήματα διαχείρισης και παρακολούθησης εργασιών, αποτελούν πλέον απαραίτητα εργαλεία για την αποτελεσματική εφαρμογή των ευέλικτων μεθοδολογιών (ανάθεση εργασιών στα μέλη της ομάδας, εκτίμηση χρόνου ολοκλήρωσης της κάθε εργασίας και χρονοπρογραμματισμός). Τα συστήματα αυτά έχουν την δυνατότητα να εξάγουν χρήσιμες αναφορές, συνήθως όμως περιορίζονται σε στατιστικά στοιχεία και δεν επικεντρώνονται στην καθαυτή διεργασία. Τα ψηφιακά αποτυπώματα που αφήνουν αυτού του τύπου τα συστήματα, μπορούν να αξιοποιηθούν από εργαλεία εξόρυξης διεργασιών για την ανάλυση των διαδικασιών, σε μια προσπάθεια αξιολόγησης και βελτίωσης του επιχειρησιακού σχεδίου [1],[2].

1.2 Αναδρομικά νευρωνικά δίκτυα και LSTM

Το αναδρομικό νευρωνικό δίκτυο (**RNN**) είναι ένας αλγόριθμος βαθιάς μάθησης, ο οποίος χρησιμοποιείται για τη μοντελοποίηση πληροφοριών σε σειριακή μορφή. Ο Δρ.Robert Hecht-Nielsen όρισε το νευρωνικό δίκτυο (Rahman et al. 2014) ως ένα υπολογιστικό σύστημα το οποίο αποτελείται από απλά στοιχεία επεξεργασίας, με μεγάλο βαθμό διασύνδεσης, τα οποία διαχειρίζονται τις πληροφορίες με δυναμική ανταπόκριση στις εξωτερικές εισόδους. Το βασικό πλεονέκτημα αυτών των νευρωνικών δικτύων είναι η δυνατότητα μοντελοποίησης της επόμενης κατάστασης με βάση την προηγούμενη και σε μερικές παραλλαγές (LSTM) επιπλέον πλεονέκτημα είναι η μνήμη που διαθέτουν από παλαιότερες καταστάσεις [1],[2].

1.3 Ποιο είναι το πρόβλημα που λύνουμε

Στην παρούσα διπλωματική εργασία δείχνουμε μέσω του πειραματισμού με το LSTM, πως γίνετε από αυτό να εξάγουμε ακριβής προβλέψεις για το ποιο θα είναι το επόμενο επιχειρησιακό activity σε μια διεργασία. Διεξάγουμε πειράματα ώστε να καταλήξουμε στην βέλτιστη αρχιτεκτονική του LSTM μοντέλου που θα κάνει τις προβλέψεις. Όπως θα δούμε στην συνέχεια κάνουμε πειράματα για δύο περιπτώσεις και με δύο dataset. Για την περίπτωση ενός τραπεζικού καταστήματος και για την περίπτωση μιας βιομηχανικής εγκατάστασης. Επίσης εξετάζουμε μέσω κλασσικών αλγορίθμων για εξόρυξη διεργασιών το κατά πόσο οι προβλέψεις που θα πάρουμε από το LSTM θα είναι ακριβείς.

1.4 Τι αποτελέσματα παίρνουμε

Συγκεκριμένα και για τις δύο περιπτώσεις τυπώνουμε τις τρείς επόμενες δραστηριότητες που προβλέπει το νευρωνικό μας δίκτυο, ότι ακολουθούν την

δραστηριότητα που λαμβάνει χώρα την παρούσα χρονική στιγμή, εμείς επικεντρωνόμαστε μόνο στην ακριβώς επόμενη δραστηριότητα.

1.5 Πως οδηγούμαστε στην επίλυση του προβλήματος

Μέσω του πειραματισμού με τις διάφορες παραμέτρους του LSTM μοντέλου μας προσπαθούμε να οδηγηθούμε στην κατασκευή ενός μοντέλου που θα αποτελείται από τον κατάλληλο αριθμό στρωμάτων, τον σωστό αριθμό νευρώνων και γενικότερα οι παράμετροι του, όπως το dropout, το activation function, ο optimizer και τα epochs θα είναι ρυθμισμένοι με τον σωστό τρόπο ώστε να παίρνουμε σωστές προβλέψεις σε άγνωστα δεδομένα.

1.6 Γιατί επιλέξαμε LSTM

Με το LSTM καταφέρνουμε και παρακάμπτουμε τα όποια προβλήματα δημιουργούνται από παλαιότερες αρχιτεκτονικές νευρωνικών δικτύων, επιτυγχάνοντας καλύτερη ακρίβεια στις προβλέψεις μας. Συγκεκριμένα το LSTM σχεδιάστηκε για να μοντελοποιεί χρονικές ακολουθίες και τις μεγάλου εύρους εξαρτήσεις τους με μεγαλύτερη ακρίβεια από άλλους τύπους RNN.

1.7 Άλλες εφαρμογές του LSTM

Τα LSTM αναδρομικά δίκτυα μπορούν να μάθουν τις εξαρτήσεις που υπάρχουν μεταξύ των ενεργειών που εκτελούνται σε μια διεργασία ώστε να λύσουν προβλήματα ακολουθιακών προβλέψεων. Λίγοι από τους τομείς εφαρμογής του LSTM είναι η αναγνώριση ομιλίας, η πρόβλεψη ανόδου ή καθόδου της τιμής μιας μετοχής, η πρόβλεψη του χρόνου που θα εκτελεστεί μια ενέργεια, η αναγνώριση και μετάφραση της νοηματικής γλώσσας κ.α. [1],[2].

Κεφάλαιο 2

2.1 Διαχείριση Επιχειρησιακών Διαδικασιών – BPM

Η Διαχείριση Επιχειρησιακών Διαδικασιών (Business Process Management ή BPM) περιλαμβάνει όλες εκείνες τις έννοιες, τις μεθόδους και τις απαραίτητες τεχνικές προκειμένου για τον σχεδιασμό, την ανάλυση, την υλοποίηση και την εν τέλει την διαχείριση των επιχειρησιακών διαδικασιών. Το θεμέλιο της Διαχείρισης των Επιχειρησιακών Διαδικασιών είναι η σαφής αναπαράσταση των διαδικασιών μιας επιχείρησης και των εργασιών/δραστηριοτήτων από τις οποίες αποτελούνται, μέσα σε σαφή οριοθετημένα πλαίσια. Από τη στιγμή που οι επιχειρησιακές διαδικασίες οριστούν, μπορούν να αναλυθούν, να βελτιωθούν και τελικά να εφαρμοστούν. [3] Ουσιαστικά είναι μια επιχειρηματική πρακτική που σκοπός της είναι να τυποποιήσει και να βελτιστοποιήσει της δραστηριότητες της επιχείρησης υπό το πρίσμα των διαδικασιών, μέσω της συνεχής παρουσίας της στη λειτουργία της επιχείρησης. [4]

2.1.1 Οι αρχές της Διαχείρισης Επιχειρησιακών Διαδικασιών

Η γενική ιδέα του BPM συνοψίζεται στις παρακάτω βασικές – αδιαμφισβήτητες αρχές [5]:

Όλες οι εργασίες είναι μέρος μιας διαδικασίας.

Μερικές φορές γίνεται η υπόθεση ότι οι όροι: διαδικασία και BPM εφαρμόζονται σε αυστηρά δομημένες εργασίες όπως πχ. η εκτέλεση παραγγελιών, η εξυπηρέτηση πελατών ή ακόμα σε δημιουργικές εργασίες όπως η ανάπτυξη προϊόντος κ.α. Παρερμηνεύονται επίσης ως συνώνυμο του αυτοματισμού και της ρουτίνας. Τίποτα δεν θα μπορούσε όμως να απέχει περισσότερο από την αλήθεια. Διαδικασία σημαίνει ο συνδυασμός ατομικών εργασιακών δραστηριοτήτων - ρουτίνας ή δημιουργικών - στο ευρύτερο πλαίσιο άλλων δραστηριοτήτων ώστε να δημιουργηθεί ένα αποτέλεσμα.

Οποιαδήποτε διαδικασία είναι καλύτερη από καμία διαδικασία.

Όταν απουσιάζει μια καλά καθορισμένη διαδικασία, το χάος βασιλεύει. Ατομικές κουλτούρες, ιδιοτροπίες, και αυτοσχεδιασμοί διέπουν τις εργασίες και τα αποτελέσματα είναι ασυνεπή και μη βιώσιμα. Μια καλά καθορισμένη διαδικασία θα παραδώσει τουλάχιστον προβλέψιμα, επαναλαμβανόμενα αποτελέσματα, και θα αποτελέσει αφορμή για βελτίωση.

Μια καλή διαδικασία είναι καλύτερη από μια κακή διαδικασία.

Αυτή η δήλωση δεν είναι τόσο ταυτολογική όπως φαίνεται. Εκφράζει την κρισιμότητα του σχεδιασμού της διαδικασίας, καθώς είναι κρίσιμος παράγοντας της απόδοσης της. Εάν μια εταιρεία επιβαρύνεται με ένα κακό σχεδιασμό διαδικασίας, πρέπει να το αντικαταστήσει με έναν καλύτερο.

Μία έκδοση διαδικασίας είναι καλύτερη από πολλές διαδικασίες.

Τυποποιώντας τις διαδικασίες σε όλα τα τμήματα μιας επιχείρησης παρουσιάζεται ένα ενιαίο πρόσωπο στους πελάτες και τους προμηθευτές και εσωτερικά επιτυγχάνεται οικονομία κλίμακας σε υπηρεσίες υποστήριξης, όπως η εκπαίδευση

και τα πληροφοριακά συστήματα επιτρέποντας την ανακατανομή των ανθρώπων από τη μία επιχειρηματική μονάδα στην άλλη, κ.α. Τα οφέλη αυτά πρέπει να σταθμίζονται έναντι των διαφορετικών αναγκών των τμημάτων και των πελατών τους και να προάγεται η τυποποίηση.

Ακόμη και μια καλή διαδικασία πρέπει να εκτελείται αποτελεσματικά.

Μία καλά σχεδιασμένη διαδικασία είναι μια αναγκαία αλλά ανεπαρκής προϋπόθεση για υψηλές επιδόσεις. Χρειάζεται να συνδυαστεί και με μια προσεκτική διαχείριση εκτέλεσης ώστε να φανεί στην πράξη η αξία της.

Ακόμη και μια καλή διαδικασία μπορεί να γίνει καλύτερη.

Ο υπεύθυνος της διαδικασίας πρέπει να είναι συνεχώς σε εγρήγορση, αναζητώντας ευκαιρίες για να κάνει τροποποιήσεις στο σχεδιασμό της διαδικασίας προκειμένου να ενισχύσει περαιτέρω την απόδοσή της.

Κάθε καλή διαδικασία γίνεται τελικά μια κακή διαδικασία.

Καμία διαδικασία δεν μένει αποτελεσματική για πάντα. Οι ανάγκες των πελατών αλλάζουν, οι τεχνολογίες αλλάζουν, ο ανταγωνισμός αλλάζει και οι πρώην καλές διαδικασίες θα πρέπει να αντικατασταθούν με νέες.

2.1.2 Λόγοι έναρξης ενός έργου BPM

Πότε όμως μία εταιρία πρέπει να λάβει σοβαρά υπόψη αυτή την νέα μεθοδολογία και να την ενσωματώσει στην λειτουργική της δομή; Δυστυχώς αυτό το ερώτημα δεν μπορεί να απαντηθεί εύκολα και με ένα γενικό τρόπο. Η πραγματική απάντηση είναι: «εξαρτάται». Εξαρτάται από τις συνθήκες και την ωριμότητα της κάθε εταιρίας. Υπάρχουν όμως κάποιες ενδείξεις που μπορεί να προκαλέσουν την αφορμή ώστε να εξεταστεί το BPM ως πιθανή λύση. Τα συνήθη εναύσματα κατηγοριοποιημένα είναι [6]:

Επιχείρηση	• Υψηλή ανάπτυξη - Δυσκολία να αντιμετωπιστούν υψηλοί ρυθμοί ανάπτυξης, ή να γίνει προληπτικός σχεδιασμός για ανάπτυξη • Συγχωνεύσεις και εξαγορές. Προσδίδουν αυξημένη πολυπλοκότητα και απαιτείται εξορθολογισμός των διαδικασιών. • Αναδιοργάνωση. Αλλαγή ρόλων και υπευθυνοτήτων • Αλλαγή στρατηγικής. Αποφάσεις για αλλαγές σε λειτουργικά, προϊοντικά ή πελατειακών σχέσεων θέματα • Μη τήρηση εταιρικών σκοπών και στόχων

	• Εναρμονισμοί με πρότυπα και κανονισμούς • Ανάγκη για επιχειρηματική ευελιξία για άμεση ανταπόκριση στις ευκαιρίες που προκύπτουν • Ανάγκη για μεγαλύτερο έλεγχο
Διοίκηση – Διαχείριση	• Αναξιόπιστες ή αντικρουόμενες πληροφορίες από την διοίκηση • Ανάγκη για παροχή περισσότερων ευθυνών στους managers • Ανάγκη για δημιουργία κουλτούρας υψηλής απόδοσης • Ανάγκη για μεγαλύτερη απόδοση ή απόσβεση επενδύσεων • Περικοπές προϋπολογισμού
Εργαζόμενοι	• Υψηλός κύκλος εργασιών με πίεση και υψηλές απαιτήσεις, χωρίς κατάλληλη υποστήριξη • Θέματα εκπαίδευσης νέων εργαζομένων • Χαμηλή ικανοποίηση εργαζομένων • Επιθυμία για αύξηση ή ενδυνάμωση εργατικού δυναμικού • Δυσκολία των εργαζομένων να συμβαδίσουν με συνεχείς αλλαγές ή αυξανόμενη πολυπλοκότητα
Πελάτες – Προμηθευτές Συνεργάτες	• Χαμηλή ικανοποίηση σχετικά με την εξυπηρέτηση • Απροσδόκητη αύξηση του αριθμού των πελατών, προμηθευτών ή συνεργατών • Μεγάλοι χρόνοι για την ικανοποίηση των αιτημάτων / απαιτήσεων • Επιθυμία για εστίαση σε πελατειακές σχέσεις • Τμηματοποίηση πελατών ή απαιτήσεων • Εφαρμογή αυστηρών επιπέδων εξυπηρέτησης • Ανάγκη για διαφοροποιημένες διαδικασίες σε μεγάλους πελάτες, προμηθευτές ή συνεργάτες • Η ανάγκη για μια πραγματική από άκρη σε άκρη (end-to-end) προοπτική
Αγαθά και υπηρεσίες	• Απαράδεκτοι χρόνοι παράδοσης στην αγορά

	• Άθλια επίπεδα εξυπηρέτησης • Παρόμοιες ή κοινές διαδικασίες σε διαφορετικά προϊόντα ή υπηρεσίες • Πολύπλοκα αγαθά ή υπηρεσίες
Διαδικασίες	• Ανάγκη για διαφάνεια των διαδικασιών μέσω μιας προοπτικής από άκρη σε άκρη (end-to-end) • Ασαφείς διαδικασίες ή κενά σε αυτές • Ασαφείς ρόλοι και αρμοδιότητες • Πολύ συχνή ή καθόλου αλλαγή διαδικασιών • Έλλειψη τυποποίησης • Έλλειψη ξεκάθαρων στόχων ή σκοπών • Έλλειψη επικοινωνίας και κατανόησης από τους εμπλεκομένους
Πληροφοριακά συστήματα	• Εισαγωγή νέων συστημάτων • Προμήθεια εργαλείων αυτοματοποίησης BPM, χωρίς να υπάρχει η γνώση για να αξιοποιηθούν • Εισαγωγή διαδικτυακών υπηρεσιών

2.1.3 Παράγοντες επιτυχίας ενός έργου BPM

Η ενσωμάτωση και τελικώς η υλοποίηση ενός έργου BPM σε μία επιχείρηση είναι μια περίπλοκη διαδικασία, καθώς τέμνει τα τμήματα, διαπερνά τα οργανωτικά τους όρια και περιλαμβάνει διαφορετικές και πολύπλοκες σχέσεις των εμπλεκόμενων τόσο εντός όσο και εκτός της επιχείρησης [6].

Παρακάτω αναλύονται κάποιες «τεχνικές» που οι επιχειρήσεις μπορούν να εφαρμόσουν ώστε να εξασφαλίσουν την επιτυχία κατά την υλοποίηση των έργων τους BPM.

- **Ανάπτυξη στρατηγικής BPM:**
  Αυτό είναι το σημείο εκκίνησης για κάθε εφαρμογή BPM. Έχοντας μια στρατηγική BPM που ευθυγραμμίζεται πλήρως με τους επιχειρηματικούς στόχους της επιχείρησης είναι ο πρώτος και πιο κρίσιμος παράγοντας επιτυχίας. Μέσω της δομημένης και συστηματικής προσέγγισης η στρατηγική πρέπει να περιλαμβάνει μια λεπτομερή επιχειρησιακή περίπτωση που να δείχνει τη διαφορά μεταξύ της τρέχουσας προσέγγισης που χρησιμοποιεί η επιχείρηση και των οφελών που αποκομίζονται σε περίπτωση υιοθέτησης του BPM. Επίσης θα πρέπει να καθορίζει ένα πλαίσιο παράδοσης που να επικεντρώνεται πρώτα σε σχετικά απλά, εφικτά έργα που έχουν σαφές επιχειρηματικό όφελος [7].

- **Δέσμευση Εμπλεκομένων :**
  Προαπαιτούμενο για την εφαρμογή των έργων BPM είναι η επίσημη υποστήριξη των ανώτατων στελεχών. Είναι πολύ σημαντικό τα υψηλά στελέχη να δίνουν την απαιτούμενη προσοχή και υποστήριξη στην υλοποίηση των έργων BPM. Περαιτέρω, η δέσμευση από τα μεσαία στελέχη (**middle management**) είναι επίσης κρίσιμη καθώς ορισμένα μέλη του προσωπικού μπορεί να εμφανίσουν μεταβολή των ρόλων και των ευθυνών τους μέσα από τις νέες ή βελτιωμένες διαδικασίες. Αρκετά συχνά παρατηρείται ότι όταν τα βασικά διευθυντικά στελέχη δεν εμπλέκονται επαρκώς, δεν αποδέχονται τις διαδικασίες για διάφορους λόγους (συνήθως από το φόβο ότι θα χάσουν τη δουλειά τους). Η διοίκηση θα πρέπει να αποσκοπεί ώστε να δημιουργεί το κατάλληλο περιβάλλον όπου ο καθένας έχει την ευελιξία να αποδίδει τα μέγιστα [7].

- **Αποτελεσματική διαχείριση αλλαγών μέσω του προσωπικού:**
  Οι διαδικασίες σχεδόν πάντα έχουν στενή σχέση τόσο με τους ανθρώπους όσο και με την τεχνολογία. Είναι ζωτικής σημασίας τα άτομα τα οποία εμπλέκονται στην εφαρμογή των διαδικασιών να υποστηρίζουν το έργο. Η έρευνα έχει δείξει ότι η διαχείριση των ανθρώπινων αλλαγών μπορεί να καταλαμβάνει από 25% έως 35% του χρόνου του έργου, του κόστους και της προσπάθειας. Ως εκ τούτου, είναι απαραίτητο η υπεύθυνη ομάδα BPM να πρέπει να δαπανά χρόνο και προσπάθεια για τη διαχείριση των ανθρώπινων αλλαγών [7].

- **Συνεργασία σε ολόκληρη την επιχείρηση:**
  Ένα σημαντικό πρώτο βήμα που πρέπει να κάνουν οι managers είναι να δείξουν στους υπαλλήλους ποια είναι η διαδικασία διαχείρισης επιχειρησιακών διαδικασιών. Αρχικά πρέπει να παρουσιάσουν μια γενική εικόνα των βασικών διαδικασιών του οργανισμού. Αυτή η επισκόπηση, που μερικές φορές ονομάζεται χαρτογράφηση, δείχνει με απλό τρόπο τις διαδικασίες του οργανισμού που αφορούν τον πελάτη και τις διαδικασίες που τις υποστηρίζουν. Η χαρτογράφηση των διαδικασιών δείχνει επίσης τη σχέση μεταξύ διαδικασιών και οντοτήτων και χαρτογραφεί τη θεμελιώδη συνεργασία εντός του οργανισμού. Όταν λείπει η συνεργασία και ο συντονισμός μεταξύ των διαδικασιών των διαφόρων οντοτήτων, αυξάνεται ο κίνδυνος προσφοράς κακών προϊόντων στον πελάτη, καθώς και δημιουργίας περιττών δαπανών. Συνεπώς η επισκόπηση των βασικών διαδικασιών πρέπει να περιλαμβάνει τις διεπαφές που υπάρχουν μεταξύ των διαφόρων οντοτήτων. Η συνεργασία δημιουργεί έναν κοινό στόχο που ενισχύσει την επιτυχία της επιχείρησης [8].

- **Καθιέρωση διακυβέρνησης BPM :**
  Κανένα έργο BPM δεν μπορεί να υλοποιηθεί χωρίς να υπάρχει και το κατάλληλο μοντέλο διακυβέρνησης. Οι επιχειρήσεις που αναπτύσσουν μια στρατηγική BPM πρέπει να στήσουν έναν ουδέτερο «φορέα διακυβέρνησης» ώστε να θέσει προτεραιότητες και διαβαθμίσεις και να καθιερώσει αποτελεσματικό έλεγχο [7].

- **Πρακτικότητα διαδικασιών:**
  Οι διαδικασίες αφορούν τον συντονισμό, την τυποποίηση των δραστηριοτήτων και την επαναληψιμότητα των αποτελεσμάτων τους. Προκειμένου να καταστούν τα αποτελέσματα επαναλαμβανόμενα, θα μπορούσε κανείς να σκεφτεί ότι οι διαδικασίες θα πρέπει να περιγράφονται ως υπερβολικά λεπτομερή έγγραφα που δεν αφήνουν περιθώρια για δημιουργικότητα, παρέμβαση ή συλλογιστική. Αυτό μπορεί να είναι κατάλληλο για ορισμένες αυτοματοποιημένες διαδικασίες μεγάλου όγκου. Ωστόσο, για διαδικασίες με ανθρώπινη διαδραστικότητα, αυτό δεν είναι ένας αποτελεσματικός τρόπος καταγραφής διαδικασιών. Στις περιπτώσεις αυτές είναι σημαντικό να παρουσιάζονται τα βασικά βήματα που ενέχονται στη δημιουργία του προϊόντος ή της υπηρεσίας και οι αλληλεπιδράσεις μεταξύ των διαφορετικών ρόλων, καθώς και η ροή πληροφοριών και υλικού μεταξύ των οντοτήτων της εν λόγω διαδικασίας. Η προσπάθεια να επιτευχθεί μια λεπτομερής περιγραφή και των πιο μικρών καθηκόντων μέσα σε μια επιχείρηση από την αρχή, με την εισαγωγή του BPM, είναι αδύνατο και μπορεί να οδηγήσει σε ασυντόνιστες διαδικασίες. Οι διαδικασίες πρέπει πρώτα απ' όλα να τεκμηριώσουν την αλληλεξάρτηση τους με άλλες διαδικασίες και τον τρόπο που συνεργάζονται οι οντότητες εντός της επιχείρησης [8].

- **Η εμπλοκή της τεχνολογίας :**
  Αμέτρητες λύσεις πληροφορικής μπορούν να συμβάλουν στην ενίσχυση της αποδοτικότητας και της αποτελεσματικότητας των επιχειρησιακών διαδικασιών. Η επιλογή, η υιοθέτηση και η αξιοποίηση της πληροφορικής θα πρέπει να αποτελούν αναπόσπαστο μέρος του BPM, το οποίο μπορεί να υποστηρίξει την επιχείρηση. Η εισαγωγή κατάλληλων επιχειρηματικών συστημάτων μπορεί να συμβάλουν σημαντικά στην διαχείριση των διαδικασιών καθώς και σε έργα ανασχεδιασμού, υποστηρίζοντας τις προσπάθειες συνεχούς βελτίωσης [9].

- **Η σημασία της συνέχειας :**
  Το BPM εισάγεται συχνά σε μια επιχείρηση μέσω βραχυπρόθεσμων έργων που στοχεύουν στην επίλυση συγκεκριμένων προβλημάτων. Ωστόσο, είναι σημαντικό να προχωρήσουμε πέρα από την επίτευξη μόνο γρήγορων λύσεων. Η αρχή της συνέχειας τονίζει ότι το BPM πρέπει να αποτελεί μόνιμη πρακτική που διευκολύνει τη συνεχή αύξηση της αποδοτικότητας και της αποτελεσματικότητας. Η εδραίωση και εφαρμογή μιας μακροπρόθεσμης προσέγγισης BPM είναι σημαντικά στοιχεία προκειμένου να αξιοποιηθούν οι δυνατότητες και η να φανεί η αξία του BPM. Είναι βέβαιο ότι κάθε μεμονωμένο έργο μπορεί να έχει οφέλη, αλλά στην καλύτερη περίπτωση θα δημιουργήσει ένα προσωρινό βέλτιστο που σύντομα θα χάσει έδαφος, καθώς το οικονομικό περιβάλλον και ο ανταγωνισμός συνεχίζουν να εξελίσσονται. Προκειμένου να αποφευχθεί το BPM να είναι ένα έργο αλλαγής έκτακτης ανάγκης πρέπει να καλλιεργηθεί η κατάλληλη νοοτροπία υποστήριξης διαδικασιών. Αυτό μπορεί να γίνει με τη δημιουργία και τη διατήρηση μιας

οργανωτικής κουλτούρας που υποστηρίζει το BPM και το διευκολύνει ως φυσικό μέρος της καθημερινής εργασίας [9].

- **Η αξία του σκοπού:**
  Το BPM δεν πρέπει να υιοθετηθεί επειδή είναι στη μόδα ή είναι "ο τρόπος να κάνουμε πράγματα". Θα πρέπει να εξυπηρετεί έναν συγκεκριμένο σκοπό που έχει στρατηγική σημασία για την επιχείρηση. Θα πρέπει να ευθυγραμμίζεται με τους επιχειρηματικούς στόχους ώστε να δημιουργεί αξία [9].

2.1.4 Κρίσιμες πτυχές κατά τη υλοποίηση έργων BPM

Η υλοποίηση έργων BPM αφορούν κατά κύριο λόγο στην ισορροπία και στη συνοχή μεταξύ των επιχειρηματικών και τεχνολογικών πλευρών. Η εξισορρόπηση τους θα επιτρέψει την ολοκλήρωση BPM έργων με τον πιο αποτελεσματικό και αποδοτικό τρόπο εστιάζοντας στα παρακάτω βασικά σημεία [6]:

- **Ταχύτητα (αποτελεσματικότητα):**
  Η ταχύτητα είναι κρίσιμη επειδή γενικός στόχος είναι η νίκη και η νίκη έρχεται με το να φτάνεις πρώτος στην γραμμή τερματισμού. Σε ένα έργο BPM στόχος είναι η εστίαση σε ρεαλιστικά οφέλη από τις επιχειρησιακές διαδικασίες και η γρήγορη επίτευξή τους.

- **Αποδοτικότητα:**
  Η αποδοτικότητα σχετίζεται με την εξασφάλιση ότι όλα τα εμπλεκόμενα μέρη αποδίδουν στον μέγιστο βαθμό. Σε ένα έργο BPM πρέπει να εξασφαλιστεί ότι όλοι συνεισφέρουν επαρκώς για να πραγματοποιήσουν τα επιθυμητά αποτελέσματα.

- **Ισορροπία:**
  Η ισορροπία επιτυγχάνεται λαμβάνοντας υπόψη όλα τα στοιχεία υλοποίησης (διοίκηση, διαδικασίες, άνθρωποι, διαχείριση έργου, πόροι και πληροφορίες) στο βαθμό που πρέπει.

- **Συνοχή:**
  Η συνοχή εξασφαλίζει ότι όλα τα στοιχεία υλοποίησης είναι ευθυγραμμισμένα και δεν λειτουργούν ανεξάρτητα.
- **Διαδικασίες:**
  Οι διαδικασίες παίζουν τον σημαντικότερο ρόλο καθώς πρέπει να πρωταγωνιστούν και η τεχνολογία και οι άνθρωποι να ακολουθούν.

- **Διοίκηση:**
  Η διοίκηση εξασφαλίζει την πορεία των εργασιών και τις ευθυγραμμίζει με τους επιχειρηματικούς στόχους.

2.1.5 Οι διαδικασίες στην επιχείρηση

Οι σύγχρονες εξελίξεις και οι τάσεις που καταγράφονται διεθνώς στον τρόπο λειτουργίας των επιχειρήσεων αυξάνουν τη σημασία των επιχειρησιακών διαδικασιών. Μερικοί από τους παράγοντες που έχουν συμβάλει στο γεγονός αυτό είναι: **i)** αυξάνονται οι επιχειρήσεις που μεταβάλλουν το πλέον παραδοσιακό οργανωτικό σχήμα σε οριζόντιο σχήμα οργάνωσης, **ii)** επικεντρώνονται στις διατμηματικές εργασίες και διαδικασίες και όχι στις ενδο-τμηματικές λειτουργίες, **iii)** ο προσανατολισμός στην εξυπηρέτηση και ικανοποίηση του πελάτη και στην παροχή ποιοτικά καλύτερων προϊόντων και υπηρεσιών, **iv)** η ανάγκη για συνεργασία με άλλες επιχειρήσεις κ.α. Η οργάνωση των επιχειρήσεων με βάση τις διαδικασίες διευκολύνει την προσαρμοστικότητα, την ευελιξία, καθώς και την συμβατότητα τους κατά τη συνεργασία τους με άλλες επιχειρήσεις/οργανισμούς. Το πρώτο σημαντικό βήμα είναι η αναγνώριση της ανάγκης, το επόμενο στάδιο όμως είναι η εφαρμογή της, η οποία προϋποθέτει επιστημονικές μεθόδους και εργαλεία προκειμένου για την ανάλυση, την συνεχή βελτίωση και προσαρμογή των διαδικασιών, με τέτοιο συστηματικό τρόπο ώστε να υποστηρίζεται η αρχιτεκτονική της λειτουργίας της επιχείρησης [3].

Παρακάτω αναλύονται βασικές τεχνικές μεθοδολογίας, καθώς και τα συστατικά μιας διαδικασίας από την οπτική των βασικών βημάτων και των εμπλεκομένων σε αυτή.

2.1.6 Μεθοδολογία

Όταν εισάγεται η διαχείριση διαδικασιών σε μια εταιρεία, είναι σημαντικό να υπάρχει κατά νου η προϋπάρχουσα αποστολή, το όραμα και η στρατηγική της επιχείρησης. Μερικές φορές, ωστόσο, μπορεί να είναι αναγκαία η αναθεώρηση της επιχειρησιακής στρατηγικής (όσον αφορά τις ανταγωνιστικές προτεραιότητες), λαμβάνοντας παράλληλα υπόψη τα πλεονεκτήματα και τις αδυναμίες της εταιρείας καθώς και τις ευκαιρίες ή τις απειλές στην αγορά. Συγκεκριμένα, οι μακροπρόθεσμοι στόχοι της εταιρείας πρέπει να προκύψουν κατά τη διάρκεια αυτής της φάσης και όλες οι διαδικασίες που θα καθοριστούν ή θα αναπτυχθούν θα πρέπει να στοχεύουν στην επίτευξη τους [10].

Κατά τους **Tonchia** και **Tramontano**, υπάρχουν οχτώ βασικά βήματα που αποτελούν βασικό άξονα της μεθοδολογίας και η οποία έχει εφαρμοστεί σε πολλές επιχειρήσεις μέχρι σήμερα:

1. **Αναγνώριση διαδικασίας:**
   Είναι ιδιαίτερα κρίσιμο καθώς θα πρέπει να προσδιοριστεί η διαδικασία και το επίπεδο λεπτομέρειας που θα είναι χρήσιμο για τη διαχείριση της διαδικασίας. Για το σκοπό αυτό χρησιμοποιείται συχνά ένας συνδυασμός προσεγγίσεων "από πάνω προς τα κάτω" (top-down) και "από κάτω προς τα πάνω" (bottom-up). Συγκεκριμένα ξεκινώντας καταγράφονται όλες οι δραστηριότητες που εμπλέκονται στη διαδικασία (στάδιο " top-down "). Εν συνεχεία εάν ο κατάλογος αποδειχθεί υπερβολικά μεγάλος και προκύψουν προβλήματα στη διαχείριση της διαδικασίας, οι δραστηριότητες ομαδοποιούνται σε δύο ή περισσότερες διαφορετικές υποδιαδικασίες (στάδιο " bottom-up ").

2. **Ορισμός πλαισίου :**
   Καθορίζονται τα όρια της διαδικασίας και ορίζεται ή έναρξη (που μπορεί να είναι π.χ. προμηθευτής) και η λήξη αυτής (π.χ. πελάτης)

3. **Καθορισμός εισροών και εκροών:**
   Ως εισροές ορίζονται οι πρώτες ύλες, οι πληροφορίες και τα δεδομένα που είναι απαραίτητα για την παραγωγή του τελικού προϊόντος ή της υπηρεσίας και ως εκροές τα τελικά προϊόντα ή υπηρεσίες

4. **Καθορισμός δραστηριοτήτων και εργασιών:**
   Προσδιορίζονται όλες οι εργασίες που περιλαμβάνονται στην διαδικασία.

5. **Ανάλυση του χρονικού πλαισίου:**
   Αφορά στα γεγονότα που ενεργοποιούν τις εργασίες καθώς και την διάρκεια των εργασιών

6. **Αξιολόγηση της απόδοσης:**
   Αναλύονται τα αποτελέσματα που προκύπτουν

7. **Καθορισμός ευθύνης:**
   Ορίζεται υπεύθυνος διαδικασίας

8. **Ανάθεση πόρων:**
   Καθορίζεται το εργατικό δυναμικό και τα πληροφοριακά συστήματα εφόσον απαιτούνται

Η παραπάνω ακολουθία δεν είναι απαραίτητο να εκτελείται με αυτή την σειρά, αλλά είναι σημαντικό να ελέγχεται κάθε βήμα ώστε να ελέγχεται και η βελτίωση που αναμένεται να προκύψει.

Υπό μία ευρύτερη άποψη προσανατολισμένη στο έργο, οι φάσεις της μεθοδολογίας που απαιτούνται για την ανάπτυξη εφαρμογών επιχειρησιακών διεργασιών κατά τον **Mathias Weske** είναι :

- Η **φάση της στρατηγικής και της οργάνωσης** είναι η πρώτη φάση της μεθοδολογίας. Είναι ανεξάρτητη από συγκεκριμένες λειτουργικές διαδικασίες, διότι ασχολείται με τον προσδιορισμό της συνολικής επιχειρηματικής στρατηγικής. Στη φάση αυτή καθορίζονται οι στρατηγικοί και οι επιχειρησιακοί στόχοι. Οργανωτικά η επιχείρηση πρέπει να είναι δομημένη με τέτοιο τρόπο ώστε οι επιχειρησιακές διαδικασίες να μπορούν να υλοποιηθούν με επιτυχία.

- Η **φάση της έρευνας** είναι η πρώτη φάση που αφορά καθαρά τις επιχειρησιακές διαδικασίες και τα έργα για την υλοποίηση αυτών. Σε αυτή τη φάση, καθορίζονται οι στόχοι του έργου, δημιουργείται η ομάδα του έργου και συλλέγονται πληροφορίες. Εκπονούνται εμπειρικές μελέτες με βάση τις τεχνικές συνέντευξης και γίνεται ανάλυση των διαθέσιμων εγγράφων. Ενώ οι δραστηριότητες σε αυτή τη φάση επικεντρώνονται σε επιχειρησιακούς τομείς,

θα πρέπει να εξεταστεί παράλληλα και η τεχνική εκτέλεση των διαδικασιών, καθώς μπορεί να έχει επιπτώσεις στην υλοποίηση τους.

- Στη **φάση σχεδιασμού**, οι συγκεντρωθείσες πληροφορίες αναλύονται, ενοποιούνται και χρησιμοποιούνται ως πρότυπα επιχειρησιακών διαδικασιών. Αυτά τα πρότυπα χρησιμεύουν ως βάση επικοινωνίας για τους διάφορους εμπλεκόμενους καθώς και για τη βελτίωση των διαδικασιών έτσι ώστε να υλοποιηθούν οι επιχειρησιακοί στόχοι που ορίζονται στη φάση της στρατηγική. Η βελτίωση της επιχειρησιακής διαδικασίας δεν αφορά μόνο την πραγματική διαδικασία αλλά και το τεχνικό και οργανωτικό περιβάλλον στο οποίο εφαρμόζεται. Το τεχνικό περιβάλλον μπορεί να βελτιωθεί, ώστε να χρησιμοποιηθούν προσεγγίσεις προσανατολισμένες στις υπηρεσίες που να παρέχουν μεγαλύτερη ευελιξία σε σχέση με τις παραδοσιακές προσεγγίσεις. Σε οργανωτικό επίπεδο, ενδέχεται να προκύψουν νέοι ρόλοι που απαιτούν νέες δεξιότητες και ικανότητες για την αποτελεσματικότερη υλοποίηση των επιχειρησιακών διαδικασιών και την καλύτερη εξυπηρέτηση των πελατών.

- Κατά τη **φάση επιλογής συστήματος διαχείρισης** χρησιμοποιούνται τα πρότυπα επιχειρησιακών διαδικασιών καθώς και πληροφορίες σχετικά με το τεχνικό και οργανωτικό περιβάλλον της επιχείρησης ώστε να επιλεγεί η κατάλληλη τεχνολογική πλατφόρμα διαχείρισης επιχειρησιακών διαδικασιών. Οι επιχειρησιακές διαδικασίες όμως μπορεί να υλοποιηθούν και χωρίς τεχνικές πλατφόρμες εφόσον υπάρχουν σαφείς γραπτές επιχειρηματικές πολιτικές και καλώς καθορισμένες επιχειρησιακές δραστηριότητες .

- Για την περίπτωση που επιλεγεί σύστημα BPM απαιτείται μια ειδική **φάση υλοποίησης και δοκιμής** ώστε τα πρότυπα επιχειρησιακών διαδικασιών να καταστούν εκτελέσιμα.

- Ακολουθεί η **φάση της ανάπτυξης**, όπου η υλοποίηση της διαδικασίας εφαρμόζεται στο εργασιακό περιβάλλον.

- Στη **φάση λειτουργίας και ελέγχου**, συλλέγονται πληροφορίες από τις διαδικασίες που ήδη εκτελούνται στο εργασιακό περιβάλλον. Οι πληροφορίες αυτές είναι χρήσιμες για τη βελτίωση της διαδικασίας με εξελικτικό τρόπο.

Η μεθοδολογία είναι επαναληπτική και βαθμιαία. Με τη συγκέντρωση γνώσεων σχετικά με τις επιχειρηματικές διαδικασίες και το περιβάλλον τους, προκύπτουν νέα ερωτήματα και ζητήματα που πρέπει να ληφθούν υπόψη κατά την επόμενη επανάληψη.

2.1.7 Κύκλος ζωής διαδικασίας

Κάθε επιχειρησιακή διαδικασία ακολουθεί έναν κύκλο ζωής (life cycle).

- **Καθορισμός - Αναγνώριση (Define - identification)**
  Η αναγνώριση των επιχειρησιακών διαδικασιών δεν είναι πάντα εύκολη υπόθεση, ιδιαίτερα όταν οι διαδικασίες δεν ανήκουν σ' ένα μόνο τμήμα της επιχείρησης. Πρώτο βήμα για την αναγνώριση είναι ο καθορισμός των ορίων τους. Πολύ σημαντικό για την «ανακάλυψή» τους, είναι να έχει κάποιος τη γενική εικόνα της επιχείρησης. Επιπλέον μέχρι να αναγνωριστούν και να καθοριστούν δεν έχουν συγκεκριμένα ονόματα, κάτι το οποίο μπορεί να οδηγήσει σε λάθος ερμηνείες, με κίνδυνο την έλλειψη κοινής αντίληψης και αποδοχής από τα στελέχη [4]. Η φάση του καθορισμού εξαρτάται από τους εμπλεκόμενους σε μεγάλο βαθμό. Κατά τη διάρκεια αυτής της φάσης, εναπόκειται στους "πελάτες" να ορίσουν και να περιγράψουν τις απαιτήσεις τους στον αναλυτή της διαδικασίας. Χωρίς σαφή ορισμό της διαδικασίας είναι απίθανο ο αναλυτής να είναι σε θέση να ολοκληρώσει με επιτυχία όλες τις επόμενες φάσεις. Για να ολοκληρωθεί αυτή τη δραστηριότητα, ο "πελάτης" οφείλει να αναπτύξει ένα αίτημα και το υποβάλλει στον αναλυτή . Αυτό πρέπει να είναι σε επίσημη μορφή, έτσι ώστε όλες οι εργασίες που θα λάβουν χώρα να καταγράφονται και να ανιχνεύονται. Κατά την παραλαβή του αιτήματος ορίζεται το έργο και αποφασίζεται εάν θα προχωρήσει στη φάση του σχεδιασμού. Αυτό το βήμα διασφαλίζει ότι δεν ολοκληρώνονται διπλές εργασίες και ότι το αίτημα είναι στο πλαίσιο των απαιτήσεων [7].

- **Σχεδιασμός και Ανάλυση (design and analysis).**
  Στη φάση αυτή πραγματοποιείται η οριοθέτηση των προς εξέταση διαδικασιών από οργανωτικής και τεχνολογικής πλευράς. Η συγκέντρωση όλων των απαραίτητων στοιχείων επιτυγχάνεται με τη διεξαγωγή συνεντεύξεων και έρευνας με τα εμπλεκόμενα στελέχη της επιχείρησης που σχετίζονται με τις αναλυόμενες διαδικασίες. Οι διαδικασίες που θα αναγνωρισθούν θα πρέπει να επαληθευτούν και να νομιμοποιηθούν. Ο αποτελεσματικότερος τρόπος νομιμοποίησης των καταγεγραμμένων διαδικασιών είναι η διεξαγωγή συναντήσεων (workshops) με τα εμπλεκόμενα στελέχη, τα οποία ελέγχουν κατά πόσο οι διαδικασίες και οι δραστηριότητες που αναλύθηκαν ανταποκρίνονται στα δεδομένα της επιχείρησης [3].
  Στόχος των αναλυτών, στη φάση αυτή, είναι να κατανοήσουν το πλαίσιο στο οποίο δρα η διαδικασία, να συγκεντρώσουν πληροφορίες αναφορικά με την λειτουργίας της, τα στοιχεία από τα οποία αποτελείται και τους πόρους που χρησιμοποιεί. Ειδικότερα στη φάση αυτή αναγνωρίζονται οι εισροές και οι εκροές της διαδικασίας, οι δραστηριότητες και εργασίες που πραγματοποιούνται καθώς και το εμπλεκόμενο ανθρώπινο δυναμικό. Στο σημείο αυτό εντοπίζονται και τυχόν προβλήματα που μπορεί να υπάρξουν κατά την εκτέλεση της διαδικασίας και καθορίζονται οι ρόλοι των εμπλεκομένων. Πρακτικά σε πρώτη φάση καταγράφεται η παρούσα κατάσταση ("as is") και εν συνεχεία προσδιορίζονται οι απαιτήσεις για τη μελλοντική λειτουργία της διαδικασίας ("to be"). Η ανάλυση των διαδικασιών είναι ένα στάδιο που βοηθάει την επιχείρηση να συγκεντρώσει πληροφορίες

γι' αυτές και να κατανοήσει καλύτερα τον ρόλο καθεμιάς στην επίτευξη των στόχων της επιχείρησης [4].
Στη φάση αυτή συνήθως επιλέγονται και οι κατάλληλες μέθοδοι μοντελοποίησης (modeling methods), επαλήθευσης, νομιμοποίησης και προσομοίωσης (simulation), ώστε ο αρχικά ανεπίσημος τρόπος ανάλυσης να γίνει επίσημος, με χρήση μιας αυστηρά καθορισμένης σημειογραφίας και συμβολογραφίας.

- **Ανάπτυξη (configuration).**
Η ανάπτυξη πραγματοποιείται με τον ενστερνισμό των διαδικασιών και πολιτικών, που πρέπει να εφαρμοστούν, από το προσωπικό της επιχείρησης. Υπάρχουν διαδικασίες που περιλαμβάνουν εργασίες που μπορούν να πραγματοποιηθούν χωρίς την υποστήριξη πληροφοριακών συστημάτων. Για τις περιπτώσεις όμως όπου είναι απαραίτητη η χρήση κάποιου πληροφοριακού συστήματος θα πρέπει να ληφθούν υπόψη και οι συσχετίσεις με αυτό. Η ανάπτυξη των διαδικασιών και των συσχετιζόμενων πληροφοριακών συστημάτων ακολουθείται από τον έλεγχό τους (testing) [3]. Ο έλεγχος είναι εξίσου σημαντικός διότι στην φάση αυτή μπορούν να εντοπιστούν πιθανά προβλήματα στο χρόνο λειτουργίας ή να αναγνωριστούν πρόσθετες δραστηριότητες, όπως για παράδειγμα η κατάρτιση του προσωπικού ή η μεταφορά δεδομένων (data migration) [11].

- **Παραγωγική Λειτουργία (enactment) – Εκτέλεση (execution).**
Μετά τον έλεγχο των αναπτυχθέντων διαδικασιών και πληροφοριακών συστημάτων ακολουθεί η φάση της παραγωγικής λειτουργίας τους. Στη φάση αυτή εξετάζονται στιγμιότυπα των διαδικασιών που εκτελούνται. Με τον όρο «στιγμιότυπο» εννοούμε την αποτύπωση, σε συγκεκριμένο χρόνο λειτουργίας της διαδικασίας. Αυτό σημαίνει ότι οι εισροές, οι εκροές και οι δραστηριότητες που την απαρτίζουν είναι συγκεκριμένες. Κατά τη φάση αυτή γίνεται και η τελική διαμόρφωση τους, έτσι ώστε να ανταποκρίνονται σε ρεαλιστικές συνθήκες [4].
Μια εναλλακτική λύση για τη μέθοδο αυτή είναι η εφαρμογή της νέας διαδικασίας σε φάσεις και όχι συνολικά. Αυτό θα επιτρέψει στην επιχείρηση τη σταδιακή μετάβαση από το παλιό καθεστώς στο νέο, και να δοθεί ο απαραίτητος χρόνος για αναθεώρηση και ανατροφοδότηση [7].

- **Επίβλεψη/Έλεγχος (monitoring):**
Στη φάση αυτή η λειτουργία των διαδικασιών παρακολουθείται κατά τον χρόνο της εκτέλεσής τους. Συγκεντρώνονται δεδομένα που αφορούν τις διαδικασίες και υπολογίζονται διάφοροι δείκτες. Σκοπός είναι να παραχθούν κατάλληλα συμπεράσματα σε σχέση με την αποδοτικότητα και την αποτελεσματικότητά των διαδικασιών αναφορικά με τους επιχειρηματικούς στόχους [4].
Στο σημείο αυτό διασφαλίζεται ότι η διαδικασίας θα εφαρμοστεί όπως προβλεπόταν στο αρχικό μοντέλο και θα επιτευχθούν τα αναμενόμενα αποτελέσματα. Οι δραστηριότητες ελέγχου μπορεί να αποτελούνται από

επίσημες εκθέσεις ανατροφοδότησης, online ερωτηματολογίων, ή/και διεξαγωγή συναντήσεων εργασίας (workshops) [7].

- **Βελτιστοποίηση (Optimization)**
  Η βελτιστοποίηση της διαδικασίας περιλαμβάνει την αξιοποίηση των πληροφοριών για την απόδοση της διαδικασίας από τις προηγούμενες φάσεις. Προσδιορίζονται τα δυνητικά ή πραγματικά σημεία συμφόρησης και οι πιθανές ευκαιρίες για εξοικονόμηση κόστους ή άλλες βελτιώσεις και στη συνέχεια, εφαρμόζονται στον σχεδιασμό της διαδικασίας, ώστε να δημιουργηθεί μεγαλύτερη επιχειρηματική αξία [12].

- **Διαχείριση (administration).**
  Η διαχείριση των διαδικασιών παίζει σημαντικό ρόλο στον κύκλο ζωής των διαδικασιών καθώς επωμίζεται το καθήκον να παρακολουθεί τις διαδικασίες σε όλα της τα στάδια και μέσω των πληροφοριών που συγκεντρώνει, να τις αξιολογεί και να επισημαίνει σημεία προς βελτίωση. Η διαχείριση των διαδικασιών πραγματοποιείται από όλους τους συμμετέχοντες σε αυτή (stakeholders) [3].

- **Ανασχεδιασμός (reengineering).**
  Η φάση αυτή έχει να κάνει με την επανεξέταση του σχεδιασμού των διαδικασιών και την συσχέτιση μεταξύ τους. Απώτερος σκοπός είναι να γίνουν ριζικές αλλαγές σε κρίσιμες για την επιχείρηση διαδικασίες και πολύ πιθανό να οδηγήσει στην κατάργηση κάποιων ή/και στην εμφάνιση νέων. Παράδειγμα τέτοιων διαδικασιών μπορεί να είναι αυτές που επηρεάζουν το κόστος ή την ποιότητα των προϊόντων/υπηρεσιών ή την ικανοποίηση των πελατών [4].

2.1.8 Ρόλοι και Εμπλεκόμενοι

Οι διαδικασίες δεν αποτελούν αυτοσκοπό, δεδομένου ότι είναι απλά ένα μέσο για να επιτευχθούν οι επιχειρηματικοί στόχοι. Οι στόχοι δεν επιτυγχάνονται μέσω των διαδικασιών αυτόματα ή κατά τύχη, αλλά με συνεχόμενη και αποτελεσματική διαχείριση, με την βοήθεια της τεχνολογίας και των ανθρώπων [6].

Οι άνθρωποι είναι πάντα ο πυρήνας της επιχείρησης και είναι το ανθρώπινο κεφάλαιο που βοηθά να γίνονται οι διαδικασίες αποτελεσματικές και αποδοτικές, ανεξάρτητα πόσο αυτές είναι αυτοματοποιημένες [7].

Ο τομέας των επιχειρησιακών διαδικασιών χαρακτηρίζεται από διαφορετικές κατηγορίες εμπλεκομένων, με διαφορετικές γνώσεις, εξειδικεύσεις, προσόντα και εμπειρία [11].

Για μια αποτελεσματική διαχείριση θα πρέπει να υπάρχουν κατ' ελάχιστον οι παρακάτω ρόλοι:

- **Διευθυντής Διαδικασιών :** είναι υπεύθυνος για την διαχείριση των διαδικασιών καθώς επίσης για την προτυποποίηση τους και τον εναρμονισμό τους στην επιχείρηση. Παράλληλα είναι υπεύθυνος για την εξέλιξη τους

λαμβάνοντας υπόψη την συνεχή διαμόρφωση των επιχειρηματικών αναγκών. Η θέση του στην οργανωτική δομή, δείχνει την έμφαση που δίνει η εταιρεία στις διαδικασίες [3].

- **Σχεδιαστής (ή Αναλυτής) Διαδικασιών :** είναι υπεύθυνος για την ανάπτυξη των μοντέλων των διαδικασιών, συλλέγοντας πληροφορίες από τους εμπλεκομένους, οπότε διαθέτει τόσο τεχνικές όσο και επιχειρηματικές γνώσεις [3]. Επιπρόσθετα διασφαλίζει ότι όλες οι διαδικασίες συμμορφώνονται με τα πρότυπα και τις κατευθυντήριες γραμμές και αρχές της επιχείρησης και παρέχει συμβουλές, κάνει συστάσεις ή προτείνει λύσεις για βελτιώσεις. Παράλληλα είναι υπεύθυνος για την κατάρτιση και εκπαίδευση του προσωπικού σε νέες διαδικασίες καθώς και των σχετικών με αυτές εγχειρίδιων χρήσης [7].

- **Συμμετέχοντες διαδικασιών :** είναι όσοι εκτελούν κάποιες από τις δραστηριότητες των διαδικασιών. Ο ρόλος τους κατά την καταγραφή είναι σημαντικός αφού παρέχουν βασικές πληροφορίες για τον τρόπο με τον οποίο εκτελούνται οι εργασίες, καθώς και πληροφορίες για τις διασυνδέσεις μεταξύ των διαδικασιών. Επίσης εκφέρουν προτάσεις βελτίωσης, δεδομένης της εμπειρίας που διαθέτουν σε λειτουργικό επίπεδο [3].

- **Υπεύθυνος διαδικασίας :** είναι αυτός που έχει την ευθύνη για την εφαρμογή της διαδικασίας σύμφωνα με τα όσα περιγράφηκαν στο αντίστοιχο μοντέλο. Στις αρμοδιότητές του είναι να παρακολουθεί τη διαδικασία, να ανιχνεύει πιθανά προβλήματα και αδυναμίες και να προτείνει τρόπους βελτίωσής της σε συνεργασία με τους υπόλοιπους εμπλεκόμενους [3].

- **Υπεύθυνος αρχιτεκτονικής συστημάτων :** είναι υπεύθυνος για την ανάπτυξη και την ρύθμιση των συστημάτων διαχείρισης των διαδικασιών κατά τέτοιο τρόπο ώστε οι διαδικασίες να εναρμονίζονται με τα διαθέσιμα πληροφορικά συστήματα της επιχείρησης, ώστε να αξιοποιηθούν κατά τον καλύτερο δυνατό τρόπο οι λειτουργικότητές τους [3].

- **Μηχανικός ανάπτυξης :** είναι εργαζόμενος εξειδικευμένος στα πληροφοριακά συστήματα και αναλαμβάνει να διαμορφώσει πληροφοριακές λύσεις για την υλοποίηση των επιχειρησιακών διαδικασιών. Σημαντική του αρμοδιότητα είναι η ανάπτυξη λειτουργικών διασυνδέσεων μεταξύ διαφορετικών εφαρμογών (interfaces) [3].

Όλοι οι παραπάνω ρόλοι συνθέτουν την ομάδα που είναι υπεύθυνη για την διαχείριση των διαδικασιών. Ανάλογα το μέγεθος, την ωριμότητα και την οργανωτική δομή της επιχείρησης, οι ρόλοι μπορεί να είναι διακριτοί ή να συγχωνεύονται με άλλους.

2.1.9 Τα οφέλη του BPM

Μέσω της Διαχείρισης Επιχειρησιακών Διαδικασιών - BPM μια επιχείρηση μπορεί να δημιουργήσει διαδικασίες υψηλής απόδοσης, που λειτουργούν με πολύ χαμηλότερο κόστος, μεγαλύτερες ταχύτητες, μεγαλύτερη ακρίβεια, μειωμένο ενεργητικό, καθώς και ενισχυμένη ευελιξία [5].

Η θεμελιώδης αρχή του BPM είναι να βοηθήσει τις επιχειρήσεις να επεξεργαστούν περισσότερες υπηρεσίες και προϊόντα με μικρότερη προσπάθεια, υψηλότερη ποιότητα και μειωμένο κόστος. Συνήθως το BPM εστιάζει σε τρία βασικά οφέλη: αποδοτικότητα, αποτελεσματικότητας και ευελιξία [7].

Αναλυτικότερα σε μια ολιστική προσέγγιση το BPM παρέχει τα ακόλουθα οφέλη σε έναν οργανισμό [7] :

1. Το BPM εξοικονομεί χρόνο και χρήματα σε έναν οργανισμό:

Βοηθάει στο να προσδιοριστούν επαναλαμβανόμενες διαδικασίες και εξαλείφει επικαλυπτόμενα καθήκοντα εργασίας. Με την τυποποίηση των επιχειρησιακών διαδικασιών, οι οργανισμοί είναι σε θέση να μειώσουν τις λειτουργικές τους δαπάνες εκτελώντας επαναλαμβανόμενα διαδικασίες που επιτυγχάνουν κάθε φορά το ίδιο αποτέλεσμα. Οι διαδικασίες οι οποίες είναι τυποποιημένες είναι πιο εύκολο να αυτοματοποιηθούν. Αυτό βοηθά στην μείωση του κύκλου εργασιών καθώς μειώνονται οι σπατάλες, ενισχύεται η αποδοτικότητα και τελικά προάγεται η αύξηση της κερδοφορίας.

2. Βελτιωμένη επιχειρησιακή ευελιξία:

Η BPM ενισχύει τη ικανότητα της επιχείρησης να προσδιορίζει ενδεχόμενες ευκαιρίες ή απειλές και βοηθάει στην προτεραιοποίηση της στρατηγικής αντιμετώπισης. Με την υιοθέτηση του BPM, οι οργανισμοί έχουν τη δυνατότητα να προσθέτουν και να αφαιρούν εργασίες ή υπηρεσίες οι οποίες χαρακτηρίζονται επιθυμητές, ουσιαστικές ή ανεπιθύμητες αντίστοιχα. Επίσης επιτρέπει στους οργανισμούς να «βγαίνουν» στην αγορά με νέα προϊόντα γρηγορότερα. Η βελτιωμένη ευελιξία μιας επιχείρησης παρέχει διορατικότητα, έλεγχο και ευκινησία ώστε να ανταποκριθεί καλύτερα στις ανάγκες και τις προσδοκίες του πελάτη.

3. Ενισχυμένη επιχειρηματική ευφυΐα:

Με την αποτελεσματική καταγραφή και την παρακολούθηση των επιχειρησιακών διαδικασιών, το BPM προσφέρει τη δυνατότητα να ανιχνεύονται και να εντοπίζονται απαραίτητες πληροφορίες και να παράγονται αναφορές για την ανώτερη Διοίκηση έχοντας επίγνωση για την επίδοση των διεργασιών. Το BPM διευκολύνει τη διάδοση των πληροφοριών σε ελάχιστο χρόνο, καθώς επίσης βελτιώνει την αξιοπιστία των πληροφοριών που είναι απαραίτητες ώστε να παρθούν ευαίσθητες αποφάσεις.

4. Βελτιωμένη λειτουργική ευθύνη:

Το BPM παρέχει υψηλή ευθύνη για όλα τα τμήματα ενός οργανισμού, παρέχοντας τη δυνατότητα παρακολούθησης και ελέγχου του προϋπολογισμού και των παραδοτέων. Η τεκμηρίωση της δραστηριότητας της επιχειρησιακής

διαδικασίας βοηθά τους οργανισμούς να αναπτύξουν ένα σύστημα ελέγχων και ισορροπιών, ελαχιστοποιώντας έτσι την πιθανότητα απάτης, σφαλμάτων ή ζημιών.

5. Συνεχής βελτίωση:

Το BPM δημιουργεί ένα περιβάλλον για τη συνεχή βελτίωση της διαδικασίας σε έναν οργανισμό και διευκολύνει την ικανότητά της να εφαρμόσει τις βελτιώσεις. Βοηθά επίσης στο να αυτοματοποιηθούν οι διαδικασίες μέσω της τεχνολογίας που σχεδόν πάντα οδηγεί σε σημαντική μείωση κόστους. Η αυτοματοποίηση μειώνει την χειρωνακτική εργασία, μειώνει τον πρωταρχικό χρόνο, και αυξάνει άμεσα τον ρυθμό επεξεργασίας. Στο πλαίσιο αυτό, είναι σημαντικό να σημειώσουμε ότι οι επιτυχημένοι οργανισμοί με δυνατότητα BPM συνήθως φαίνεται να καταργούν σταδιακά ανόμοια συστήματα.

6. Διακυβέρνηση σύμφωνα με κανονισμούς και πρότυπα:

Στο σημερινό επιχειρηματικό περιβάλλον υπάρχει ένα ευρύ φάσμα κυβερνητικών κανόνων και κανονισμών που πρέπει ο κάθε οργανισμός να ακολουθήσει. Η επιτυχής εφαρμογή BPM προσπαθεί να επιτύχει αποτελεσματικά, τον καθορισμό συνεκτικών ελέγχων σε κάθε επίπεδο της διαδικασίας. Περιλαμβάνει εργαλεία, διαδικασίες, πολιτικές και επιχειρηματικές μετρήσεις που διέπουν την εταιρεία ώστε να υπάρχει πάντα «μια εκδοχή της αλήθειας». Αυτό βοηθά τον οργανισμό να παρακολουθεί τις υποχρεώσεις του, και να εξασφαλίζει ότι ακολουθεί και τηρεί τα ισχύοντα πρότυπα. Έχοντας σαφώς καθορισμένες διαδικασίες το BPM παρέχει τη δυνατότητα να αποφευχθούν δυνητικά δαπανηρές συνέπειες από μη συμμόρφωση του οργανισμού με τα απαιτούμενα πρότυπα.

7. Αποτελεσματική μέτρηση:

Ο μηχανικός της Επιστήμης των Υπολογιστών Tom De Marco είπε κάποτε «Δεν μπορείς να ελέγξεις κάτι το οποίο δεν μπορεί να μετρηθεί» Το BPM προσπαθεί να ποσοτικοποιήσει τα αποτελέσματα των επιχειρησιακών δραστηριοτήτων: κόστος, απόδοση, κύκλου ζωής, την ποιότητα, την ικανοποίηση των πελατών, ή οποιαδήποτε άλλο αποτέλεσμα χρησιμοποιώντας εργαλεία μέτρησης της παραγωγής (όπως Lean και Six Sigma). Η αποτελεσματική μέτρηση κλείνει την διαδικασία ανατροφοδότησης του κύκλου διαχείρισης της διαδικασίας, και παρέχει στα στελέχη της επιχείρησης σημαντικές πληροφορίες που μπορούν να αξιοποιήσουν για να κάνουν περαιτέρω βελτιώσεις.

8. Αποτελεσματική διαχείριση του κινδύνου:

Η καλή διαχείριση του κινδύνου αποτελεί αναπόσπαστο στοιχείο κάθε διαδικασίας. Στο BPM, οι τεκμηριωμένες διαδικασίες αναθεωρούνται και αξιολογούνται από τους αναλυτές της διαδικασίας υπό το πρίσμα του κινδύνου δια του οποίου αποτελεσματικοί έλεγχοι ενσωματώνονται σε όλες τις διαδικασίες και σε όλα τα επίπεδα του προσωπικού. Οι αναλυτές της διαδικασίας είναι σε θέση να μειώσουν σημαντικά τον συνολικό κίνδυνο σε μια οργάνωση με την επιβολή μιας αυστηρής διαχείρισης της διαδικασίας σε όλες τις επιχειρηματικές μονάδες.

9. Αποτελεσματική λειτουργική διαχείριση:

Οργανισμοί που έχουν εφαρμόσει με επιτυχία BPM είναι μάρτυρες της λειτουργικής αποδοτικότητας μέσω από συντομότερους χρονικούς κύκλους, της μείωσης του κόστους, και της ικανότητας να χειριστούν πρόσθετη εργασίες χωρίς όμως την ύπαρξη γραμμικής αύξησης του προσωπικού. Αυτό προκύπτει από τη βελτίωση της διαδικασίας και την αποτροπή χρήσης παλιών αντιπαραγωγικών μεθόδων ή πρακτικών. Κατέχοντας ένα αποτελεσματικό σύστημα διαχείρισης διαδικασιών, οι ηγέτες των επιχειρήσεων μπορούν να διατηρούν μια ολοκληρωμένη κατανόηση των δικών τους διαδικασιών, να τις μετρούν αποτελεσματικά, και να λαμβάνουν σωστές αποφάσεις για το πώς μπορούν να εξελίξουν τις επιχειρήσεις τους.

10. Ορατότητα Αποδοτικότητας.

Το BPM βελτιώνει την ορατότητα της αποδοτικότητας σε όλο το εύρος της διαδικασίας και την καθιστά εμφανή στα μέλη του προσωπικού που είναι υπεύθυνα γι' αυτήν. Με την παρακολούθηση της απόδοσης μιας διαδικασίας, ένα μέλος του προσωπικού μπορεί να αντιδράσει αναλόγως και να αποκαταστήσει τυχόν περιττές ενέργειες ή προβλήματα σε πολύ πιο γρήγορο ρυθμό. Το BPM παρέχει τα μέσα για την διεξαγωγή μετρήσεων επιδόσεων σε έναν οργανισμό και μπορεί να εμφανίσει τα αποτελέσματα σε πίνακες διαχείρισης. Οι αναλυτές της διαδικασίας μπορούν να διερευνήσουν περαιτέρω ώστε να απομονώσουν τις αιτίες της συμφόρησης, όπως χρονικές καθυστερήσεις και υψηλό κόστος επεξεργασίας.

2.2 Εξόρυξη Διεργασιών (Process Mining)

Η **εξόρυξη διεργασιών** είναι μια τεχνική διαδικασίας μάνατζμεντ, που είναι χρήσιμη για την ανάλυση επιχειρηματικών διαδικασιών βασισμένες σε καταγραφή γεγονότων. Η βασική ιδέα είναι να εξορύξουμε γνώση από καταγεγραμμένα γεγονότα αποθηκευμένα από ένα σύστημα πληροφοριών. Η εξόρυξη διεργασιών στοχεύει στο να βελτιώνει αυτό το σύστημα πληροφοριών με παρεχόμενες τεχνικές και εργαλεία για να ανακαλύψει τη διαδικασία, τον έλεγχο, τα δεδομένα και τις κοινωνικές δομές από την καταγραφή γεγονότων.

Οι τεχνικές της εξόρυξης διεργασιών συχνά χρησιμοποιούνται όταν δεν υπάρχει καμιά επίσημη περιγραφή της διαδικασίας για το πώς μπορεί να αποκτηθεί με άλλους τρόπους, ή όταν η ποιότητα μιας υπαρκτής τεκμηρίωσης είναι αμφισβητήσιμη. Για παράδειγμα, οι οικονομικοί έλεγχοι ενός συστήματος ροής μάνατζμεντ, η διεξαγωγή καταγραφής ενός συστήματος σχεδιασμού επιχειρηματικών πόρων, και το ηλεκτρονικό σύστημα καταχώρησης ενός νοσοκομείου μπορεί να χρησιμοποιηθεί για να ανακαλύψει μοντέλα που να περιγράφουν διαδικασίες, οργανισμούς και προϊόντα. Ακόμη, αυτή η καταγραφή γεγονότων μπορεί να χρησιμοποιηθεί για να συγκρίνουμε καταγεγραμμένα γεγονότα με κάποιο προγενέστερο πρότυπο για να δούμε εάν η παρατηρούμενη πραγματικότητα συμμορφώνεται σε κάποιο κατευθυντήριο ή περιγραφικό μοντέλο.

Η σύγχρονη τάση του μάνατζμεντ όπως η BAM (Business Activity Monitoring), BOM (Business Operation Management), BPI (Business Process intelligence), απεικονίζει το ενδιαφέρον στο να υποστηρίζουν τη λειτουργικότητα της διάγνωσης στο πλαίσιο της τεχνολογίας διοίκησης διεργασιών (π.χ. τα συστήματα ροής μάνατζμεντ αλλά και άλλες διαδικασίες πληροφοριακών συστημάτων).

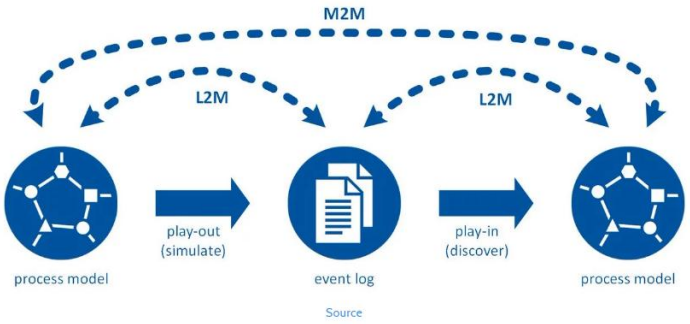
Υπάρχουν τρεις κατηγορίες τεχνικών εξόρυξης διαδικασίας. Η ταξινόμηση αυτή βασίζεται στο κατά πόσο υπάρχει ένα προγενέστερο μοντέλο και, αν ναι, πώς χρησιμοποιείται [13].

- *Ανακάλυψη*: Δεν υπάρχει ένα προγενέστερο πρότυπο, δηλαδή, με βάση ένα συμβάν ημερολογίου κάποιο μοντέλο είναι κατασκευασμένο. Για παράδειγμα, χρησιμοποιώντας τον αλγόριθμο άλφα, ένα μοντέλο διαδικασίας μπορεί να ανακαλυφθεί βασιζόμενο σε χαμηλού επιπέδου εκδηλώσεις. Υπάρχουν πολλές τεχνικές για να κατασκευάσουν αυτόματα μοντέλα διεργασίας (για παράδειγμα in terms of a Petri net), με βάση ορισμένες καταγραφές συμβάντων. Πρόσφατα, η ερευνητική διαδικασία εξόρυξης άρχισε επίσης να στοχεύει σε άλλες προοπτικές (π.χ. τα δεδομένα, τους πόρους, το χρόνο, κ.λπ.). Για παράδειγμα, η τεχνική που περιγράφεται στην (Aalst, Reijers, & song, 2005), μπορούν να χρησιμοποιηθούν για την κατασκευή ενός κοινωνικού δικτύου.

- *Συμμόρφωση*: Υπάρχει ένα προγενέστερο μοντέλο. Αυτό το μοντέλο συσχετίζεται με το αρχείο καταγραφής συμβάντων και διαφορές μεταξύ του ημερολογίου και του μοντέλου αναλύονται. Για παράδειγμα, μπορεί να υπάρχει ένα μοντέλο της διαδικασίας που αναφέρεται ότι για εντολές αγοράς άνω του 1 εκατομμυρίου ευρώ, απαιτούνται δύο έλεγχοι. Ένα άλλο παράδειγμα είναι ο έλεγχος της λεγόμενης «four-eyes» αρχής. Συμμόρφωση εξέτασης μπορεί να χρησιμοποιηθεί για την ανίχνευση αποκλίσεων για να εμπλουτίσει το μοντέλο. Ένα παράδειγμα είναι η επέκταση του μοντέλου διαδικασίας με δεδομένα απόδοσης, δηλαδή, ένα προγενέστερο μοντέλο διαδικασίας που χρησιμοποιείται για την προβολή των σημείων συμφόρησης. Ένα άλλο παράδειγμα είναι ο ανθρακωρύχος που περιγράφονται στην απόφαση (Rozinat & Aalst, 2006β), η οποία παίρνει ένα πρότυπο διαδικασίας και αναλύει κάθε επιλογή στο μοντέλο διαδικασίας. Για κάθε επιλογή στο αρχείο καταγραφής συμβάντων είναι η γνώμη για να δείτε ποιες πληροφορίες είναι συνήθως διαθέσιμες τη στιγμή που η επιλογή γίνεται. Τότε κλασικές τεχνικές εξόρυξης δεδομένων χρησιμοποιούνται για να δείτε ποια στοιχεία των δεδομένων επηρεάζουν την επιλογή. Ως αποτέλεσμα, ένα δέντρο απόφαση παράγεται για κάθε επιλογή της διαδικασίας.

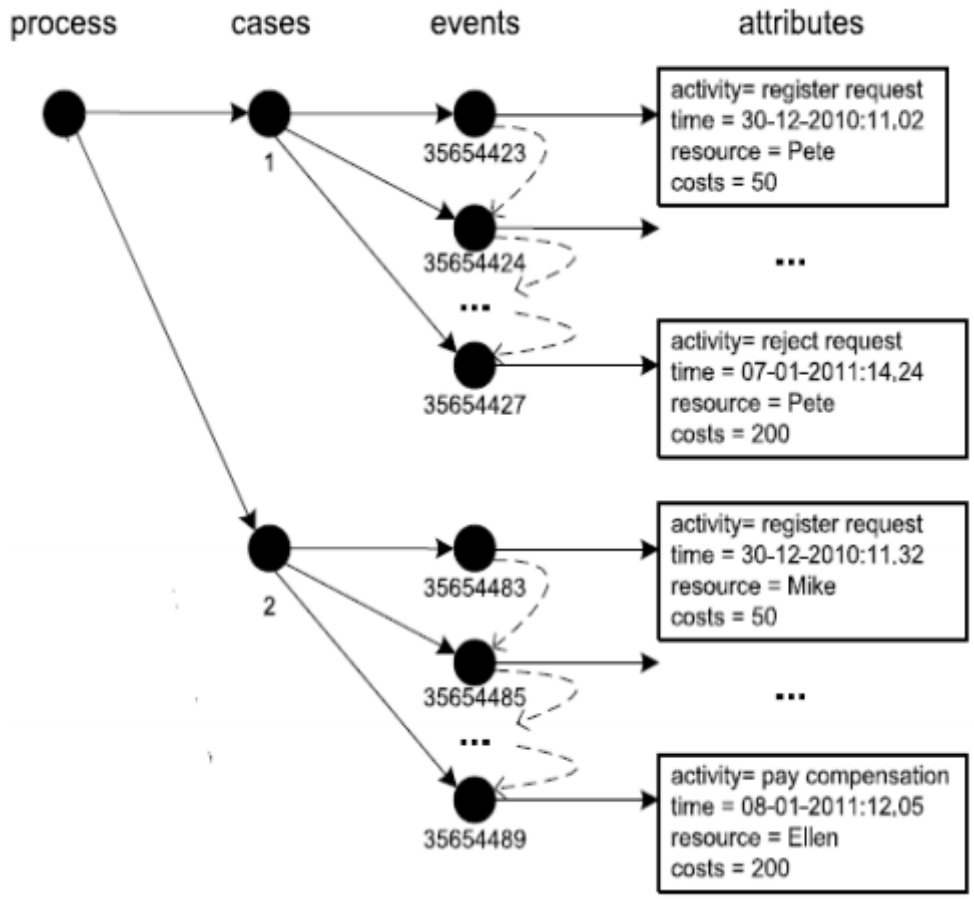
- *Επέκταση*: Υπάρχει ένα μοντέλο των προτέρων. Το μοντέλο αυτό επεκτείνεται με μια νέα πτυχή ή προοπτικές, δηλαδή, ο στόχος δεν είναι να ελέγχει τη συμμόρφωση, αλλά για να εμπλουτίσουν το μοντέλο. Ένα παράδειγμα είναι η επέκταση του μοντέλου διαδικασίας με δεδομένα απόδοσης, δηλαδή, περίπου ένα μοντέλο διαδικασίας των προτέρων δυναμικά σχολιασμένο με δεδομένα απόδοσης (π.χ. οι δυσκολίες εμφανίζονται με χρώμα τμήματα του μοντέλου διαδικασίας).

2.2.1 Event Logs



Εικόνα 1 L2M (Από Log σε Model), M2M (Από Model σε Model)

Το εναρκτήριο σημείο για την εξόρυξη διεργασιών είναι το **event log**. Το **event log** είναι ένα αρχείο όπου καταγράφουμε δραστηριότητες και την στιγμή που πραγματοποιήθηκε η εκάστοτε δραστηριότητα (**timestamp**). Κάθε φορά εκτελείται μια διεργασία συμβάν της εν λόγω διεργασίας (**process instance**). Κάθε **process instance** το καλούμε περίπτωση (**case**) ή ίχνος (**trace**). Κάθε **trace** περιγράφει τον κύκλο ζωής ενός συγκεκριμένου **case** (**process instance**) με βάση των δραστηριοτήτων που έχουν εκτελεστεί. Κάθε **case** έχει πολλές δραστηριότητες (καλά καθορισμένα βήματα σε μια διεργασία) οι οποίες είναι σε σειρά. Έτσι κάθε δραστηριότητα ανήκει σε ένα συγκεκριμένο **case** (**process instance**) αλλά η ίδια δραστηριότητα μπορεί να εκτελείται ξανά σε κάποιο άλλο **process instance**. Τα συμβάντα (**events**) που ανήκουν σε ένα **case** μπορούν να ληφθούν και σαν ‘’μια’’ εκτέλεση κάποιας διεργασίας. Η βασικότερη και σημαντικότερη πληροφορία που πρέπει να καταγραφεί σε ένα **event log** είναι ο αριθμός του κάθε **case**, το όνομα κάθε δραστηριότητας και το **timestamp**.



Εικόνα 2 Διαδικασία που περιέχει cases, όπου κάθε case περιέχει events

Τα **event log** μπορεί να αποθηκεύουν και επιπλέον πληροφορίες για τα γεγονότα (**events**). Στην πραγματικότητα όποτε υπάρχει η δυνατότητα, η τεχνικές εξόρυξης διεργασιών χρησιμοποιούν επιπλέον πληροφορίες, όπως η πηγή (πρόσωπο ή συσκευή) που εκτελεί η θέτει σε λειτουργία μια δραστηριότητα, επιπλέον δεδομένα που έχουν καταγραφεί με το **event** (για παράδειγμα την ηλικία ενός ασθενή, τις θερμίδες που κατανάλωσε κ.τ.λ.), το κόστος και πολλές άλλες πληροφορίες. Μπορούμε επίσης να καταγράψουμε περισσότερα **timestamps** που αφορούν κάποια δραστηριότητα (όχι μόνο τον χρόνο έναρξης αλλά και τον χρόνο περάτωσης). Τα γνωρίσματα (**attributes**) κάθε **trace** είναι πολύ σημαντικά για την ανάλυση των διεργασιών. Ένα **event** μπορεί επιπλέον να περιέχει συναλλαγματική πληροφορία (για παράδειγμα μπορεί να αναφέρεται σε κάποια από τις παρακάτω ενέργειες “**assign**”, “**start**”, “**complete**”, “**suspend**”, “**resume**”, “**abort**”). Για παράδειγμα για να μετρήσουμε την διάρκεια μιας δραστηριότητας είναι σημαντικό να ξέρουμε το εναρκτήριο **event** αλλά και το **event** ολοκλήρωσης της δραστηριότητας. Τα **event logs** έχουν τα παρακάτω υποχρεωτικά γνωρίσματα:

- Case id: Κάθε **event** αναφέρεται σε ένα **case** (**process instance**). Αν ένα **event** είναι σχετικό με πολλαπλά **cases**, πρέπει να επαναλαμβάνεται όταν δημιουργούνται τα **event logs**.

- Activity: Κάθε **event** πρέπει να σχετίζετε με ένα **activity**. Τα **events** αναφέρονται στα **activity instances** (συμβάντα δραστηριοτήτων) στο **process model**.
- Timestamp: Τα **events** μέσα σε ένα **case** πρέπει να είναι στην σειρά. Επιπροσθέτως τα **timestamps** δεν χρειάζονται απλά για να γνωρίζουμε την σειρά των δραστηριοτήτων αλλά και για να υπολογίζουμε την απόδοση.

Ακόμα έχουν τα παρακάτω επιπλέον γνωρίσματα:

- Resource: Το άτομο, η μηχανή ή το λογισμικό που εκτελεί το **event**.
- Type: Ο τύπος της ενέργειας σε ένα **event** (start, complete, suspend, resume κ.λ.π.)
- Costs: Τα κόστη που σχετίζονται με το εκάστοτε **event**.
- Customer: Πληροφορίες για το άτομο ή τον οργανισμό από τον οποίο ή για τον οποίο εκτελέστηκε το **event**.

Γνωρίζονται την πηγή (**resource**) μπορούμε να ανακαλύψουμε την οργανωτική δομή της επιχείρησης και να προσθέσουμε οργανωτική προοπτική σε ένα μοντέλο διεργασιών. Παρομοίως, πληροφορίες για τα **timestamps** και τις συχνότητες μπορούν να χρησιμοποιηθούν για να προσθέσουμε επιπλέον πληροφορίες για την απόδοση του μοντέλου. Από τα **timestamps** μπορούμε να υπολογίσουμε τον χρόνο μεταξύ των δραστηριοτήτων, την διάρκεια μιας δραστηριότητας και να ανακαλύψουμε πιθανά προβλήματα (σημεία συμφόρησης) αλλά και τον λόγο που προκαλεί το εν λόγω πρόβλημα. Στις παρακάτω εικόνες μπορούμε να δούμε παραδείγματα **event logs**.

Patient ID	Activity	Resource	Timestamp	Blood pressure
123	First consult	dr. Anna	2018-01-05 11:15	100 / 65
789	First consult	dr. Anna	2018-01-05 11:30	134 / 89
123	Blood test	Lab	2018-01-09 15:30	105 / 66
123	Physical test	dr. Ben	2018-01-09 16:30	102 / 64
123	X-ray scan	Team 1	2018-01-11 09:30	
789	X-ray scan	Team 1	2018-01-11 10:30	
123	Second consult	dr. Anna	2018-01-12 12:45	102 / 63
123	Surgery	dr. Charlie	2018-01-24 13:00	97 / 67
456	First consult	dr. Ben	2018-01-24 13:40	95 / 62
123	Final consult	dr. Anna	2018-01-27 10:20	100 / 65
789	Physical test	dr. Anna	2018-01-30 08:30	124 / 67

Εικόνα 3 Παράδειγμα event log με διαδικασίες που εκτελούνται σε ένα ιατρείο (1)

Case id	Event id	Properties			
		Timestamp	Activity	Resource	Cost
1	4798669	02/06/2014 14:00	First Visit	Pete	150
	4798670	07/06/2014 11:00	Surgery	Rose	55
	4798677	09/06/2014 17:00	Second Visit	Pete	150
	4798679	12/06/2014 12:15	Radiotherapy	Alfred	200
	4798680	14/06/2014 12:15	Chemotherapy	Michael	300
	4798685	17/06/2014 10:00	Evaluate	Pete	175
2	7777171	05/06/2014 09:15	First Visit	John	120
	7777195	13/06/2014 14:30	Surgery	Nick	610
	7777189	15/06/2014 15:45	Second Visit	John	120
	7777179	27/06/2014 11:45	Chemotherapy	Michael	300
	7777195	29/06/2014 14:30	Radiotherapy	Alfred	200
	7777185	08/07/2014 08:15	Evaluate	Pete	180
3	7932191	12/06/2014 14:45	First Visit	John	120
	7932192	23/06/2014 11:00	Surgery	Rose	55
	7932196	25/06/2014 08:00	Second Visit	John	120
	7932192	01/07/2014 16:00	Radiotherapy	Alfred	200
	7932193	07/07/2014 17:30	Chemotherapy	Michael	300
	7932196	21/07/2014 08:00	Evaluate	John	185

Εικόνα 4 Παράδειγμα event log με διαδικασίες που εκτελούνται σε ένα ιατρείο (2)

Κάθε γραμμή του **event log** ανταποκρίνεται σε ένα **event**. Το **event** “First Visit” αναφέρεται σε τρία **case** (1, 2, 3) και έχει επιπλέον γνωρίσματα: resource και cost.

2.2.2 Εξαγωγή Event log

Τα δεδομένα του **event log** δεν είναι πάντα σε μορφή που μπορεί να χρησιμοποιηθεί για **process mining**. Από την στιγμή που έχουμε πολλά δεδομένα αποθηκευμένα στις αποθήκες δεδομένων (**data storage**) του πληροφοριακού μας συστήματος, είναι συχνό φαινόμενο να κατασκευάζουμε από την αρχή το **event log** που θα χρησιμοποιήσουμε για την εξόρυξη διεργασιών. Τα απαραίτητα δεδομένα για να κατασκευάσουμε ένα **event log** είναι κρυμμένα μεταξύ άλλων επιχειρησιακών δεδομένων. Ας πάρουμε για παράδειγμα την δραστηριότητα ‘create order’ (δημιουργία παραγγελίας) σε ένα **ERP** σύστημα όπου τα δεδομένα για την ημερομηνία, τον χρόνο δημιουργίας της παραγγελίας και τον χρήστη που έκανε την παραγγελία μπορεί να είναι αποθηκευμένα στα δεδομένα της παραγγελίας αλλά να μην συμπεριλαμβάνονται στο **event log**.

Η εξαγωγή των πληροφοριών από τα επιχειρησιακά δεδομένα είναι μια χρονοβόρα διαδικασία και απαιτεί γνώσεις πάνω στο συγκεκριμένο πεδίο. Επιπλέον η μετατροπή του **event log** στην επιθυμητή μορφή δεν είναι πάντα ξεκάθαρη. Η απόφαση για το τι και το πώς πρέπει να εξάγουμε από την πηγή δεδομένων μας και να το συμπεριλάβουμε στο **event log** μας επηρεάζει τα αποτελέσματα της εξόρυξης

διεργασιών δραστικά κατά την φάση ανάλυσης. Τα περισσότερα έργα επικεντρώνονται σε ένα συγκεκριμένο μέρος της διεργασίας έτσι ώστε να δοθεί ιδιαίτερη προσοχή στα **events** που συμβαίνουν στο συγκεκριμένο μέρος της διεργασίας. Το παραπάνω απαιτεί επίσης γνώση της επιχειρηματικής διεργασίας και γνώση των **events** που πρέπει να αναζητήσουμε στα δεδομένα μας. Όλα τα παραπάνω κάνουν την προετοιμασία ενός **event log** για **process mining** ιδιαίτερα χρονοβόρα και πολύπλοκη.

2.2.3 Event log standards

Όπως είδαμε στα παραπάνω παραδείγματα ένα **event log** μπορεί να είναι ένας πίνακας, για παράδειγμα ένα **csv** αρχείο. Υπάρχουν όμως πολλά ακόμα **formats** για τα **event log** που μπορούμε να χρησιμοποιήσουμε στην εξόρυξη διεργασιών.

MXML format

Το **MXML** βγήκε το 2003. Χρησιμοποιώντας **MXML** μπορούμε να αποθηκεύσουμε **event logs** χρησιμοποιώντας **XML** σύνταξη. Το **MXML** έχει βασική σημειογραφία για να αποθηκεύει **timestamps**, **resources** και τύπους ενεργειών (**transaction types**). Επιπλέον μπορούμε να προσθέσουμε αυθαίρετα στοιχεία δεδομένων στα **events** και στα **cases**. Αυτό οδήγησε στην επέκταση του **MXML** σε διάφορες εκδόσεις όπου τα διάφορα γνωρίσματα ερμηνεύονται με συγκεκριμένο τρόπο. Για παράδειγμα το **SA-MXML** είναι μια σημασιολογικά σχολιασμένη έκδοση του **MXML** που χρησιμοποιείται από το **ProM**. Το **SA-MXML** ενσωματώνει αναφορές μεταξύ των στοιχείων στα **logs** και των εννοιών στις οντολογίες. Για παράδειγμα μια πηγή (**resource**) μπορεί να έχει μια αναφορά σε ένα **concept** σε μια οντολογία, που περιγράφει έναν ρόλο ιεραρχιών, οργανωτικών οντοτήτων και θέσεων. Για να κατανοήσουμε αυτές τις σημασιολογικές περιγραφές, τα υπάρχοντα **XML** στοιχεία ερμηνεύονται με καινούργιο τρόπο. Επιπλέον επεκτάσεις (**extensions**) ανακαλύφθηκαν με παρόμοιο τρόπο. Παρόλο που αυτή η προσέγγιση δούλεψε αρκετά καλά στην πράξη, τα διάφορα **extensions** αποκάλυψαν ελλείψεις του **MXML** μορφοτύπου. Παρακάτω βλέπουμε ένα παράδειγμα ενός **XML** αρχείου.

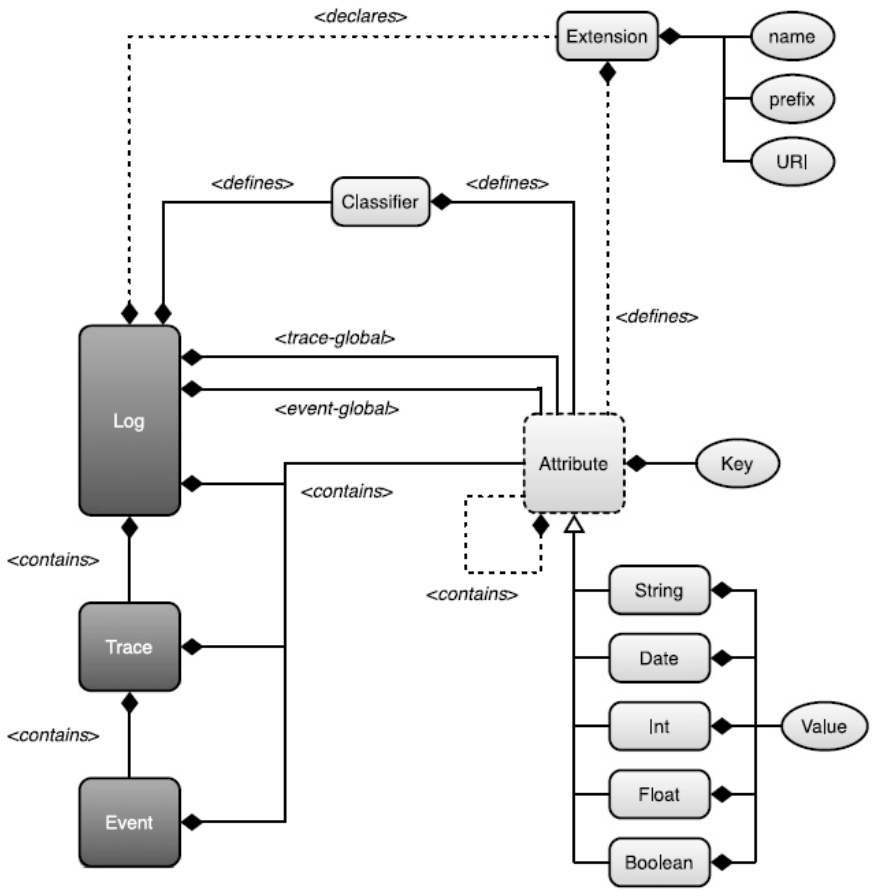
```
<ProcessInstance id="Order 1" description="instance with Order 1">
  <Data>
    <Attribute name="totalValue">2142.38</Attribute>
  </Data>
  <AuditTrailEntry>
    <WorkflowModelElement>Create</WorkflowModelElement>
    <EventType>complete</EventType>
    <originator>Wil</originator>
    <timestamp>2009-01-03T15:30:00.000+01:00</timestamp>
    <Data>
      <Attribute name="currentValue">2142.38</Attribute>
      <Attribute name="requestedBy">Eric</Attribute>
      <Attribute name="supplier">Fluxi Inc.</Attribute>
      <Attribute name="expectedDelivery">2009-01-12T12:00:00.000+01:00</Attribute>
    </Data>
  </AuditTrailEntry>
</ProcessInstance>
```

Εικόνα 5 XML αρχείο που περιγράφει ένα order process

XES Standard - eXtensible Event Stream

Το **XES** είναι ο διάδοχος του **MXML**. Τον Σεπτέμβριο του 2010, η συγκεκριμένη μορφοποίηση υιοθετήθηκε από το **IEEE Task Force** και καθιερώθηκε για την εξόρυξη διεργασιών. Αυτή η νέα μορφοποίηση αναπτύχθηκε από την ερευνητική ομάδα **Architecture of Information Systems** στο **Eindhoven University of Technology** σε συνεργασία με την **Fluxicon**. Αυτό το νέο **format** για τα **event log** επίλυσε κάποια θέματα που δημιουργήθηκαν από την χρήση του **MXML** ως προς την καταγραφή της εκτέλεσης διεργασιών και επέτρεψε μεγαλύτερη ευελιξία. Είναι ακόμα δυνατό να δημιουργήσουμε **event logs** σε **MXML** μορφοποίηση κάτω υπό ορισμένες συνθήκες.

Σε σύγκριση με το **MXML**, το **XES** είναι πολύ λιγότερο περιοριστικό και περισσότερο επεκτάσιμο. Το **XES** ορίζει μια γραμματική για μια γλώσσα βασισμένη σε ετικέτες, της οποίας στόχος είναι να παρέχει στους σχεδιαστές πληροφοριακών συστημάτων μια ενοποιημένη και επεκτάσιμη μεθοδολογία για την καταγραφή-σύλληψη τριών συμπεριφορών συστημάτων, με βάση αρχεία καταγραφής και ροές γεγονότων. Η συγκεκριμένη μορφοποίηση περιλαμβάνει ένα ‘’ **XML 4 Schema**’’ που περιγράφει την δομή ενός **XES event log/stream** και ένα ‘’**XML Schema**” που περιγράφει την δομή μιας επέκτασης όπως αυτής ενός **log/stream**. Επιπλέον το **XES format** περιλαμβάνει μια βασική συλλογή επεκτάσεων που καλούνται πρωτότυπα ‘’**XES extension**’’ και παρέχουν σημασιολογική εξήγηση σε κάποια γνωρίσματα που καταγράφονται στο αρχείο καταγραφής γεγονότων.



Εικόνα 6 UML διάγραμμα για το XES

log	Ένα έγγραφο XES που περιέχει μια καταγραφή (**log**) με οποιαδήποτε αριθμό ιχνών (**traces**)
trace	Κάθε ίχνος περιγράφει μια ακολουθία γεγονότων που ανταποκρίνονται σε μια περίπτωση (**case**)
event	Ένα γεγονός αναπαριστά μια δραστηριότητα που έχει παρατηρηθεί. Άμα το γεγονός συμβεί σε κάποιο ίχνος, είναι ξεκάθαρο σε ποια περίπτωση ανήκει αυτό το γεγονός. Άμα το γεγονός δεν συμβεί μέσα σε κάποιο ίχνος, τότε χρειάζεται να βρεθεί τρόπος συσχέτισης γεγονότων με περιπτώσεις. Για αυτό χρησιμοποιούμε συνδυασμό ταξινομητή ίχνους και ταξινομητή γεγονότος
attributes	Πληροφορίες για οποιοδήποτε στοιχείο (**log, trace**, **event**) αποθηκεύονται στα γνωρίσματα (**attributes**). Τα γνωρίσματα μπορεί να είναι και εμφωλευμένα. Οι βασικοί τύποι γνωρισμάτων είναι **String**, **Date**, **Int**, **Float** και **Boolean.**
extension	Οι επεκτάσεις δίνουν σημασία σε συγκεκριμένα **attributes.** Βασικές επεκτάσεις είναι: **-** concept (για ονοματολογία) **-** lifecycle (για συναλλαγματικές ιδιότητες) **-** org (για οργανωτική προοπτική) **-** time (για τα timestamps) **-** semantic (για αναφορές σε οντολογίες) Πιθανές χρήσεις των επεκτάσεων είναι: **-** Να παρουσιάσουνε ένα σετ κατανοητών γνωρισμάτων, τα οποία είναι απαραίτητα για να γίνει ανάλυση του **log**. **-** Να ορίσουμε γενικά κατανοητά γνωρίσματα για την εφαρμογή σε ένα συγκεκριμένο τομέα (για παράδειγμα, γνωρίσματα ιατρικής φύσεως για διεργασίες σε ένα νοσοκομείο) ή για την υποστήριξη χαρακτηριστικών και απαιτήσεων κάποιας εφαρμογής

	Μια επέκταση έχει: - Περιγραφικό όνομα - Πρόθεμα όλων των γνωρισμάτων που ορίζονται από την επέκταση - URI που δείχνει τον ορισμό της επέκτασης
classifier	Υπάρχουν δύο είδη ταξινομητών: - ταξινομητής γεγονότων - ταξινομητής ίχνους Ο ταξινομητής αναθέτει σε κάθε γεγονός ή σε κάθε ίχνος μια ταυτότητα, η οποία το κάνει συγκρίσιμο με τα άλλα γεγονότα ή ίχνη. Κάθε ταξινομητής ορίζεται από μια ταξινομημένη λίστα με κλειδιά γνωρισμάτων (**attribute keys**). Κάθε ζευγάρι γεγονότων ή ιχνών που έχουν όμοιες τιμές θεωρούνται ίσα από τον ταξινομητή. Αυτά τα γνωρίσματα είναι υποχρεωτικά γνωρίσματα γεγονότων

case	event	completeTime
trace 0	register application	2013-04-16 10:08:02
trace 0	check credit	2013-04-16 10:16:03
trace 0	calculate capacity	2013-04-16 10:16:28
trace 0	check system	2013-04-16 10:20:25
trace 0	accept	2013-04-16 10:21:30
trace 0	send decision e-mail	2013-04-16 10:26:53
trace 1	register application	2013-04-16 10:10:02
trace 1	check credit	2013-04-16 10:16:07
trace 1	calculate capacity	2013-04-16 10:16:21
trace 1	check system	2013-04-16 10:20:11
trace 1	accept	2013-04-16 10:24:16
trace 1	send decision e-mail	2013-04-16 10:29:58
trace 2	register application	2013-04-16 10:15:42
trace 2	check credit	2013-04-16 10:22:41
trace 2	calculate capacity	2013-04-16 10:23:37
trace 2	check system	2013-04-16 10:29:18
trace 2	accept	2013-04-16 10:37:54
trace 2	send decision e-mail	2013-04-16 10:44:03
trace 3	register application	2013-04-16 10:19:56
trace 3	check credit	2013-04-16 10:23:07
trace 3	calculate capacity	2013-04-16 10:28:41
trace 3	check system	2013-04-16 10:33:33
trace 3	accept	2013-04-16 10:42:51
trace 3	send decision e-mail	2013-04-16 10:50:47
trace 4	register application	2013-04-16 10:28:43
trace 4	check credit	2013-04-16 10:32:08
trace 4	calculate capacity	2013-04-16 10:37:11
trace 4	check system	2013-04-16 10:40:47
trace 4	accept	2013-04-16 10:45:33
trace 4	send decision e-mail	2013-04-16 10:48:13
trace 5	register application	2013-04-16 10:33:28
trace 5	check credit	2013-04-16 10:36:29
trace 5	calculate capacity	2013-04-16 10:38:52
trace 5	check system	2013-04-16 10:43:58
trace 5	accept	2013-04-16 10:44:26
trace 5	send decision e-mail	2013-04-16 10:45:09
trace 6	register application	2013-04-16 10:35:07
trace 6	check credit	2013-04-16 10:42:11
trace 6	calculate capacity	2013-04-16 10:42:47
trace 6	check system	2013-04-16 10:51:33
trace 6	reject	2013-04-16 10:54:35
trace 6	send decision e-mail	2013-04-16 11:04:09

Εικόνα 7 Event log. Παρατηρούμε πως κάθε trace έχει πολλά events

```
<?xml version="1.0" encoding="UTF-8" ?>
<log xes.version="1.0" xes.features="nested-attributes" openxes.version="1.0RC7" xmlns="http://www.xes-standard.org/">
    <extension name="Metadata_Time" prefix="meta_time" uri="http://www.xes-standard.org/meta_time.xesext"/>
    <extension name="Lifecycle" prefix="lifecycle" uri="http://www.xes-standard.org/lifecycle.xesext"/>
    <extension name="Metadata_Lifecycle" prefix="meta_life" uri="http://www.xes-standard.org/meta_life.xesext"/>
    <extension name="Time" prefix="time" uri="http://www.xes-standard.org/time.xesext"/>
    <extension name="Metadata_Concept" prefix="meta_concept" uri="http://www.xes-standard.org/meta_concept.xesext"/>
    <extension name="Completeness metadata" prefix="meta_completeness" uri="http://www.xes-standard.org/meta_completeness.xesext"/>
    <extension name="3TU metadata" prefix="meta_3TU" uri="http://www.xes-standard.org/meta_3TU.xesext
    <extension name="Concept" prefix="concept" uri="http://www.xes-standard.org/concept.xesext"/>
    <extension name="General metadata" prefix="meta_general" uri="http://www.xes-standard.org/meta_ge
    <global scope="trace">
        <string key="concept:name" value="UNKNOWN"/>
    </global>
    <global scope="event">
        <string key="lifecycle:transition" value="UNKNOWN"/>
        <string key="concept:name" value="UNKNOWN"/>
        <date key="time:timestamp" value="1970-01-01T01:00:00.000+01:00"/>
    </global>
    <classifier name="MXML Legacy Classifier" keys="concept:name lifecycle:transition"/>
    <float key="meta_time:duration_max" value="2433.443"/>
    <string key="meta_3TU:language" value="eng"/>
```

επεκτάσεις

Κάθε trace έχει ένα γνώρισμα concept:name

Κάθε γεγονός έχει τα γνωρίσματα concept:name, lifecycle:transition και time:timestamp

Όνομα ταξινομητή και κλειδιά γνωρισμάτων (concept:name, lifecycle:transition)

Εικόνα 8 Event log σε μορφή XML

```
<trace>
    <string key="concept:name" value="trace 0"/>
    <event>
        <date key="time:timestamp" value="2013-04-16T10:08:01.821+02:00"/>
        <string key="concept:name" value="register application"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <date key="time:timestamp" value="2013-04-16T10:16:02.889+02:00"/>
        <string key="concept:name" value="check credit"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <date key="time:timestamp" value="2013-04-16T10:16:27.858+02:00"/>
        <string key="concept:name" value="calculate capacity"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <date key="time:timestamp" value="2013-04-16T10:20:25.117+02:00"/>
        <string key="concept:name" value="check system"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <date key="time:timestamp" value="2013-04-16T10:21:29.939+02:00"/>
        <string key="concept:name" value="accept"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <string key="lifecycle:transition" value="complete"/>
        <string key="concept:name" value="send decision e-mail"/>
        <date key="time:timestamp" value="2013-04-16T10:26:53.166+02:00"/>
    </event>
</trace>
```

γνωρίσματα: concept:name, time:timestamp, lifecycle:transition

Εικόνα 9 Trace και τα events που περιέχει

2.3.1 Γενικοί ορισμοί

Κάθε συσκευή όπως και τα περισσότερα πληροφοριακά συστήματα που βρίσκονται πλέον σε ένα αυτοματοποιημένο και διασυνδεδεμένο περιβάλλον, εκτελεί και καταγράφει γεγονότα, με στόχο τη διατήρηση ιστορικού ή και την παρακολούθηση του συστήματος στο οποίο συμμετέχει. Τα γεγονότα αυτά αφορούν συνήθως τις διεργασίες ενός οργανισμού, όπως για παράδειγμα, την ανάληψη μετρητών από ένα μηχάνημα ATM μιας τράπεζας, τη ρύθμιση ενός ιατρικού μηχανήματος σε μια κλινική, την ηλεκτρονική αίτηση στο πληροφοριακό σύστημα μιας δημόσιας υπηρεσίας ή την ολοκλήρωση μιας παραγγελίας σε ένα ηλεκτρονικό κατάστημα. Στο άρθρο [14] μάλιστα περιγράφεται η έννοια του Διαδικτύου Γεγονότων (Internet of Events) ως το σύνολο των γεγονότων που παράγονται από: - Διαδίκτυο Περιεχομένου (Internet of Content): Το σύνολο της πληροφορίας που δημιουργούν οι άνθρωποι για την αύξηση της γνώσης σε ένα γνωστικό πεδίο. Σε αυτό περιέχονται τα ιστολόγια, τα άρθρα, οι

διαδικτυακές εγκυκλοπαίδειες κ.λπ. - Διαδίκτυο Ανθρώπων (Internet of People): Το σύνολο των δεδομένων που αφορούν την κοινωνική αλληλεπίδραση, όπως η ηλεκτρονική αλληλογραφία, το Facebook, τα forum κ.λπ. - Διαδίκτυο Αντικειμένων (Internet of Things): Το σύνολο των φυσικών αντικειμένων τα οποία είναι συνδεδεμένα στο δίκτυο, έχουν μοναδικό αναγνωριστικό και παρουσία σε μια δομή που προσομοιάζει το Διαδίκτυο. Παραδείγματα τέτοιων αντικειμένων είναι οι συσκευές RFID, NFC κ.ά. - Διαδίκτυο Τοποθεσιών (Internet of Locations): Το σύνολο των δεδομένων που έχουν χωρική διάσταση. Κυρίως χάρη στις κινητές συσκευές πολλά πλέον γεγονότα διαθέτουν γεω-χωρικά χαρακτηριστικά. Η ευρέως διαδεδομένη πλέον καταγραφή γεγονότων από κάθε ηλεκτρονική συσκευή και λογισμικό καθιστά εφικτή την ανάλυση των καταγεγραμμένων δεδομένων, τη διεξαγωγή και την αξιοποίηση της πληροφορίας που εμπεριέχουν. Η Εξόρυξη Διεργασιών στοχεύει στην εκμετάλλευση αυτής της πληροφορίας μέσω διαφόρων τεχνικών και αλγορίθμων προκειμένου να προσφέρει επιπλέον γνώση στην επιχείρηση σχετικά με τις διεργασίες που αυτή εκτελεί. Με τη βοήθεια της Εξόρυξης Δεδομένων για παράδειγμα μπορεί να ανακαλυφθεί το μοντέλο διεργασιών σε μια επιχείρηση στην οποία δεν έχει γίνει σαφής καταγραφή ή σχεδίαση των διεργασιών εκ των προτέρων. Είναι δυνατόν να αναγνωριστούν σημεία συμφόρησης στο μοντέλο διεργασιών και να τροποποιηθούν ανάλογα οι διεργασίες που εκτελούνται ή να αξιολογηθεί το κατά πόσο οι διαδικασίες που ακολουθούνται ικανοποιούν πρότυπα και περιορισμούς που απαιτούνται. Η πρόταση αυτή αποτυπώνεται από [14] οι οποίοι ορίζουν ως στόχο της Εξόρυξης Επιχειρηματικών Διεργασιών την ανακάλυψη, παρακολούθηση και βελτιστοποίηση των πραγματικών διεργασιών της επιχείρησης μέσω της διεξαγωγής πληροφορίας από τα αρχεία καταγραφής γεγονότων των πληροφοριακών συστημάτων της επιχείρησης. Το σημείο αναφοράς για την Εξόρυξη Διεργασιών είναι ένα αρχείο καταγραφής γεγονότων (event log). Στο αρχείο αυτό καταγράφονται αποκλειστικά και με συγκεκριμένη δομή τα δεδομένα των γεγονότων που εκτελούνται. Όλες οι τεχνικές Εξόρυξης Διεργασιών θεωρούν δυνατή την ακολουθιακή καταγραφή των γεγονότων έτσι ώστε κάθε γεγονός να αναφέρεται σε μια δραστηριότητα και να συνδέεται με μια δεδομένη περίπτωση. Ως δραστηριότητα (activity), σύμφωνα με τους [15] θεωρούμε ένα καλώς ορισμένο βήμα μιας διεργασίας. Κατά συνέπεια, ως γεγονός μπορούμε να ορίσουμε την εκτέλεση μιας δραστηριότητας σε μία δεδομένη χρονική στιγμή και με δεδομένες παραμέτρους. Με τον όρο περίπτωση (case) καλούμε ένα στιγμιότυπο (μια εκτέλεση) της διεργασίας αυτής, η οποία περιγράφεται από ένα ίχνος (trace) από γεγονότα, δηλαδή μια ακολουθία από δραστηριότητες. Τα αρχεία καταγραφής γεγονότων που κατασκευάζονται από το εκάστοτε σύστημα μπορούν να περιέχουν επιπλέον πληροφορία σχετικά με τα γεγονότα όπως τον πόρο (π.χ. τον εργαζόμενο, το μηχάνημα ή το λογισμικό) που εκτέλεσε το γεγονός, τον χρόνο της εκτέλεσης του γεγονότος και άλλα σημαντικά χαρακτηριστικά.

2.3.2 Ταξινόμηση Τεχνικών Εξόρυξης Διεργασιών

Σε όλη την έκταση της βιβλιογραφίας που σχετίζεται με την Εξόρυξη Διεργασιών, όπως στα άρθρα των [15], [16] και [17], οι τεχνικές Εξόρυξης Διεργασιών μπορούν να ταξινομηθούν σε τρεις βασικές κατηγορίες, στις τεχνικές ανακάλυψης, τεχνικές ελέγχου συμμόρφωσης και τεχνικές βελτιστοποίησης. Όλοι οι οργανισμοί και οι επιχειρήσεις εκτελούν κάποιες διαδικασίες προκειμένου να διαχειριστούν διαφορετικές απαιτήσεις και να εκπληρώσουν τις υποχρεώσεις τους. Σε κάποιες

περιπτώσεις οι διαδικασίες αυτές ορίζονται από το πληροφοριακό σύστημα, που χρησιμοποιεί ο οργανισμός, όμως στις περισσότερες περιπτώσεις οι διαδικασίες αυτές δεν έχουν επισήμως οριστεί και δεν έχουν απαραίτητα καταγραφεί λεπτομερώς. Ακόμα και στις περιπτώσεις στις οποίες οι διαδικασίες που ακολουθεί η επιχείρηση έχουν καταγραφεί, ο τρόπος με τον οποίο ολοκληρώνεται μια διαδικασία μπορεί να διαφέρει από εκείνον που έχει αρχικά οριστεί. Οι τεχνικές της Εξόρυξης Διεργασιών που ανήκουν στην κατηγορία της ανακάλυψης ασχολούνται με την αυτοματοποιημένη κατασκευή του μοντέλου διεργασιών που προκύπτει από τα γεγονότα που καταγράφονται στο αρχείο καταγραφής γεγονότων. Ως μοντέλο διεργασιών ορίζεται το μοντέλο με το οποίο μπορούν να αναπαραχθούν όλα ή τα περισσότερα από τα ίχνη που εμπεριέχονται στο **event log**. Η είσοδος που δέχονται οι τεχνικές αυτές είναι το **event log** ενώ το αποτέλεσμα της εξόδου είναι το παραγόμενο μοντέλο διεργασιών. Ένα παράδειγμα αξιοποίησης του παραγόμενου μοντέλου διεργασιών που προκύπτει μέσω του **event log** είναι η χρήση του ως βάση συζήτησης για τα προβλήματα που εντοπίζονται στον τρόπο εκτέλεσης των διεργασιών του οργανισμού, εφόσον διευκολύνει την παρουσίαση των διεργασιών που εκτελούνται. Επίσης μπορεί να βοηθήσει στη βελτίωση ενός ήδη ορισμένου μοντέλου διεργασιών ώστε αυτό να προσεγγίζει περισσότερο την πραγματική διεργασία που εκτελείται.

Για την αναπαράσταση ενός μοντέλου διεργασιών μπορούν να χρησιμοποιηθούν διάφορες σημειογραφίες όπως **Petri nets**, **BPMN**, διαγράμματα **UML** κ.ά. Μια βασική σημειογραφία που χρησιμοποιείται για την αναπαράσταση του μοντέλου διεργασιών είναι τα συστήματα μετάβασης [18]. Ένα σύστημα μετάβασης αποτελείται από καταστάσεις (states), δραστηριότητες (activities) και μια σχέση μετάβασης μεταξύ των καταστάσεων οι οποίες σημειώνονται μέσω των δραστηριοτήτων. Οι καταστάσεις συμβολίζονται με μαύρους κύκλους και οι μεταβάσεις με τόξα. Μία μετάβαση συνδέει δύο καταστάσεις και ονοματοδοτείται από τη δραστηριότητα την οποία αναπαριστά, ενώ δύο ή περισσότερες μεταβάσεις μπορούν να έχουν το ίδιο όνομα.

Τα Petri nets βρίσκουν μεγάλη εφαρμογή στις τεχνικές ανακάλυψης του μοντέλου διεργασιών μιας και οι πιο διαδεδομένοι αλγόριθμοι της κατηγορίας αυτής, όπως ο α-αλγόριθμος (**Alpha Miner**), παράγουν μοντέλα με αυτή τη σημειογραφία. Η επιλογή των Petri nets γίνεται λόγω της δυνατότητας της μοντελοποίησης παράλληλων δραστηριοτήτων αλλά και της επανάληψης που επιτρέπουν. Ένα Petri net [19], [20] είναι ένα κατευθυνόμενο διμερές γράφημα το οποίο αποτελείται από δύο τύπους κόμβων, τις θέσεις και τις μεταβάσεις. Οι κόμβοι συνδέονται με κατευθυνόμενα τόξα ενώ δεν επιτρέπεται η σύνδεση δύο κόμβων ίδιου τύπου. Οι κόμβοι-θέσεις συμβολίζονται με κύκλους και οι μεταβάσεις συμβολίζονται με ορθογώνια παραλληλόγραμμα. Η δομή του γραφήματος είναι στατική, όμως με χρήση του κανόνα πυροδότησης, κατά μήκος του γραφήματος μπορούν να κινούνται τεκμήρια (tokens). Ο αυστηρός ορισμός είναι:

Ένα Petri net είναι μια πλειάδα (P, T, F) όπου:

- P είναι ένα πεπερασμένο σύνολο θέσεων
- T είναι ένα πεπερασμένο σύνολο μεταβάσεων τέτοιο ώστε $(P \cap T) = \emptyset$
- $F \subseteq (P \times T) \cup (P \times T)$ είναι ένα σύνολο τόξων, το οποίο καλείται σχέσεις ροής

Σύμφωνα με τον ορισμό αυτό:

- θέση εισόδου μιας μετάβασης t καλείται μια θέση p αν και μόνο αν υπάρχει ένα κατευθυνόμενο τόξο από την p στην t
- θέση εξόδου μιας μετάβασης t καλείται μια θέση p αν και μόνο αν υπάρχει ένα κατευθυνόμενο τόξο από την t στην p

Κάθε θέση περιέχει μηδέν ή περισσότερα τεκμήρια (tokens) τα οποία συμβολίζονται με μαύρες τελείες στη θέση αυτή. Η κατάσταση (state) $M \in P \rightarrow \mathbb{N}$ ενός Petri net δηλώνει την κατανομή των tokens στις διάφορες θέσεις του γραφήματος με τη μορφή ενός ακέραιου αριθμού. Ο αριθμός των tokens και κατά συνέπεια η κατάσταση του γραφήματος μεταβάλλονται κατά την εκτέλεση του συστήματος. Οι μεταβάσεις αλλάζουν την κατάσταση ενός Petri net σύμφωνα με τους κανόνες πυροδότησης:

1. Μια μετάβαση t είναι ενεργοποιημένη αν και μόνο αν κάθε θέση εισόδου περιέχει τουλάχιστον ένα token.

2. Μια ενεργοποιημένη μετάβαση t πυροδοτείται όταν καταναλώνει ένα token από κάθε θέση εισόδου της και παράγει ένα token για κάθε θέση εξόδου.

Μια υποκατηγορία των Petri-nets που χρησιμοποιείται συχνά στη μοντελοποίηση των διεργασιών είναι τα Δίκτυα Ροής Εργασιών (Workflow-nets, εν συντομία WF-net). Ένα WF-net είναι ένα Petri net με μια θέση έναρξης (source) από την οποία αποκλειστικά ξεκινά η διεργασία και μια θέση τερματισμού (sink) στην οποία αποκλειστικά η διεργασία ολοκληρώνεται. Όλοι οι ενδιάμεσοι κόμβοι βρίσκονται σε κάποιο μονοπάτι από την έναρξη προς τον τερματισμό. Τα **WF-nets** φαίνονται κατάλληλα για τη μοντελοποίηση των διεργασιών γιατί μπορούν να περιγράψουν τον κύκλο ζωής κάθε τύπου περιπτώσεων. Το μοντέλο διεργασιών αρχικοποιείται μία φορά για κάθε περίπτωση, κάθε στιγμιότυπό του έχει μία καλώς ορισμένη έναρξη και λήξη ενώ οι ενδιάμεσες δραστηριότητες ακολουθούν μία προκαθορισμένη διαδικασία.

Τις σημειογραφίες αυτές αξιοποιούν πολλοί από τους αλγορίθμους που ανήκουν στην κατηγορία ανακάλυψης μοντέλου διεργασιών. Όπως γίνεται αντιληπτό από τη βιβλιογραφία, η κατηγορία της ανακάλυψης έχει κερδίσει το ενδιαφέρον της επιστημονικής κοινότητας έχοντας ως συνέπεια τη μεγάλη ανάπτυξη αλγορίθμων και τεχνικών. Παρακάτω συγκεντρώνουμε τις κυριότερες από τις διαφορετικές προσεγγίσεις τις οποίες ακολουθούν οι αλγόριθμοι αυτοί [16],[20]:

- **Οικογένεια α-αλγορίθμων:** ο α-αλγόριθμος ήταν από τους πρώτους αλγορίθμους της κατηγορίας που απέδωσε επαρκώς το φαινόμενο των σύγχρονων δραστηριοτήτων. Πολλές επεκτάσεις και παραλλαγές του αλγορίθμου αυτού έχουν προταθεί προκειμένου να ξεπεραστούν οι αδυναμίες του. Είσοδος του αλγορίθμου στην αρχική του μορφή είναι ένα αρχείο καταγραφής γεγονότων που περιέχει δραστηριότητες από ένα δεδομένο σύνολο δραστηριοτήτων. Ο αλγόριθμος εξετάζει τις αιτιολογικές εξαρτήσεις των δραστηριοτήτων του **event log** και βάσει αυτών κατασκευάζει το κατάλληλο **WF-net**.
- **Βάσει περιοχής:** Οι αλγόριθμοι αυτής της κατηγορίας διαχωρίζονται σε αλγορίθμους περιοχής βάσει κατάστασης και αλγορίθμους περιοχής βάσει γλώσσας. Οι αλγόριθμοι περιοχής βάσει κατάστασης ξεκινούν από ένα σύστημα

μετάβασης το οποίο έχει δημιουργηθεί από το **event log** και συνθέτουν από αυτό ένα **Petri net**, το οποίο μπορεί να χρησιμοποιηθεί για την κατασκευή του μοντέλου διεργασιών. Η έκταση ενός συστήματος μετάβασης είναι πολύ μεγαλύτερη από εκείνη του δυικού του **Petri net** εξαιτίας της έλλειψης της αναπαράστασης σύγχρονων δραστηριοτήτων. Οι αλγόριθμοι της κατηγορίας αυτής βασίζονται στον ορισμό της περιοχής και στοχεύουν στην ανακάλυψη περιοχών που αντιστοιχούν σε θέσεις του δυικού **Petri net**. Ως περιοχή [21] ορίζεται ένα σύνολο καταστάσεων του συστήματος μετάβασης στο οποίο όλες οι μεταβάσεις με το ίδιο όνομα έχουν την ίδια σχέση, δηλαδή είτε εισέρχονται σε αυτό το σύνολο, είτε εξέρχονται είτε δεν διασταυρώνονται με το σύνολο αυτό. Οι αλγόριθμοι περιοχής βάσει κατάστασης συνήθως εμφανίζουν προβλήματα όταν στο **event log** εμφανίζεται θόρυβος ή το αρχείο είναι ατελές. Θόρυβος (**noise**) [20] περιέχεται σε ένα αρχείο καταγραφής γεγονότων όταν το αρχείο περιέχει σπάνια συμπεριφορά η οποία εμφανίζει μικρή συχνότητα και δεν αντιπροσωπεύει την τυπική συμπεριφορά της διεργασίας. Ένα αρχείο καταγραφής γεγονότων είναι ατελές (**incomplete**) όταν περιέχει τόσο λίγα γεγονότα έτσι ώστε η διαδικασία ανακάλυψης του ελέγχου ροής καθίσταται αδύνατη. Οι αλγόριθμοι περιοχής βάσει γλώσσας ακολουθούν μία αντίστοιχη λογική. Η κύρια διαφοροποίησή τους εντοπίζεται στην είσοδο που δέχονται η οποία είναι κάποιο είδος “γλώσσας” και όχι ένα σύστημα μετάβασης που έχει προκύψει από το **event log**, προκείμενου να κατασκευάσουν το ζητούμενο **Petri net**. Τα μειονεκτήματα των αλγορίθμων περιοχής βάσει γλώσσας είναι ότι η γραμμική ανισότητα του συστήματος επιδέχεται πολλές λύσεις εκ των οποίων ελάχιστες αντιστοιχούν σε λογικές θέσεις του **Petri net** καθώς και το γεγονός ότι το παραγόμενο **Petri net** δεν μπορεί να αναπαράγει συμπεριφορά που δεν εμπεριέχεται στο **event log**.

- **Ευρετικοί:** Οι αλγόριθμοι αυτοί χρησιμοποιούν μια αναπαράσταση που προσομοιάζει με αιτιολογικό δίκτυο. Σε αντίθεση με τους αλγόριθμους της πρώτης κατηγορίας κατά την κατασκευή του μοντέλου λαμβάνουν υπόψιν τη συχνότητα των γεγονότων και των ακολουθιών τους και βασίζονται στην υπόθεση ότι τα μονοπάτια με τη μικρότερη συχνότητα δεν πρέπει να ενσωματώνονται στο μοντέλο διεργασιών.
- **Γενετικοί:** Οι ευρετικοί, οι ασαφείς και οι α-αλγόριθμοι παρέχουν ένα μοντέλο με έναν ευθύ και ντετερμινιστικό τρόπο. Οι γενετικοί αλγόριθμοι βασίζονται σε εξελικτικές προσεγγίσεις οι οποίες επιχειρούν με μια επαναληπτική διαδικασία να πλησιάσουν τη διαδικασία της φυσικής εξέλιξης. Όπως κάθε γενετικός αλγόριθμος, οι αλγόριθμοι της κατηγορίας αυτής αποτελούνται από τέσσερα βήματα την αρχικοποίηση, την επιλογή, την αναπαραγωγή και τον τερματισμό. Ο πληθυσμός αποτελείται από τα πιθανά μοντέλα διεργασιών τα οποία δημιουργούνται με τυχαίο τρόπο κατά το βήμα της αρχικοποίησης, με δραστηριότητες που εμπεριέχονται στο **event log**. Η επιλογή ενός μοντέλου από τον πληθυσμό γίνεται με κριτήριο την καταλληλότητα (**fitness**) του μοντέλου. Στο βήμα της αναπαραγωγής χρησιμοποιούνται οι γενετικοί τελεστές της διασταύρωσης και της μετάλλαξης ώστε να δημιουργηθεί μία νέα γενιά. Η διαδικασία επαναλαμβάνεται μέχρι να βρεθεί μία λύση που να ικανοποιεί τη ζητούμενη καταλληλότητα. Οι αλγόριθμοι αυτής της κατηγορίας, όπως και των προηγουμένων επιστρέφουν το μοντέλο διεργασιών με τη ζητούμενη αναπαράσταση. Η Εξόρυξη Διεργασιών που ακολουθεί την εξελικτική

προσέγγιση παρουσιάζει μεγαλύτερη ευελιξία και ευρωστία. Όπως και οι ευρετικοί αλγόριθμοι, διαχειρίζονται αποδοτικά τον θόρυβο και την ατέλεια του **log**. Εξαιτίας της εξελικτικής προσέγγισης που ακολουθούν οι γενετικοί αλγόριθμοι Εξόρυξης Διεργασιών είναι εύκολη η επέκταση και η προσαρμογή τους. Όμως δεν είναι αποδοτικοί για μοντέλα και **logs** μεγάλου μεγέθους καθώς απαιτούν μεγάλο χρόνο για την ανακάλυψη ενός μοντέλου που ικανοποιεί την ζητούμενη καταλληλότητα.

Στην επόμενη κατηγορία, τις τεχνικές ελέγχου συμμόρφωσης κατατάσσονται όλες οι τεχνικές και οι αλγόριθμοι οι οποίοι συσχετίζουν τα γεγονότα **event log** με τις δραστηριότητες που περιγράφονται από το μοντέλο και στοχεύουν στη σύγκριση της παρατηρούμενης συμπεριφοράς σε σχέση με την αναμενόμενη συμπεριφορά που υποδεικνύει το μοντέλο διεργασιών. Οι τεχνικές αυτές δέχονται ως είσοδο το **event log** και το μοντέλο διεργασιών, το οποίο μπορεί είτε να έχει σχεδιαστεί με το χέρι είτε να έχει ανακαλυφθεί από το **event log** μέσω άλλων τεχνικών Εξόρυξης Διεργασιών. Το μοντέλο διεργασιών [15] μπορεί να αναπαριστά την οργάνωση ακόμα και τις πολιτικές του οργανισμού και δεν περιορίζεται απαραίτητα στις διαδικασίες. Το εξαγόμενο αποτέλεσμα των τεχνικών αυτών είναι διαγνωστικές πληροφορίες που τις περισσότερες φορές υποδεικνύουν και ποσοτικοποιούν τη διαφοροποίηση ή την απόκλιση των παρατηρούμενων γεγονότων από το μοντέλο διεργασιών. Παρακάτω αναφέρονται τα εξής παραδείγματα στα οποία οι τεχνικές αυτές μπορούν να βοηθήσουν [16]:

- Στον έλεγχο των καταγεγραμμένων διεργασιών
- Στην αναγνώριση ιδιαζουσών περιπτώσεων και στην ανεύρεση των κοινών χαρακτηριστικών τους
- Στην αναγνώριση των τμημάτων των διεργασιών όπου παρατηρείται η μεγαλύτερη απόκλιση
- Σε ελεγκτικές πρακτικές (**auditing**)
- Στην αξιολόγηση της ποιότητας ενός ανακαλυφθέντος μοντέλου διεργασιών (όταν δίνεται ως είσοδος στην τεχνική)
- Στην καθοδήγηση εξελικτικών αλγορίθμων
- Ως το σημείο έναρξης για την βελτιστοποίηση του μοντέλου διεργασιών

Τις περισσότερες φορές με τις τεχνικές ελέγχου συμμόρφωσης εξετάζονται τέσσερα ποιοτικά χαρακτηριστικά κατά τη σύγκριση του μοντέλου διεργασιών με το **event log**, η καταλληλότητα (**fitness**), η απλότητα, η ακρίβεια και η γενίκευση. Η καταλληλότητα ενός μοντέλου [16] θεωρείται καλή όταν τα περισσότερα ίχνη (**traces**) που περιέχονται στο **event log** μπορούν να αναπαραχθούν από αυτό, ενώ η καταλληλότητα του μοντέλου θεωρείται ιδανική όταν όλα τα ίχνη του **event log** μπορούν να αναπαραχθούν από το μοντέλο διεργασιών που εξετάζεται. Η απλότητα ενός μοντέλου είναι έννοια ευρέως κατανοητή. Η καταλληλότητα και η απλότητα του μοντέλου δεν επαρκούν ως κριτήρια αξιολόγησης του ανακαλυφθέντος μοντέλου αφού ο σκοπός του μοντέλου διεργασιών είναι να μπορεί να αναπαραστήσει και άλλα ίχνη που μπορεί να μην εμφανίζονται στο **event log** που ελέγχεται. Έτσι το μοντέλο μπορεί να θεωρηθεί ιδανικά κατάλληλο για τα δεδομένα ίχνη αλλά να μην μπορεί να αναπαράγει άλλα επιπλέον ίχνη που δεν εξετάζονται τη στιγμή εκείνη. Για το λόγο

αυτό χρησιμοποιούνται τα χαρακτηριστικά της ακρίβειας και της γενίκευσης. Ένα μοντέλο διεργασιών ορίζεται ως ακριβές όταν αναπαριστά συγκεκριμένη συμπεριφορά, σχετική με την παρατηρούμενη στο **event log**, ενώ αντίστοιχα θεωρούμε ότι γενικεύει όταν δεν περιορίζει τη συμπεριφορά που αναπαριστά στα παραδείγματα του **log**. Η έλλειψη της ακρίβειας σε ένα μοντέλο διεργασιών συσχετίζεται με την έννοια **underfitting** της Εξόρυξης Δεδομένων υποδηλώνοντας ότι το μοντέλο παράγει συμπεριφορά που αποκλίνει από τη συμπεριφορά που περιέχεται στο αρχείο καταγραφής γεγονότων. Αντίστοιχα ένα μοντέλο που δεν παρουσιάζει γενίκευση χαρακτηρίζεται με τον ορισμό **overfitting** μιας και περιορίζει τη συμπεριφορά που αναπαριστά συγκεκριμένα στη συμπεριφορά του event log.

Οι τεχνικές που επιχειρούν τον έλεγχο συμμόρφωσης ακολουθούν μία από τις τρεις προσεγγίσεις. Η πρώτη προσέγγιση βασίζεται στην έννοια του αποτυπώματος (**footprint**). Ως αποτύπωμα ορίζεται ένας πίνακας που αναπαριστά τις εξαρτήσεις μεταξύ των δραστηριοτήτων του μοντέλου. Σε αυτές τις μεθόδους αναγνωρίζονται οι αιτιολογικές εξαρτήσεις του μοντέλου διεργασιών και συγκρίνονται με εκείνες που εμφανίζονται στο **event log**. Η δεύτερη προσέγγιση βασίζεται στην αναπαραγωγή των ιχνών του **event log** μέσω του μοντέλου διεργασιών που δίνεται ως είσοδος. Στις περισσότερες τεχνικές από αυτές υπολογίζεται ο αριθμός των ιχνών που αναπαράγονται πλήρως από το μοντέλο διεργασιών. Η τρίτη και πιο εξελιγμένη προσέγγιση αφορά τον υπολογισμό της βέλτιστης ευθυγράμμισης μεταξύ κάθε ίχνους του **log** και της πλησιέστερης σε αυτό συμπεριφοράς που περιγράφεται από το μοντέλο διεργασιών.

Η τρίτη κατηγορία συγκεντρώνει τις τεχνικές βελτιστοποίησης οι οποίες στοχεύουν στη βελτίωση ή την επέκταση του μοντέλου διεργασιών μέσω της πληροφορίας που συγκεντρώνει η καταγραφή των πραγματικών γεγονότων στο **event log**. Οι τεχνικές αυτές δέχονται ως είσοδο το μοντέλο διεργασιών και το αρχείο καταγραφής γεγονότων και παράγουν ένα βελτιωμένο μοντέλο διεργασιών. Η τροποποίηση του μοντέλου διεργασιών γίνεται βάσει των διαγνωστικών που προκύπτουν από την ευθυγράμμιση του μοντέλου διεργασιών με το **log** μέσω των τεχνικών ελέγχου συμμόρφωσης. Προς αυτή την κατεύθυνση μπορούν επίσης να αξιοποιηθούν περαιτέρω χαρακτηριστικά τα οποία καταγράφονται στο **event log** όπως ο χρόνος εκτέλεσης των **events** ή ο πόρος που εκτελεί κάθε **event** έτσι ώστε να κατασκευαστούν κοινωνικά δίκτυα αρμονικά με τη ροή εργασιών και την απόδοση των πόρων.

Τα πλεονεκτήματα της εφαρμογής των τεχνικών Εξόρυξης Διεργασιών γίνονται άμεσα αντιληπτά για την περίπτωση που επιχειρείται η μοντελοποίηση και η βελτιστοποίηση των διεργασιών ελέγχου ροής ενός οργανισμού. Όμως οι τεχνικές ανακάλυψης και ελέγχου συμμόρφωσης δεν περιορίζονται στον έλεγχο ροής αλλά μπορούν να συνεισφέρουν και σε άλλες πλευρές ενός οργανισμού. Η Εξόρυξη Διεργασιών [16],[20] μπορεί να καλύψει τις ακόλουθες προοπτικές:

- **Προοπτική ελέγχου ροής:** η Εξόρυξη Διεργασιών μπορεί να εστιάσει στον έλεγχο ροής όπως για παράδειγμα την ταξινόμηση των δραστηριοτήτων ενός οργανισμού, με στόχο να αξιολογήσει όλα τα πιθανά μονοπάτια διεξαγωγής των διεργασιών με τη βοήθεια κάποιας σημειογραφίας

- **Οργανωτική προοπτική:** η Εξόρυξη Διεργασιών μπορεί να εστιάσει στις πληροφορίες σχετικά με τους πόρους που εκτελούν τις δραστηριότητες, οι οποίες εμπεριέχονται στο αρχείο καταγραφής γεγονότων, προκειμένου να ανακαλύψει τις συσχετίσεις τους και τις ευθύνες τους. Με αυτό το εργαλείο μπορεί να δομηθεί καλύτερα το ανθρώπινο δυναμικό του οργανισμού και να απονεμηθούν κατάλληλα οι πόροι του οργανισμού στα αντίστοιχα τμήματά του
- **Προοπτική περίπτωσης:** οι περιπτώσεις που καταγράφονται στο **event log** παρουσιάζουν επιπλέον χαρακτηριστικά πέρα από το μονοπάτι που ακολουθούν και μπορούν να αξιολογηθούν βάσει αυτών των χαρακτηριστικών εξίσου
- **Προοπτική χρόνου:** στο **event log** καταγράφεται ο χρόνος εκτέλεσης και η συχνότητα κάθε δραστηριότητας, δεδομένα τα οποία μπορούν να αξιοποιηθούν για την αξιολόγηση του συστήματος, την αναγνώριση προβληματικών περιοχών όπως σημείων συμφόρησης, την κατανομή των πόρων σε αυτά αλλά και να προβλεφθεί ο εναπομείναντας χρόνος για τις τρέχουσες περιπτώσεις

2.3.3 Συσχέτιση Εξόρυξης Διεργασιών με άλλους τομείς

Προκειμένου κάποιος να μπορέσει να κατανοήσει πλήρως τη θέση του τομέα της Εξόρυξης Διεργασιών σε σχέση με τον τομέα Διαχείρισης Επιχειρηματικών Διεργασιών που αναφέρθηκε στην προηγούμενη παράγραφο, είναι χρήσιμη μια πολύ σύντομη υπενθύμιση του κύκλου ζωής της Διαχείρισης Επιχειρηματικών Διεργασιών. Ο τυπικός κύκλος ζωής του τομέα **BPM** αποτελείται από τέσσερις φάσεις:

- **Σχεδίαση διεργασίας:** Στη φάση αυτή εκτελείται οποιαδήποτε ενέργεια επιχειρεί τη μοντελοποίηση των επιχειρηματικών διεργασιών, είτε αυτές έχουν οριστεί στο παρελθόν είτε σχεδιάζονται για πρώτη φορά και καταλήγει στην κατασκευή του μοντέλου διεργασιών. Το μοντέλο διεργασιών (**process model**) όπως αναλύεται και παρακάτω, εκφράζει την επιθυμητή συμπεριφορά της διεργασίας η οποία αναμένεται να ακολουθείται σε κάθε εκτέλεσή της. Αποτελεί βασική έννοια του τομέα και σημείο αναφοράς και σύνδεσης του **BPM** με την Εξόρυξη Διεργασιών την οποία μελετάμε.
- **Παραμετροποίηση συστήματος:** Εφόσον έχει ολοκληρωθεί η σχεδίαση της ζητούμενης διεργασίας ξεκινά η κατασκευή του συστήματος που θα υλοποιεί την διεργασία όπως αυτή έχει οριστεί από το μοντέλο διεργασιών. Στο βήμα αυτό συνήθως αναπτύσσεται λογισμικό που ανταποκρίνεται στις απαιτήσεις ή παραμετροποιείται κατάλληλα κάποιο υπάρχον σύστημα/τεχνολογία έτσι ώστε να υλοποιεί την επιθυμητή λειτουργικότητα.
- **Θέσπιση διεργασίας:** Στο στάδιο αυτό το υλοποιημένο σύστημα που αφορά τις διεργασίες του οργανισμού τίθεται σε λειτουργία
- **Διάγνωση:** Η φάση της διάγνωσης λαμβάνει χώρα ανά τακτά χρονικά διαστήματα προκειμένου να αξιολογηθεί το παραπάνω σύστημα, να διαγνωστούν πιθανά σφάλματα, βελτιωθεί η αποδοτικότητά του, ενημερωθούν οι τεχνολογίες στις οποίες βασίζεται, προστεθούν ή τροποποιηθούν διεργασίες και γενικότερα να διασφαλιστεί η αρμονία του συστήματος σε λειτουργία με το ζητούμενο περιβάλλον. Το βήμα αυτό τις περισσότερες φορές οδηγεί σε έναν νέο κύκλο σχεδίασης,

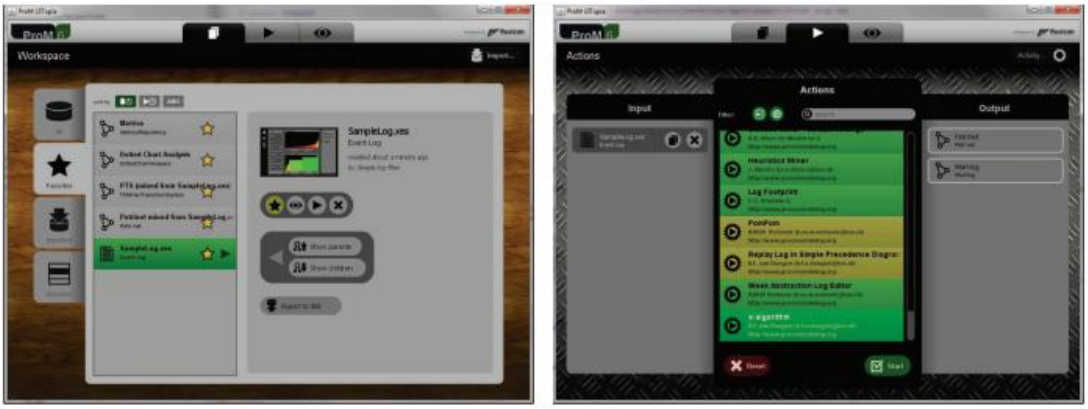
παραμετροποίησης και θέσπισης διεργασίας έτσι ώστε να ενσωματωθούν στο νέο σύστημα όλα τα παραπάνω.

Τα πλεονεκτήματα της Εξόρυξης Διεργασιών ως προς την χρήση της στη φάση διάγνωσης είναι εύκολα κατανοητά, όμως δεν περιορίζονται στη φάση αυτή και οι τεχνικές Εξόρυξης Διεργασιών μπορούν να αξιοποιηθούν και στις υπόλοιπες. Ένα παράδειγμα είναι η χρήση τεχνικών Εξόρυξης Διεργασιών για τη λειτουργική υποστήριξη όπου οι παραγόμενες προβλέψεις και συστάσεις μπορούν να επηρεάσουν τις τρέχουσες περιπτώσεις των διεργασιών [22].

Η διαφοροποίηση της Εξόρυξης Διεργασιών από την Ευφυΐα Επιχειρηματικών Διεργασιών (BPI) είναι δύσκολο να αναγνωρισθεί. Εάν βασιστούμε στον ορισμό του **BPI** [23] που υποστηρίζει ότι το **BPI** μπορεί να θεωρηθεί ως η Επιχειρηματική Ευφυΐα που εστιάζει στις διεργασίες, οι όροι **BPI** και Εξόρυξη Διεργασιών είναι ισοδύναμοι. Ενδεικτικές ερωτήσεις που κατά κύριο λόγο ζητείται να απαντηθούν με τα εργαλεία του **BPI** ή της Εξόρυξης Διεργασιών είναι το ποιες διεργασίες πραγματικά εκτελούνται από το ανθρώπινο δυναμικό του οργανισμού, ποια είναι τα σημεία συμφόρησης σε κάθε διεργασία και πως μπορούν να διορθωθούν, αν μπορούμε να προβλέψουμε τα προβλήματα που θα παρουσιάσουν οι τρέχουσες περιπτώσεις, τι αντίμετρα μπορούμε να προτείνουμε, ποια είναι η κατάλληλη επανασχεδίαση της διεργασίας δεδομένων κάποιων στόχων κ.ά. Προσοχή επίσης απαιτείται στη συσχέτιση της Εξόρυξης Διεργασιών με την Εξόρυξη Δεδομένων. Οι δυο τομείς δεν πρέπει να συγχέονται μιας και οι κλασσικές τεχνικές της Εξόρυξης Δεδομένων όπως η ταξινόμηση, η ομαδοποίηση, η παλινδρόμηση κ.λπ. δεν λαμβάνουν υπόψιν την πληροφορία των διεργασιών και τις περισσότερες φορές χρησιμοποιούνται για την ανάλυση ενός μόνο βήματος μιας διεργασίας. Επίσης [15] τα μοντέλα διεργασιών δεν μπορούν να περιγραφτούν ή να συγκριθούν με τις υπάρχουσες δομές της Εξόρυξης Δεδομένων και απαιτούν ειδικούς τύπους δομών και αλγορίθμων. Η Εξόρυξη Διεργασιών αναγνωρίζεται ως μια γέφυρα μεταξύ των τεχνικών που εστιάζουν στις διεργασίες και των τεχνικών που εστιάζουν στα δεδομένα, δηλαδή μεταξύ του **BPM** και τομέων όπως **Data Mining** και **Machine Learning** [25].

2.3.4 Εργαλεία Εξόρυξης Διεργασιών

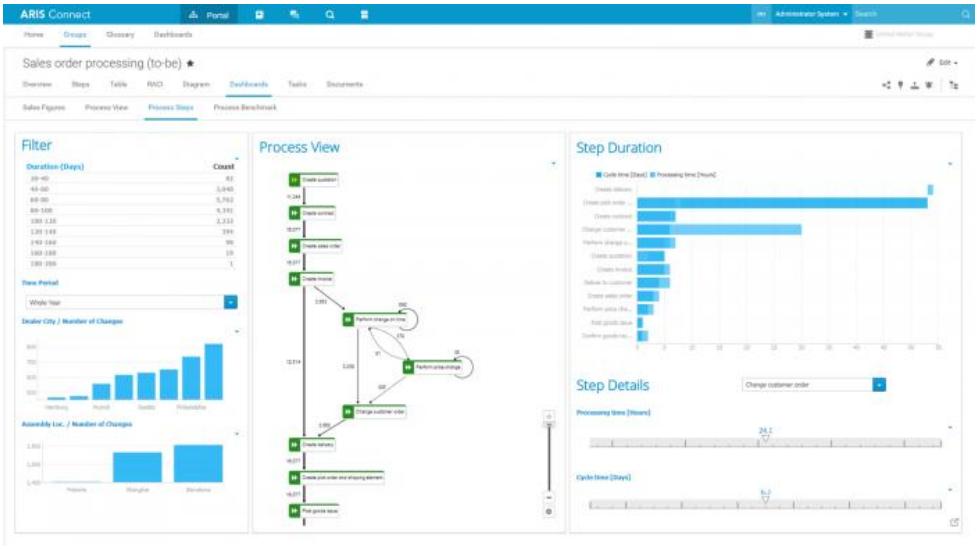
Prom



Εικόνα 10 Prom interface

Με την εξέλιξη του τομέα της Εξόρυξης Διεργασιών αρκετά εμπορικά λογισμικά έχουν ενσωματώσει τεχνικές Εξόρυξης Διεργασιών και έχουν αναπτυχθεί πολλές υλοποιήσεις για τους διαφορετικούς αλγορίθμους του τομέα. Επειδή κάθε σύστημα εξάγει και αποθηκεύει τα αρχεία γεγονότων με τη δική του μορφή, οι περισσότερες από αυτές τις υλοποιήσεις λειτουργούν απομονωμένα από τις υπόλοιπες, γεγονός που καθιστά την πρακτική εφαρμογή της Εξόρυξης Διεργασιών δυσκολότερη. Η πλατφόρμα Εξόρυξης Διεργασιών **ProM**, αποτελεί το σημείο αναφοράς του τομέα. Το **ProM** είναι ένα επεκτάσιμο περιβάλλον Εξόρυξης Διεργασιών, ευέλικτο ως προς τη μορφή των δεδομένων εισόδου και εξόδου. Είναι διαθέσιμο με open source δικαιώματα έτσι ώστε να είναι δυνατή η επαναχρησιμοποίηση του κώδικα από τους δημιουργούς νέων επεκτάσεων. Το **ProM** αποτελείται από επεκτάσεις (**plug-ins**) και επιτρέπει την αλληλεπίδραση μεταξύ των επεκτάσεων του. Μια επέκταση είναι ουσιαστικά η υλοποίηση ενός αλγορίθμου ο οποίος χρησιμοποιείται με κάποιο τρόπο στον τομέα, όπου η υλοποίηση αυτή ικανοποιεί τις προδιαγραφές της πλατφόρμας. Η προσθήκη ενός νέου **plug-in** σε μία εγκατάσταση **ProM** δεν απαιτεί κάποια αλλαγή στο ίδιο το εργαλείο όπως για παράδειγμα τη μεταγλώττιση της πλατφόρμας, ενώ στις τελευταίες εκδόσεις του **ProM** έχει προστεθεί εφαρμογή γραφικού περιβάλλοντος για τη διαχείριση των επεκτάσεων αυτών. Στις πρώτες αναφορές του **ProM**, περιλαμβάνονταν ελάχιστες επεκτάσεις για την κάλυψη των βασικότερων λειτουργιών και τεχνικών της Εξόρυξης Διεργασιών, όπως η εισαγωγή, εξαγωγή, ανάλυση και ο μετασχηματισμός αρχείων. Στις τελευταίες εκδόσεις του λογισμικού μέσω της εφαρμογής διαχείρισης επεκτάσεων (**ProM Package Manager**) είναι διαθέσιμες περισσότερες από 280 επεκτάσεις. Μια ακόμη σημαντική διαφοροποίηση του εργαλείου στην παρούσα έκδοση σε σχέση με τις αρχικές εκδόσεις του είναι η υποστήριξη και καθιέρωση του προτύπου **XES** και η αντικατάσταση του αρχικού προτύπου **MXML** με αυτό.

ARIS Process Performance Manager

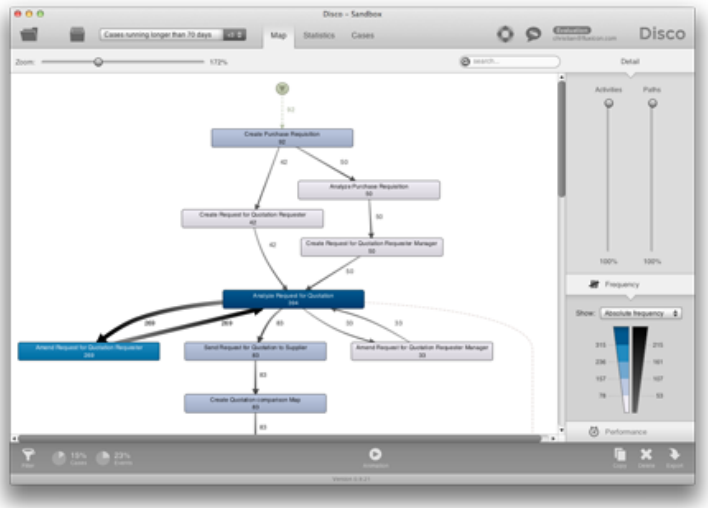


Εικόνα 11 ARIS interface

Το **Aris Process Performance Manager** είναι ένα εμπορικό λογισμικό για εξόρυξη επιχειρησιακών διεργασιών. Χρησιμοποιεί τις διεργασίες για να ανακαλύψει και να αναλύσει ψηφιακά μια επιχείρηση και να βελτιώσει την απόδοση της. Το **Aris Process Performance Manager** μπορεί να κάνει τα παρακάτω:

- Εφαρμόσει συμπεράσματα της εξόρυξης διεργασιών για να κάνει την επιχείρηση αποτελεσματικότερη
- Να αναλύσει παραμέτρους των διεργασιών από την πιο συχνά παρατηρήσιμη ροή διεργασιών και να αποτυπώσει οπτικά όλες τις υπάρχουσες διεργασίες
- Να χρησιμοποιήσει πληροφορίες πάνω στις επιχειρηματικές διεργασίες και στις εξαρτήσεις μεταξύ αυτών με σκοπό την μείωση του κόστους και της πολυπλοκότητας
- Να εφαρμόσει επιπλέον μετρικές στην ανάλυση ενός μέρους της διεργασίας

Disco (Fluxicon)

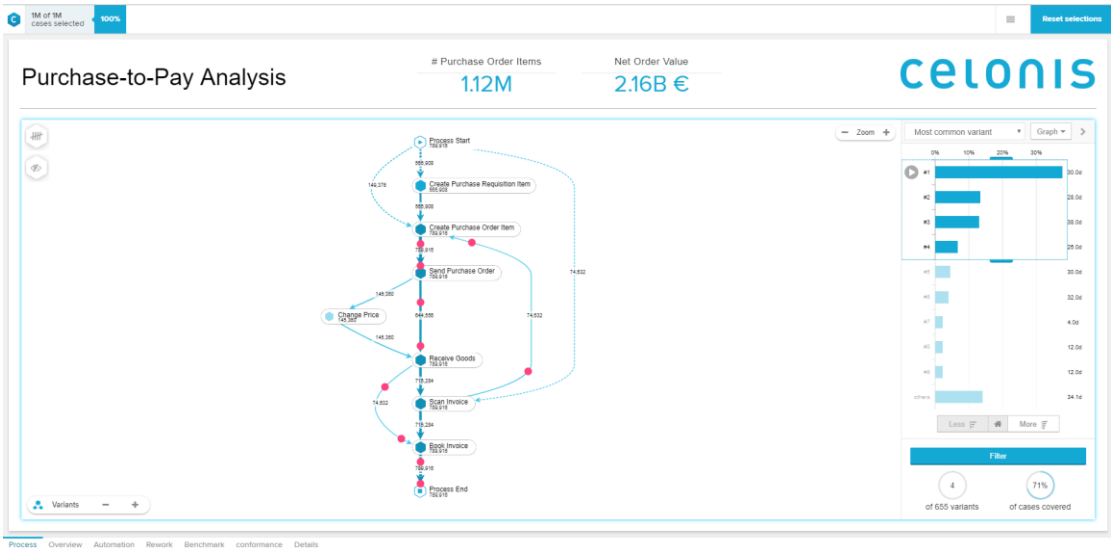


Εικόνα 12 Disco interface

Το **Disco** είναι ένα ευρέως διαδεδομένο εργαλείο για εξόρυξη διεργασιών το οποίο είναι ισχυρό, εύκολο στην χρήση του και γρήγορο. Η επαναστατική εμπορική τεχνολογία που χρησιμοποιεί για την εξόρυξη διεργασιών βοηθάει τους ερευνητές να αναπαραστήσουν οπτικά χάρτες των δεδομένων των διεργασιών σε σύντομο χρονικό διάστημα. Με το **Disco** μπορούμε να κάνουμε τα παρακάτω:

- Να εισάγουμε και να εξάγουμε δεδομένα σε διάφορες μορφές (**formats**)
- Να εξερευνήσουμε δεδομένα των ξεχωριστών περιπτώσεων (**case**) και των **events**.
- Να κάνουμε φιλτράρισμα των δεδομένων (**data filtering**) και φιλτράρισμα των διαδικασιών και των προτύπων τους (**process pattern-oriented filtering**)
- Αυτόματη ανακάλυψη διεργασιών
- **Process map animation** για την ανακάλυψη σημείων συμφόρησης και σκιαγράφηση της απόδοσης
- Λεπτομερής στατιστικές μετρήσει και γραφήματα

CELONIS



Εικόνα 13 Celonis interface

Το **Celonis** είναι ένα έξυπνο λογισμικό εξόρυξης διεργασιών το οποίο αναλύει και οπτικοποιεί επιχειρηματικές διεργασίες βασιζόμενο σε δεδομένα IT Συμβάλει στην ανακάλυψη αδυναμιών και κάνει τις διεργασίες περισσότερο κατανοητές, γρηγορότερες και λιγότερο κοστοφόρες. Το **Celonis** βοηθάει τους οργανισμούς – επιχειρήσεις να κατανοήσουν σε σύντομο χρονικό διάστημα και να βελτιώσουν αποτελεσματικά την ροή των λειτουργικών διεργασιών για την αναβάθμιση της επιχείρησης. Το **Celonis** αναλυτικά επικεντρώνεται στα παρακάτω:

- Εξόρυξη δεδομένων από τα λειτουργικά συστήματα της εταιρείας
- Στην οπτικοποίηση των διεργασιών σε πραγματικό χρόνο
- Στην ανακάλυψη-αναγνώριση των βασικών λόγων που προκαλούν προβλήματα σε μια διεργασία και στην ανακάλυψη τρόπων για την βελτίωση της
- Στον έλεγχο και στην μέτρηση του αντίκτυπου που έχει η στρατηγική που ακολουθείται σε μια διεργασία και στην υποκίνηση του οργανισμού προς την συνεργατική εργασία για την διασφάλιση της συνεχούς επιτυχίας
- Στην χρήση εργαλείων AI για την εξόρυξη πληροφορίας και για την βελτίωση της ταχύτητας εκτέλεσης των επιχειρησιακών διεργασιών

2.4 Νευρωνικά Δίκτυα Μακράς και Βραχείας Μνήμης

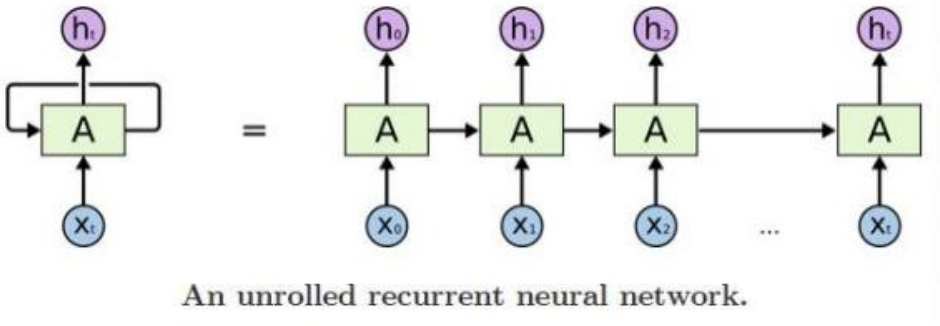
2.4.1 Γενικά χαρακτηριστικά ενός αναδρομικού δικτύου

Τα αναδρομικά νευρωνικά δίκτυα (NN), όπως έχουμε αναφέρει, είναι μία ειδική κατηγορία νευρωνικών δικτύων που επεξεργάζονται αποτελεσματικά κάθε είδους ακολουθιακά δεδομένα. Παραδείγματα, αποτελούν η φωνή, η γραφή, η οπτική πληροφορία που προκύπτει από μία κίνηση ή μία δράση ή ακόμα και οικονομικά δεδομένα και δείκτες που προκύπτουν από στοιχεία προηγούμενων χρονικών περιόδων. Ένα αναδρομικό δίκτυο μετασχηματίζει κάθε νέα είσοδο με τρόπο που εξαρτάται τόσο από την ίδια την είσοδο όσο και από τις προηγούμενες εισόδους που έχει δεχτεί. Φορμαλιστικά η βασική αυτή αρχή περιγράφεται ως εξής: Αν x_t είναι η είσοδος την χρονική στιγμή t, f(·) η συνάρτηση που περιγράφει την επίδραση του αναδρομικού δικτύου πάνω στην είσοδο και h_t η έξοδος του, τότε:

$$h_t = f(x_t, h_{t-1}) = f(x_t, f(x_{t-1}, h_{t-2})) = \cdots = f(x_t, f(x_{t-1}, \ldots, (f(x_1, h_0)) \ldots))$$

Εικόνα 14 Εξίσωση εξόδου αναδρομικού δικτύου

Είναι φανερό πως υπάρχει στενή σχέση ανάμεσα στα γραφικά μοντέλα (**graphical models**) και τα αναδρομικά δίκτυα. Αυτή η σύνδεση ανάμεσα στις δύο κατηγορίες μοντέλων μπορεί να φανεί καλύτερα αν, όπως στο παρακάτω σχήμα, το αναδρομικό δίκτυο ”ξεδιπλωθεί”, σχεδιάζοντας ”εικονικές” υπολογιστικές μονάδες για να αναπαρασταθούν οι αναδρομικές εκτελέσεις της συνάρτησης f πάνω στις εισόδους x_t .



An unrolled recurrent neural network.

Εικόνα 15 Αριστερά ένα μη ξεδιπλωμένο NN και δεξιά ένα ξεδιπλωμένο

Σημειώνεται πως η έξοδος h_t συχνά αναφέρεται και ως κατάσταση του δικτύου. Στις παραπάνω εξισώσεις δεν εισήχθη για λόγους απλότητας η ύπαρξη ενός συνόλου παραμέτρων θ ή βαρών που όπως και στις άλλες περιπτώσεις νευρωνικών μοντέλων καθορίζουν την επίδραση του μοντέλου πάνω στην είσοδο τους. Οι παράμετροι αυτοί, και για τα αναδρομικά νευρωνικά δίκτυα είναι υπό μάθηση. Τα βάρη σε ένα αναδρομικό μοντέλο μπορούν να διαχωριστούν σε δύο κατηγορίες: αυτά που επιδρούν πάνω στην είσοδο (W) και αυτά που καθορίζουν την σημασία που

δίνεται στην προηγούμενη κατάσταση του μοντέλου για τον υπολογισμό της

$$h_t = \phi(W x_t + U h_{t-1} + b)$$

επόμενης κατάστασης ( ).

Εικόνα 16 Γενική μορφή των εξισώσεων ενός αναδρομικού δικτύου

όπου φ(·) μία μη γραμμική συνάρτηση ενεργοποίησης. Λόγω αυτού του μηχανισμού τα αναδρομικά δίκτυα έχουν την δυνατότητα να μοντελοποιούν χρονικές εξαρτήσεις ακόμα και ανάμεσα σε μη συνεχόμενες παρατηρήσεις, αφού μέσω της αναδρομικότητας υλοποιούν έναν μηχανισμό μνήμης. Όμως, στην γενική μορφή τους τα νευρωνικά αναδρομικά δίκτυα δεν καταφέρνουν να μοντελοποιήσουν αποτελεσματικά εξαρτήσεις μακράς διάρκειας (όπου η χρήση του όρου μακράς είναι σχετική και εξαρτάται από το πρόβλημα).

Η δυσκολία αυτή πηγάζει από την αδυναμία του τρόπου εκπαίδευσης τους, που υλοποιείται μέσω της συνέργειας του αλγορίθμου **Back Propagation** στο χρόνο και του αλγορίθμου **Stochastic Gradient Descent S.G.D**, στο να διατηρήσει σημαντικές κλίσεις για εισόδους τους παρελθόντος. Κύρια αιτία είναι το γεγονός πως ο υπολογισμός της κλίσης του κόστους ως προς κάποια παρελθοντική είσοδο περιλαμβάνει την παραγώγιση μίας σύνθεσης συναρτήσεων που οδηγεί σε ένα όλο και αυξανόμενου αριθμού παραγόντων γινόμενο, καθώς κινούμαστε προς το παρελθόν. Το πρόβλημα αυτό συνήθως αναφέρεται ως εξαφάνιση της κλίσης (**the vanishing gradient problem**) και έχει σαν αποτέλεσμα η διαδικασία εκπαίδευσης να μην εντοπίζει ικανοποιητικά καλές τιμές για τα βάρη του μοντέλου ώστε αυτό να εντοπίζει πρότυπα που προκύπτουν μεταξύ εισόδων που απέχουν στον άξονα του χρόνου.

Η παραπάνω δυσλειτουργία των αναδρομικών δικτύων, αντιμετωπίστηκε μέσω της εισαγωγής των νευρώνων Μακράς-Βραχείας Μνήμης (**Long Short Term Memory** ή **L.S.T.M**) που προτάθηκαν πρώτη φορά από τους **Hochreiter** και **Schmidhuber**. Το πιο εύρωστο αυτό είδος αναδρομικών δικτύων έχει φανεί πολύ αποτελεσματικό σε εφαρμογές όπως η επεξεργασία και μετάφραση φυσικής γλώσσας ή η αναγνώριση συνεχούς χειρόγραφου.

2.4.2 Οι νευρώνες Μακράς- Βραχείας Μνήμης (Long Term Short Term Memory)

Για την παρουσίαση των δικτύων **LSTM** [24],[25], αρχικά θα περιγραφεί ο φορμαλισμός που τους διέπει και στη συνέχει θα δοθεί μία σύντομη ποιοτική ερμηνεία της λειτουργίας τους. θεωρούμε ότι η κατάσταση ενός νευρώνα **LSTM** τη χρονική στιγμή t είναι μια συλλογή από διανύσματα του χώρου R^d. Συγκεκριμένα έχουμε:

- Την θύρα εισόδου (**input gate**) i_t
- Την θύρα λησμόνησης (**forget gate**) f_t
- Ένα κύτταρο μνήμης (**memory cell**) c_t
- Την θύρα εξόδου (**output gate**) o_t
- Την κατάσταση ή κρυφή κατάσταση (**hidden state**) h_t

Δεδομένου ότι οι μη γραμμικές συναρτήσεις ενεργοποίησης που χρησιμοποιούνται είναι η σιγμοειδής συνάρτηση σ(·) και η υπερβολική εφαπτομένη tanh(·).

$$
\begin{aligned}
i_t &= \sigma(W^i x_t + U^i h_{t-1} + b^i) \\
f_t &= \sigma(W^f x_t + U^f h_{t-1} + b^f) \\
o_t &= \sigma(W^o x_t + U^o h_{t-1} + b^o \\
u_t &= \tanh(W^u x_t + U^u h_{t-1} + b^u \\
c_t &= i_t \odot u_t + f_t \odot c_{t-1} \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

Εικόνα 17 Εξισώσεις κατάστασης που χαρακτηρίζουν έναν νευρώνα LSTM

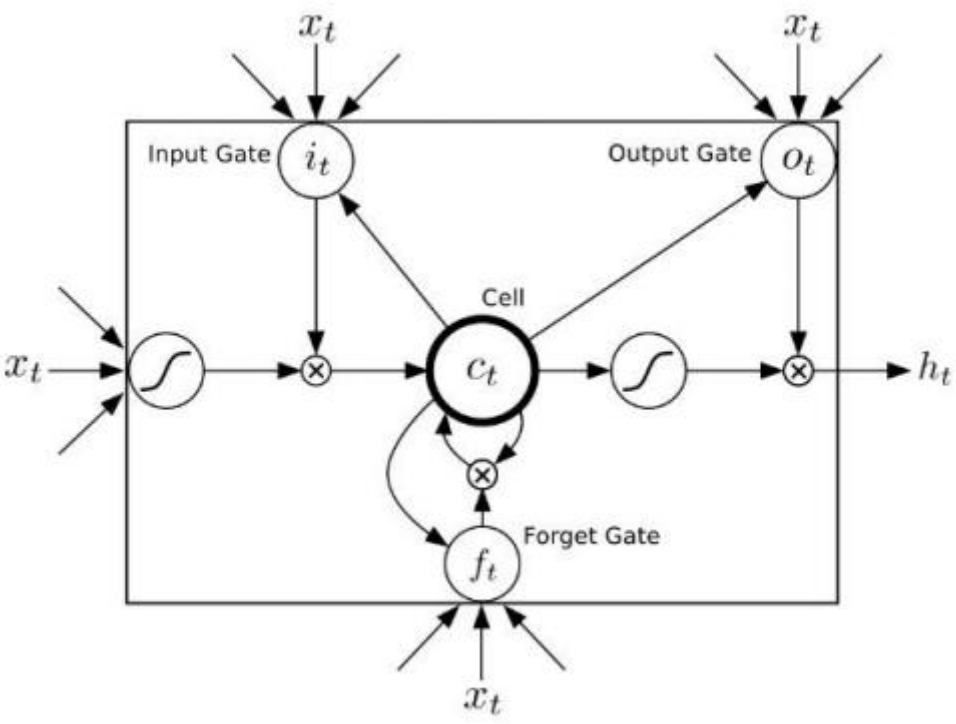
όπου x_t η είσοδος την χρονική στιγμή t και ⊙ το γινόμενο **Hadamard**. Διαισθητικά μπορούμε να ερμηνεύσουμε την λειτουργία του μοντέλου ως εξής: πως η θύρα λησμόνησης ελέγχει το κατά πόσο οι παρελθούσες καταστάσεις πρέπει να ληφθούν υπόψη κατά το επόμενο βήμα και η θύρα εισόδου ελέγχει το ποια διανύσματα του νευρώνα θα μεταβληθούν και σε ποιο βαθμό. Πιο συγκεκριμένα:

• Η θύρα λησμόνησης ελέγχει το κατά πόσο πρέπει το περιεχόμενο του κυττάρου μνήμης ct να ξεχαστεί κατά την μετάβαση στο επόμενο βήμα λειτουργίας.

• Η θύρα εξόδου ελέγχει την έκθεση της εσωτερικής μνήμης στα υπόλοιπα μέρη του νευρώνα.

• Η θύρα εισόδου ελέγχει το ποια διανύσματα του νευρώνα θα μεταβληθούν και σε ποιο βαθμό.

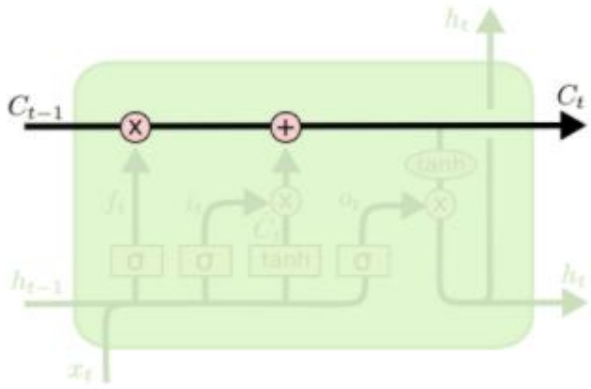
• Η κρυφή κατάσταση αναπαριστά μία περιορισμένη και ελεγχόμενη εξωτερίκευση της αποθηκευμένης στο κύτταρο πληροφορίας η οποία εξαρτάται και από την νέα είσοδο x_t και όλες τις προηγούμενες λειτουργίες



Εικόνα 18 Κύτταρο LSTM και πύλες

2.4.3 Η κύρια ιδέα πίσω από το LSTM

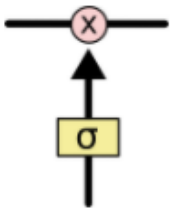
Το ‘‘κλειδί’’ στα **LSTMs** είναι η κατάσταση κελύφους (**cell state**), η οριζόντια γραμμή που περνάει από την κορυφή του διαγράμματος παρακάτω. Η κατάσταση κελύφους είναι κάτι σαν ένας ιμάντας μεταφοράς. Περνάει από όλη την αλυσίδα, με μόνο κάποιες πολύ μικρές γραμμικές αλληλεπιδράσεις. Είναι πολύ εύκολο για την πληροφορία να περάσει χωρίς αλλαγές.



Εικόνα 19 Η κατάσταση κελύφους περνάει από όλη την αλυσίδα

Το **LSTM** έχει την δυνατότητα να αφαιρέσει ή να προσθέσει πληροφορία στην κατάσταση κελύφους μέσω των πυλών που διαθέτει. Οι πύλες είναι ένας τρόπος για

να αφήσουμε ή όχι την πληροφορία να περάσει. Απαρτίζονται από ένα σιγμοειδή νευρωνικό στρώμα και από και εκεί εκτελούν πολλαπλασιαστικές λειτουργίες.

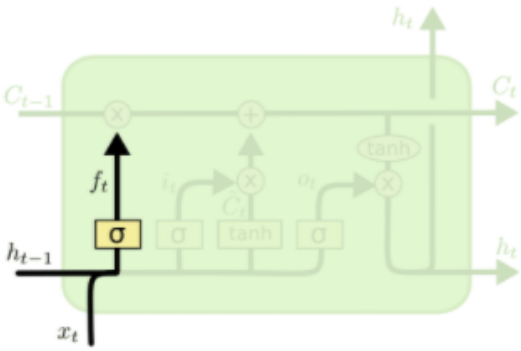


Εικόνα 20 Σιγμοειδή στρώμα πύλης όπου αποφασίζεται πόση πληροφορία θα περάσει

Το σιγμοειδή στρώμα παρέχει σαν έξοδο αριθμούς μεταξύ του μηδενός και του ένα, περιγράφοντας έτσι πόση πληροφορία πρέπει να περάσει. Μια τιμή ίση με μηδέν σημαίνει πως δεν πρέπει να περάσει καθόλου, αντίθετα μια τιμή ίση με ένα αφήνει όλη την πληροφορία να περάσει. Το **LSTM** έχει τρεις πύλες για να προστατεύει και να ελέγχει την κατάσταση κελύφους.

2.4.4 Αναλυτική επεξήγηση λειτουργίας LSTM

Το πρώτο βήμα για το **LSTM** είναι να αποφασίσει τι πληροφορία θα πετάξει από την κατάσταση κελύφους. Αυτή η απόφαση λαμβάνεται στο σιγμοειδή στρώμα το οποίο ονομάζεται στρώμα θύρας λησμόνησης (**forget gate layer**). Κοιτάει το h_{t-1} και το x_t και δίνει σαν έξοδο έναν αριθμό μεταξύ του μηδενός και του ένα για κάθε αριθμό C_{t-1} στην κατάσταση κελύφους. Το ‘1’ σημαίνει πως πρέπει να κρατήσει τα πάντα, ενώ το ‘0’ σημαίνει πως πρέπει να πετάξει τα πάντα. Ας πάρουμε το παράδειγμα ενός γλωσσικού μοντέλου που προσπαθεί να προβλέψει την επόμενη λέξη βάση των προηγούμενων λέξεων. Σε ένα τέτοιο πρόβλημα, η κατάσταση κελύφους πρέπει να λάβει υπόψιν της το γένος της λέξης ώστε να χρησιμοποιηθούν οι κατάλληλες αντωνυμίες. Όταν βλέπουμε μια καινούργια λέξη θέλουμε να ξεχάσουμε το γένος της παλιάς λέξης.



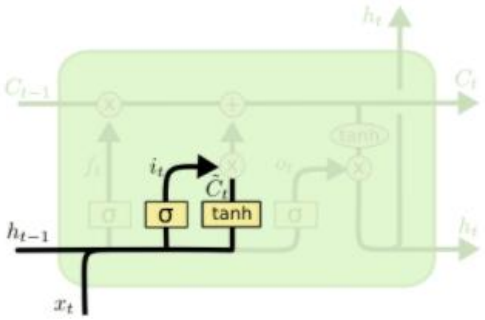
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Εικόνα 21 Θύρα λησμόνησης (Forget gate)

Το επόμενο βήμα είναι να αποφασίσει τι νέα πληροφορία θα αποθηκεύσει στην κατάσταση κελύφους. Αυτό το βήμα έχει δύο μέρη. Αρχικά το σιγμοειδή στρώμα το οποίο καλείται ‘’στρώμα θύρας εισόδου’’ (**input gate layer**) αποφασίζει ποιες τιμές θα ενημερωθούν, Στην συνέχεια, το στρώμα εφαπτομένης (**tanh layer**) δημιουργεί έναν πίνακα με νέες πιθανές τιμές, $\check{C}_t$, που μπορεί να προστεθούν στην κατάσταση.

Στο επόμενο βήμα συνδυάζουμε τα δύο παραπάνω βήματα για να ενημερώσουμε την κατάσταση κελύφους.

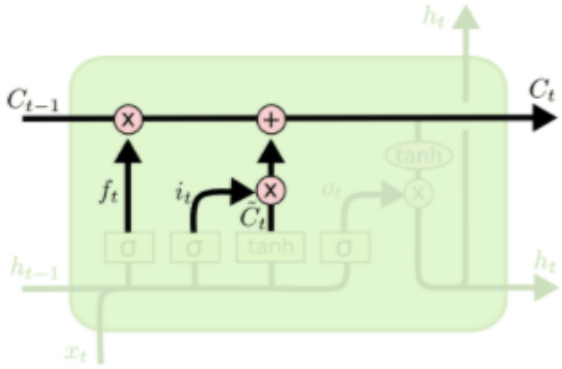
Στο παράδειγμα για το γλωσσικό μοντέλο θέλουμε να προσθέσουμε το γένους της νέας λέξης στην κατάσταση κελύφους για να αντικαταστήσουμε την προηγούμενη λέξη.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Εικόνα 22 Input gate layer και tanh layer

Τώρα πρέπει να ενημερώσουμε την παλιά κατάσταση κελύφους, C_{t-1}, στην νέα κατάσταση C_t. Στα προηγούμενα βήματα αποφασίστηκε τι να κάνουμε σε αυτό το βήμα. Πολλαπλασιάζουμε την παλιά κατάσταση με f_t, ξεχνώντας έτσι αυτά που έχουμε αποφασίσει πως πρέπει να ξεχαστούν στο προηγούμενο βήμα. Στην συνέχεια προσθέτουμε το $i_t * \check{C}_t$. Αυτές είναι οι νέες υποψήφιες τιμές, κλιμακωμένες ανάλογα με το πόσο έχουμε αποφασίσει να ενημερώσουμε κάθε τιμή της κατάστασης. Στην περίπτωση του γλωσσικού μοντέλου, σε αυτό το σημείο πετάμε την πληροφορία του γένους της παλιάς λέξης και προσθέτουμε πληροφορία όπως αποφασίστηκε στα προηγούμενα βήματα.

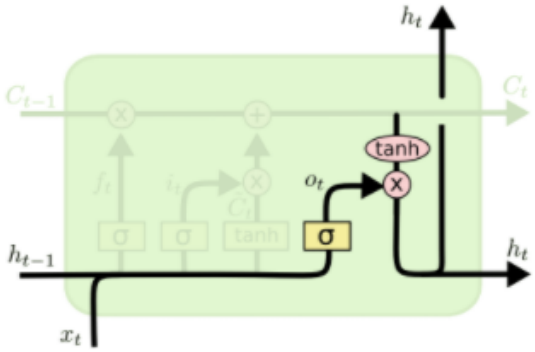


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Εικόνα 23 Ενημέρωση cell state από Ct-1 στην νέα κατάσταση Ct

Στο τέλος χρειάζεται να αποφασιστεί τι θα βγει στην έξοδο. Η έξοδος θα βασιστεί στην κατάσταση κελύφους και θα είναι φιλτραρισμένη. Αρχικά μέσω του σιγμοειδούς στρώματος αποφασίζουμε ποια μέρη της κατάστασης κελύφους θα στείλουμε στην έξοδο. Στην συνέχεια περνάμε την κατάσταση κελύφους μέσω του **tanh** (για να μετατρέψουμε τις τιμές μεταξύ -1 και 1) και πολλαπλασιάζουμε αυτό που θα πάρουμε με την έξοδο από την σιγμοειδή πύλη. Με αυτό τον τρόπο παίρνουμε σαν έξοδο μόνο τα μέρη που έχουμε αποφασίσει εμείς πως θέλουμε.

Για το παράδειγμα του γλωσσικού μοντέλου, από την στιγμή που έχει εντοπίσει κάποια καινούργια λέξη, μπορεί να χρειαστεί να εξάγει πληροφορία σχετικά με κάποιο ρήμα σε περίπτωση που αυτό είναι η έξοδος που έχει βρει το **Lstm**. Για παράδειγμα μπορεί να εξάγει στην έξοδο πληροφορία για το αν η λέξη είναι στον πληθυντικό ή στον ενικό, ώστε να ξέρουμε σε τι μορφή πρέπει να συζευχτεί κάποιο ρήμα σχετικά με τις λέξεις που ακολουθούν.

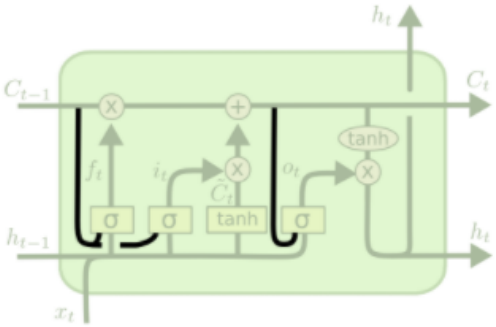


$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Εικόνα 24 Output layer

2.4.5 Παραλλαγές LSTM

Η παραπάνω περιγραφή αντιστοιχεί στην συνηθισμένη μορφή του **LSTM**. Συνήθως όμως το **LSTM** προσαρμόζεται με παραλλαγές ανάλογα με το πρόβλημα που έχουμε να λύσουμε. Μια γνωστή παραλλαγή που παρουσιάστηκε από τους Gers & Schmidhuber (2000), είναι η πρόσθεση ενδιάμεσων συνδέσεων (**peephole connections**). Αυτό σημαίνει πως αφήνουμε τα στρώματα κάθε πύλης να βλέπουν την κατάσταση κελύφους.

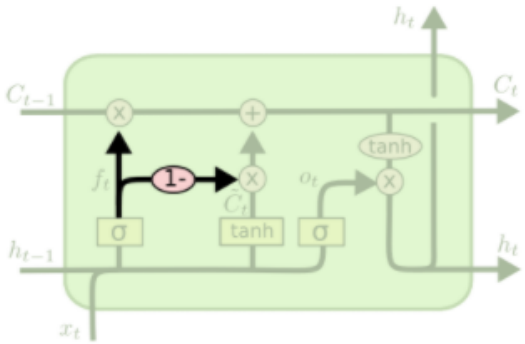


$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] + b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] + b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] + b_o\right)$$

Εικόνα 25 Forget gate, input gate, output gate και ενδιάμεσες συνδέσεις μεταξύ τους

Η παραπάνω εικόνα προσθέτει ενδιάμεσες συνδέσεις σε όλες τις πύλες.

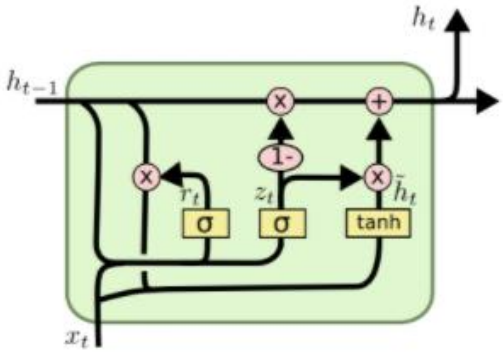
Μια άλλη παραλλαγή είναι αυτή που χρησιμοποιεί σε συνδυασμό πύλες λησμόνησης και εισόδου. Αντί να αποφασίζει ξεχωριστά τι να ξεχάσει και τι νέα πληροφορία να προσθέσει, παίρνει αυτές τις αποφάσεις μαζί. Ξεχνάει μόνο όταν υπάρχει κάποια είσοδος. Εισάγουμε νέες τιμές στην κατάσταση μόνο όταν ξεχνάμε κάποιες παλιότερες.



$$C_t = f_t * C_{t-1} + (1 - f_t) * \tilde{C}_t$$

Εικόνα 26 Λησμόνηση τιμών μόνο όταν υπάρχει είσοδος και εισαγωγή νέων τιμών μόνο όταν ξεχνάμε παλιότερες

Μια ελαφρώς πιο διαφορετική παραλλαγή του **LSTM** είναι αυτή της **Gated Recurrent Unit**, ή **GRU** που παρουσιάστηκε από τον **Cho** (2014). Συνδυάζει τις πύλες λησμόνησης και τις πύλες εισόδου σε μια πύλη ενημέρωσης. Επίσης συγχωνεύει την κατάσταση κελύφους και την κρυφή κατάσταση (**hidden state**). Το τελικό μοντέλο είναι απλούστερο από το κλασσικό **LSTM** και η χρήση του έχει παρουσιάσει άνοδο.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Εικόνα 27 GRU παραλλαγή του LSTM με συνδυασμό forget gate και input gate σε πύλη ενημέρωσης

Κεφάλαιο 3

3.1 Προτεινόμενη προσέγγιση

Το **πρώτο βήμα** που πρέπει να πραγματοποιηθεί για να καταλήξουμε στην κατασκευή ενός βέλτιστου μοντέλου LSTM είναι αυτό της επεξεργασίας των δεδομένων που έχουμε για την εξαγωγή προβλέψεων. Για να εξάγουμε προβλέψεις από το **event log** προτείνουμε αρχικά από **XES** αρχείο να το μετατρέψουμε σε μορφή **txt**. Παίρνουμε όλα τα **events** κάθε ίχνους , τα εκχωρούμε στο **txt** αρχείο μας και στην συνέχεια κρατάμε μόνο τα ακρωνύμια κάθε γεγονότος για την εξαγωγή προβλέψεων με μεγαλύτερη ταχύτητα. Τα **events** αυτά είναι τα δείγματα που θα χρησιμοποιήσει ο αλγόριθμος μας για να εκπαιδευτεί και να είναι ικανός στην συνέχεια να κάνει εξαγωγή προβλέψεων για ακολουθίες γεγονότων πάνω στις οποίες δεν έχει εκπαιδευτεί.

1. Αρχείο XES
2. For event in trace do
3. εκχώρησε event στο txt file
4. Μετέτρεψε τα ονόματα των events στα ακρωνύμια τους

Στο **δεύτερο βήμα** προτείνουμε να καταλήξουμε σε κάποια συμπεράσματα για το **event log** μας και για το κατά πόσο το προβλεπτικό μοντέλο που θέλουμε να κατασκευάσουμε θα κάνει αξιόπιστες προβλέψεις, μέσω του πειραματισμού με κλασσικούς αλγόριθμους εξόρυξης διεργασιών. Από τα μοντέλα διεργασιών που θα κατασκευάσουν οι αλγόριθμοι μας καταλήγουμε σε συμπεράσματα ως προς την αξιοπιστία των δεδομένων μας. Για παράδειγμα μέσω μετρικών που μπορούμε να πάρουμε για κάθε εξαχθέν μοντέλο, όπως το **precision**, το **fitness** και το **simplicity** καταλήγουμε σε συμπεράσματα για το αν τα δεδομένα μας είναι πολύπλοκα (πολλά διαφορετικά δείγματα) και για το αν υπάρχει θόρυβος σε αυτά. Επίσης μέσω αυτών των μετρικών μπορούμε να βγάλουμε ένα αρχικό συμπέρασμα για το πόσο πολύπλοκο να κάνουμε το LSTM μοντέλο μας (πόσα Layers, Neurons κ.τ.λ.). Αν έχουμε για παράδειγμα μεγάλο simplicity είναι περισσότερο πιθανό να μην χρειάζεται να κατασκευάσουμε κάποιο μοντέλο με πολλά στρώματα και νευρώνες.

Στο **τρίτο βήμα** μέσω πειραμάτων με διάφορα χαρακτηριστικά του LSTM μοντέλου κάθε φορά αναδιοργανώνουμε και ανακατασκευάζουμε την προσέγγιση μας ώστε τελικά να καταλήξουμε σε ένα μοντέλο που θα έχει την βέλτιστη προβλεπτική ικανότητα.

3.1.1 Process mining algorithms

3.1.1.1 Alpha miner

Έστω L ένα αρχείο καταγραφής συμβάντων και $T \subseteq \mathcal{A}$. Το $\alpha(L)$ ορίζεται ως εξής:

1) $T_L = \{t \in T \mid \exists_{\sigma \in L} t \in \sigma\}$

2) $T_I = \{t \in T \mid \exists_{\sigma \in L} t = first(\sigma)\}$

3) $T_O = \{t \in T \mid \exists_{\sigma \in L} t = last(\sigma)\}$

4) $X_L = \{(A, B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge \forall_{\alpha \in A} \forall_{b \in B}\ \alpha \rightarrow_L b \wedge \forall_{\alpha 1,2 \in A}\ \alpha 1 \#_L \alpha 2 \wedge \forall_{b1,b2 \in B}\ b1 \#_L b2\}$

5) $Y_L = \{(A, B) \in X_L \mid \forall_{(A',B') \in XL}\ A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B')\}$

6) $P_L = \{_{(A,)} \mid (A, B) \in Y_L\} \cup \{i_L , o_L\}$

7) $F_L = \{(\alpha, {}_{(A,)}) \mid (A, B) \in Y_L \wedge \alpha \in A\} \cup \{(p_{(A,B)} , b) \mid (A, B) \in Y_L \wedge b \in B\} \cup \{(i_L , t) \mid t \in T_I\} \cup \{(t, o_L) \mid t \in T_O\}$

8) $(L) = (P_L , T_L , F_L)$

Έστω L ένα **event log** από ένα σύνολο T, όπου T το σύνολο των δραστηριοτήτων. Στο πρώτο βήμα του αλγόριθμου, ελέγχουμε ποιες δραστηριότητες εμφανίζονται στο **event log** (T_L). Αυτές οι δραστηριότητες θα αντιστοιχούν στις μεταβάσεις του μοντέλου που θα παραχθεί. T_I είναι το σύνολο των αρχικών δραστηριοτήτων και T_O είναι το σύνολο των τελικών δραστηριοτήτων. Οι δραστηριότητες δηλαδή, που εμφανίζονται είτε πρώτες είτε τελευταίες στο αρχείο καταγραφής συμβάντων. Τα βήματα τέσσερα και πέντε αποτελούν τον πυρήνα του αλγόριθμου άλφα. Η πρόκληση είναι να καθοριστούν οι τόποι του μοντέλου μας όπως και οι συνδέσεις τους. Στόχος μας είναι να κατασκευάσουμε τους τόπους που τους συμβολίζουμε με $p_{(A,B)}$ έτσι ώστε το A να είναι το σύνολο των μεταβάσεων εισόδου ($\bullet\ p_{(A,B)} = A$) και το B να είναι το σύνολο των μεταβάσεων εξόδου ($p(A,B)\ \bullet = B$) των $p_{(A,B)}$. Για να είμαστε και πιο συγκεκριμένοι, στο βήμα τέσσερα προσπαθούμε να βρούμε τα ζεύγη (A, B) των συνόλων των δραστηριοτήτων τέτοια ώστε κάθε στοιχείο $a \in A$ και κάθε στοιχείο $b \in B$ να έχουν μια συναφή εξάρτηση (δηλαδή $a \rightarrow_L b$) και όλα τα στοιχεία του A να είναι ανεξάρτητα ($a1 \#_L a2$) όπως και όλα τα στοιχεία του B να είναι ανεξάρτητα ($b1 \#_L b2$). Στο πέμπτο βήμα, κατασκευάζουμε ένα σύνολο στοιχείων Y_L το οποίο διατηρεί μόνο τα μέγιστα στοιχεία του συνόλου X_L . Στο βήμα έξι, το P_L είναι ένα σύνολο από τόπους, και έτσι κρατάμε το σύνολο των τόπων που βρήκαμε στο βήμα πέντε και προσθέτουμε όλες τις μεταβάσεις εισόδου και εξόδου. Στο βήμα επτά τέλος, προσθέτουμε τα βέλη, καθορίζουμε κυρίως τη σχέση ροής ενώνοντας κάθε τόπο $p_{(A,B)}$ με κάθε στοιχείο του $a \in A$, πηγαίες μεταβάσεις (**source transitions**), με κάθε στοιχείο του $b \in B$, μεταβάσεις στόχοι (**target transitions**). Επιπλέον, ο αλγόριθμος άλφα παράγει ένα βέλος από κάθε πηγαίο τόπο (**source place**) i_L σε κάθε αρχική μετάβαση $t \in T_i$ και ένα βέλος από κάθε τελική μετάβαση $t \in T_O$ στον τόπο O_L . Το βήμα οκτώ είναι το αποτέλεσμα όπου είναι ένα Petri-net $a(L)$, και P_L είναι οι τόποι, T_L οι μεταβάσεις και F_L τα βέλη, που περιγράφει την δραστηριότητα που είναι καταγεγραμμένη στο αντίστοιχο **event log**.

Αν και ο αλγόριθμος άλφα έχει τη δυνατότητα να ανακαλύπτει μοντέλα από **event logs**, ο αλγόριθμος αυτός κρύβει και πολλά μειονεκτήματα. Υπάρχουν πολλά διαφορετικά μοντέλα τα οποία έχουν ακριβώς την ίδια συμπεριφορά, δηλαδή δυο μοντέλα μπορεί να έχουν διαφορετική δομή αλλά στην πραγματικότητα να αναπαριστούν το ίδιο ίχνος. Αυτό συμβαίνει επειδή υπάρχουν διάφορα είδη αλγορίθμων που χρησιμοποιούν τις ίδιες γλώσσες μοντελοποίησης. Έτσι μπορεί να παραχθούν δύο διαφορετικά Petri-nets από το ίδιο **event log**. Είναι πιθανό δηλαδή ένα από τα μοντέλα που θα παραχθούν να είναι άσκοπα περίπλοκο με αποτέλεσμα να αναπαριστά περισσότερη συμπεριφορά χωρίς αυτό να είναι αναγκαίο. Αυτό

συμβαίνει επειδή από ένα αρχείο καταγραφής συμβάντων μπορούν να παραχθούν πολλά και διαφορετικά μοντέλα, ανάλογα με τους αλγόριθμους που χρησιμοποιούμε, που να εξηγούν την συμπεριφορά του. Οι τέσσερεις διαστάσεις ποιότητας μας βοηθούν να καταλάβουμε ποιο από αυτά τα μοντέλα είναι το καταλληλότερο και ακριβέστερο για την κάθε περίπτωση [26].

3.1.1.2 Heuristic miner

Θεωρούμε W **event log** για το T και α, b ∈ T:

- $|\alpha >_W b|$ είναι οι φορές που το $\alpha >_W b$ συμβαίνει στο W
- $\alpha =>_W b = (|\alpha >_W b| - | b >_W \alpha|)/(|\alpha >_W b| + | b >_W \alpha| + 1)$

Ο Heuristic miner αλγόριθμος κάνει εξόρυξη του **control-flow** ενός **process model**. Για να το κάνει αυτό λαμβάνει υπόψιν του μόνο την σειρά των **events** σε ένα **case**. Με άλλα λόγια, η σειρά των **events** μεταξύ των **cases** δεν είναι σημαντική. Για να υπολογίζουμε την σειρά των **events** μέσα σε ένα **case** χρησιμοποιούμε το **timestamp**. Με βάση τα παραπάνω μπορούμε να ορίσουμε ένα **event log** ως ακολούθως. Θεωρούμε T το set των δραστηριοτήτων, το $\sigma \in T^*$ είναι ένα **event trace**, δηλαδή μια ακολουθία από αναγνωριστικά δραστηριοτήτων. Το $W \subseteq T^*$ είναι **event log**. Να σημειωθεί πως αφού το W είναι **event log** κάθε **event trace** μπορεί να εμφανιστεί περισσότερες από μια φορά σε ένα **log**. Στα πρακτικά **mining tools** οι συχνότητες κατέχουν σημαντικό ρόλο. Για να βρούμε ένα μοντέλο διεργασιών που βασίζεται σε ένα **event log**, το **log** πρέπει να αναλυθεί ως προς τις συσχετίσεις του (**casual dependencies**). Για παράδειγμα αν μια δραστηριότητα ακολουθείται πάντα από μια άλλη, είναι πολύ πιθανό ότι υπάρχει συσχέτιση (**dependency relation**) μεταξύ των δύο δραστηριοτήτων. Για να αναλύσουμε αυτές τις σχέσεις χρησιμοποιούμε την παρακάτω σημειογραφία. Θεωρούμε W ως **event log** στο T:

$W \subseteq T^*$. Επίσης α, b ∈ T:

1) $\alpha >_W b$ αν υπάρχει trace $\sigma = t_1t_2t_3 \ldots t_n$ and $i \in \{1, \ldots, n-1\} : \sigma \in W$ and $t_i = \alpha$ and $t_{i+1} = b$

2) $\alpha \rightarrow_W b$ αν $\alpha >_W b$ και $b />_W \alpha$

3) $\alpha \#_W b$ αν $\alpha />_W b$ και $b />_W \alpha$, και

4) $\alpha ||_W b$ αν $\alpha >_W b$ και $b >_W \alpha$

5) $\alpha >>_W b$ αν υπάρχει trace $\sigma = t_1t_2t_3 \ldots t_n$ και $i \in \{1, \ldots, n-2\} : \sigma \in W$ και $t_i = \alpha$ και $t_{i+1} = b$ και $t_{i+2} = \alpha$

6) $a >>>_W b$ αν υπάρχει trace $\sigma = t_1t_2t_3 \ldots t_n$ και $i < j$ και $i, j \in \{1, \ldots, n\} : \sigma \in W$ και $t_i = \alpha$ και $t_j = b$

Ας θεωρήσουμε το **event log** W = {ABCD, ABCD, ACBD, ACBD, AED}. Η πρώτη συσχέτιση $>_W$ περιγράφει ποιες δραστηριότητες εμφανίστηκαν σε ακολουθία (η μια δραστηριότητα να έπεται κάποιας άλλης). Μπορούμε εύκολα να διαπιστώσουμε πως, $A >_W B$, $A >_W C$, $A >_W E$, $B >_W C$, $B >_W D$, $C >_W B$, C

$>_W$ D και E $>_W$ D. Η σχέση $\rightarrow_W$ μπορεί να υπολογιστεί από το $>_W$ και αναφερόμαστε σε αυτή ως την (ευθύ) εξαρτώμενη σχέση που εξάγεται από το **event log** W. Ισχύει A $\rightarrow_W$ B, A $\rightarrow_W$ C, A $\rightarrow_W$ E, B $\rightarrow_W$ D, C $\rightarrow_W$ D, και E $\rightarrow_W$ D. Να σημειωθεί πως B $/\rightarrow_W$ C διότι ισχύει C $>_W$ B. Η σχέση $||_W$ υποδεικνύει ταυτόχρονη συμπεριφορά, δηλαδή B$||_W$ C και C$||_W$ B. Άμα δύο δραστηριότητες έπονται η μια της άλλης απευθείας σε οποιαδήποτε σειρά, τότε όλες οι πιθανές αλληλεπιδράσεις, συνδυασμοί πραγματοποιούνται, κατά συνέπεια υπάρχει πιθανότητα να είναι παράλληλες. Η σχέση $\#_W$ δίνει ζευγάρια μεταβάσεων, οι οποίες δεν ακολουθούν οι μια την άλλη απευθείας. Αυτό σημαίνει πως δεν υπάρχει απευθείας εξαρτώμενες σχέσεις μεταξύ τους και δεν είναι πιθανό το να είναι παράλληλες. Σε αντίθεση με τον **Heuristics miner,** άμα πάρουμε για παράδειγμα την προσέγγιση του **Alpha miner** αυτές οι τρείς βασικές συσχετίσεις (A $\rightarrow_W$ B, A$\#_W$ B, ή A$||_W$ B) χρησιμοποιούνται για την κατασκευή του **Petri net**. Ακόμα αυτή η προσέγγιση προϋποθέτει άψογη πληροφόρηση 1) το **log** πρέπει να είναι άρτια ολοκληρωμένο (για παράδειγμα αν μια δραστηριότητα έπεται απευθείας μιας άλλης δραστηριότητας, αυτή η συμπεριφορά πρέπει να περιέχεται στο **log**) και 3) δεν υπάρχει θόρυβος στο **log** (δηλαδή όλες οι εγγραφές μέσα στο **log** είναι σωστές). Επιπρόσθετα ο συγκεκριμένος αλγόριθμος δεν λαμβάνει υπόψιν του την συχνότητα των **traces** στο **log**. Επιπλέον σε ορισμένες περιπτώσεις σπάνια τα **logs** δεν θα περιέχουν θόρυβο. Ειδικά η διαφορά μεταξύ **errors**, δραστηριότητες που έχουν χαμηλή συχνότητα, ακολουθίες δραστηριοτήτων που έχουν χαμηλή συχνότητα αλλά και οι διάφορες εξαιρέσεις δημιουργούν προβλήματα και δυσλειτουργίες. Συμπερασματικά στην πραγματικότητα, είναι περισσότερο δύσκολο το να αποφασιστεί αν για παράδειγμα μεταξύ δύο δραστηριοτήτων (A και B), κάποια από τις τρείς σχέσεις (A $\rightarrow_W$ B, A$\#_W$ B, ή A$||_W$ B) ισχύει. Για παράδειγμα η σχέση εξάρτησης όπως χρησιμοποιείται στον **Alpha miner** (A $\rightarrow_W$ B) ισχύει μόνο αν στο **log** υπάρχει **trace** στο οποίο το A ακολουθείται απευθείας από το B (δηλαδή υπάρχει η σχέση A $>_W$ B) και δεν υπάρχει **trace** στο οποίο το B ακολουθείται απευθείας από το A (δηλαδή δεν υπάρχει η σχέση B $>_W$ A). Παρόλαυτα σε μια περίπτωση που υπάρχει θόρυβος ένα εσφαλμένο παράδειγμα μπορεί να καταστρέψει εντελώς την εξαγωγή του σωστού αποτελέσματος. Ακόμα και αν έχουμε χιλιάδες **log traces** όπου το A ακολουθείται απευθείας από το B, τότε το B $>_W$ A παράδειγμα που είναι βασισμένο σε εσφαλμένη εγγραφή θα αποτρέψει την εξαγωγή του σωστού αποτελέσματος. Όπως σημειώθηκε προηγουμένως, η πληροφόρηση σχετικά με την συχνότητα δεν χρησιμοποιείται σε αυτή την προσέγγιση. Για αυτό τον λόγο στον **Heuristics Miner** χρησιμοποιούνται τεχνικές που είναι λιγότερο ευαίσθητες στον θόρυβο και στα ελλιπή **logs**. Η κύρια ιδέα του αλγορίθμου είναι να λαμβάνει υπόψιν την συχνότητα των ακόλουθων σχέσεων και να συμπεραίνει τις συμπληρωματικές (A $\rightarrow_W$ B, A$\#_W$ B, ή A$||_W$ B) [29]

3.1.1.3 Inductive miner

Ο **inductive miner** είναι ένα από τα περισσότερο ευρέως χρησιμοποιούμενα εργαλεία για την εξόρυξη διεργασιών λόγω τις προσαρμοστικότητας και της

επεκτασιμότητας που τον χαρακτηρίζουν. Κάποια από τα πολλά πλεονεκτήματα του συγκεκριμένου αλγορίθμου είναι τα παρακάτω:

- Όλα τα εξορυγμένα μοντέλα ανταποκρίνονται στον θόρυβο
- Το μοντέλο πάντα ταιριάζει στο **event log**, για παράδειγμα το μοντέλο ανταποκρίνεται στα **traces** του **log**.
- Το μοντέλο μπορεί να χειριστεί μη συνηθισμένη συμπεριφορά, δηλαδή **events** που εμφανίζονται σπάνια και να διαχειριστεί **logs** μεγάλου μεγέθους διασφαλίζοντας ταυτόχρονα με την χρήση κριτηρίων ορθότητας (**formal correctness criteria**) ενέργειες όπως για παράδειγμα την ανακάλυψη του πρωτότυπου μοντέλου
- είναι ιδιαίτερα επεκτάσιμος και επιτρέπει την χρήση διαφόρων παραμέτρων
- τα αποτελέσματα που επιστρέφει μπορούν εύκολα να χρησιμοποιηθούν από άλλες σημειογραφίες όπως τα **Petri nets** και τα **BPMN**

Ο **inductive miner** λειτουργεί αναδρομικά με την χρήση της στρατηγικής **διαίρει και βασίλευε**: διαχωρίζει το **log**, κατασκευάζει μέρος του δέντρου διεργασιών (**process tree**) και συνεχίζει με τον επιπλέον διαχωρισμό των προηγουμένως διαχωριζόμενων μερών του **log**. Το δέντρο διεργασιών μπορεί να μετατραπεί σε **Petri Net**. Επιπροσθέτως το **ProM** προφέρει ένα **plugin** που μπορεί να εξορύξει το **Petri Net** απευθείας με την χρήση του αλγορίθμου **inductive miner**.

Για παράδειγμα αν έχουμε τα ακόλουθα **traces** σε ένα **event log**:

<a,b,c,d,e,g>
<a,b,c,d,f,g>
<a,c,d,b,f,g>
<a,b,d,c,c,g>
<a,d,c,b,f,g>

Εικόνα 28 Traces πριν τον διαχωρισμό από τον Inductive miner

Ο **inductive miner** διαχωρίζει το **log** ως ακολούθως, διότι προφανώς, όπως παρατηρούμε κάθε **trace** αρχίζει με την δραστηριότητα a, συνεχίζει με κάποιον συνδυασμό δραστηριοτήτων και πάντοτε τελειώνει με την δραστηριότητα g:

<a | b,c,d,e | g>
<a | b,c,d,f | g>
<a | c,d,b,f | g>
<a | b,d,c,c | g>
<a | d,c,b,f | g>

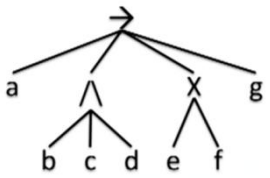
Εικόνα 29 Διαχωρισμός των traces

Το πρώτο επίπεδο του δέντρου διεργασιών έχει κατασκευαστεί. Στην συνέχεια ο **miner** διαχωρίζει τα εναπομείναντα κομμάτια του **log** τα οποία είναι τα παρακάτω:

b,c,d | e
b,c,d | f
c,d,b | f
b,d,c | c
d,c,b | f

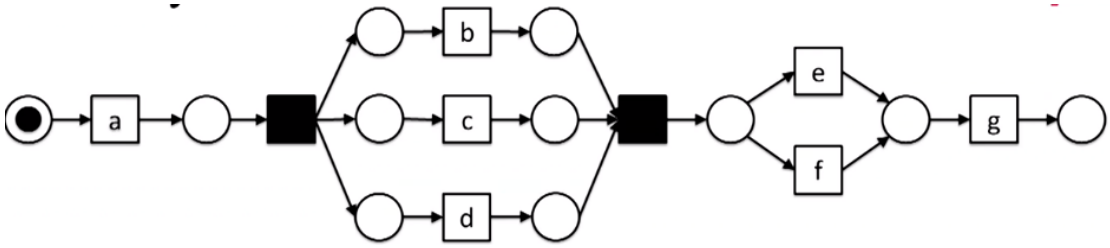
Εικόνα 30 Διαχωρισμός των traces

Στα αριστερά οι δραστηριότητες b, c, d συμβαίνουν σε διαφορετική σειρά κάθε φορά, δηλαδή συμβαίνουν παράλληλα και στο δεξί μέρος συμβαίνουν είτε η δραστηριότητα e είτε η δραστηριότητα f. Συμπερασματικά το ακόλουθο δέντρο διεργασιών, αποτυπώνει το **event log**:



Εικόνα 31 Δέντρο διεργασιών που αποτυπώνει το event log

Στην εικόνα που ακολουθεί βλέπουμε το **Petri Net**.



Εικόνα 32 Petri net

3.1.2 Μέγεθος του dataset

Αρχικά μπορούμε να κάνουμε πειράματα με το μέγεθος του **dataset** άμα έχουμε μεγάλο **event log**.

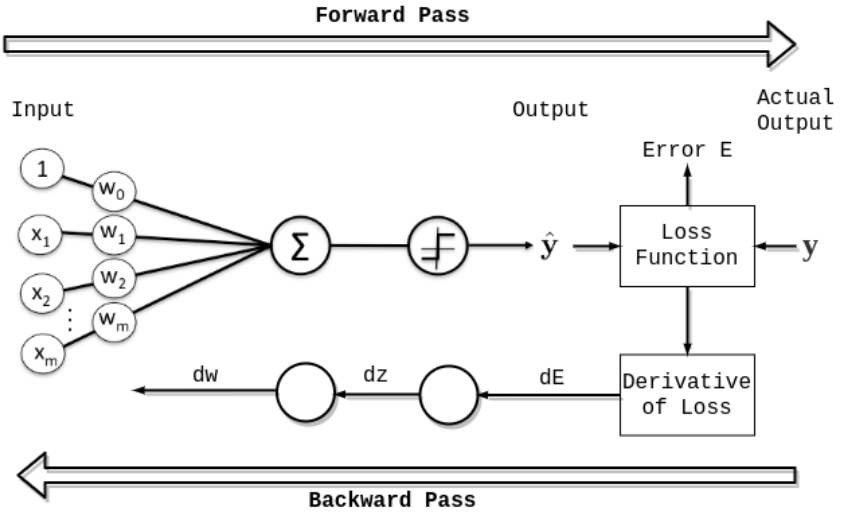
Η ποσότητα των δεδομένων είναι τις περισσότερες φορές ο πιο βασικός παράγοντας που καθορίζει την απόδοση του μοντέλου μας. Η ποσότητα που πρέπει να πάρουμε εξαρτάται από το τι θέλουμε να προβλέψουμε, το **accuracy** που θέλουμε να επιτύχουμε, τα **input features** που έχουμε, τον θόρυβο των **training data**, τον θόρυβο στα εξαχθέντα **features**, την πολυπλοκότητα του προβλήματος μας άλλα και από την πολυπλοκότητα του επιλεγμένου αλγορίθμου. Άρα ο τρόπος για να βρούμε το πόσα δεδομένα πρέπει να χρησιμοποιήσουμε πρέπει να κάνουμε δοκιμές με διάφορα μεγέθη για τα **training data**. Στο σύνολο των περιπτώσεων όταν το **dataset** μας είναι πολύ μικρού μεγέθους (μερικών εκατοντάδων δειγμάτων) κάνουμε **overfit** το μοντέλο μας διότι προσπαθεί να μάθει βάση ελαχίστων δειγμάτων τα οποία δεν έχουν μεγάλη διαφοροποίηση. Όταν αυξάνουμε το μέγεθος του **dataset** συνήθως μειώνουμε το **overfitting**.

- Πολυπλοκότητα προβλήματος: Η άγνωστη συνάρτηση που συσχετίζει καλύτερα τα **input variables** με τα **output variables**.
- Πολυπλοκότητα του **learning algorithm**: Ο αλγόριθμος που χρησιμοποιείται για την εκμάθηση του **mapping function** από συγκεκριμένα παραδείγματα.

3.1.3 Αριθμός των epochs

3.1.3.1 Τι είναι το epoch;

Ένα **epoch** σημαίνει ότι κάνουμε **train** το νευρωνικό μας δίκτυο με όλα τα **training data** για ένα κύκλο. Σε ένα **epoch** χρησιμοποιούμε όλα τα **data** ακριβώς μια φορά. Ένα **forward pass** και ένα **backward pass** μαζί, θεωρούνται σαν ένα **pass**.



Εικόνα 33 Forward pass και Backward pass των data σε ένα epoch

Ένα **epoch** αποτελείται από ένα ή περισσότερα **batches** όπου χρησιμοποιούμε μέρος του **dataset** για να εκπαιδεύσουμε το νευρωνικό δίκτυο. Ένα **iteration** είναι όταν περνάμε από όλα τα **training examples**.

3.1.3.2 Πως τα epochs επηρεάζουν την απόδοση του μοντέλου μας;

Έχοντας έναν μεγάλο αριθμό **epochs** δεν αυξάνεται απαραίτητα η απόδοση του μοντέλου μας. Ο αριθμός των **epochs** μπορεί να αυξήσει το **accuracy** μέχρι ένα ορισμένο σημείο μετά το οποίο, άμα το ξεπεράσουμε, αρχίζουμε και κάνουμε **overfit** το μοντέλο μας. Ακόμα, έχοντας πολύ μικρό αριθμό **epochs** θα οδηγήσει το μοντέλο μας να κάνει **underfit** τα δεδομένα μας. Ο κατάλληλος αριθμός των **epochs** είναι αυτός στον οποίο το **accuracy** σταματάει να αυξάνεται, συνήθως μεταξύ 1 και 10 **epochs**. Για να καταλάβουμε πότε το **accuracy** σταματάει να αυξάνεται χρησιμοποιούμε την μέθοδο του **early stop** που σταματάει το **training** όταν παρατηρείται το παραπάνω [31],[32].

3.1.4 Batch size

Επιπροσθέτως πρέπει να γίνουν δοκιμές με το μέγεθος του **batch size**. Για μοντέλα όπως το **LSTM** και το **CNN**, το **batch-size** κατέχει μεγάλη σημαντικότητα για την εκμάθηση των χαρακτηριστικών των προτύπων (**patterns**). Για ένα μοντέλο για να είναι ικανό να καταλάβει ποια είναι τα περισσότερο κοινά πρότυπα και χαρακτηριστικά στο σετ εκπαίδευσης (**train set**) πρέπει να του παρέχουμε τα δείγματα σε μορφή ξεχωριστών μερών-ομάδων (**batches**). Παρέχοντας στο μοντέλο μας, ένα σύνολο δειγμάτων με κάθε μέρος (**batch**) του **train set**, που του δίνουμε σαν είσοδο, το μοντέλο είναι ικανό με αυτό τον τρόπο να ξεχωρίσει και να αναγνωρίζει στην συνέχεια, κοινά χαρακτηριστικά, έχοντας σαν σημείο αναφοράς τα

δείγματα του **batch**. Για παράδειγμα ας υποθέσουμε πως θέλουμε να εκπαιδεύσουμε ένα μοντέλο που μας δίνει 1 αν ένας άνθρωπος βρίσκεται σε μια εικόνα και 0 αν δεν βρίσκεται κάποιος άνθρωπος στην εικόνα. Μπορούμε να εκπαιδεύσουμε το μοντέλο μας παρέχοντας του κάθε φορά μια εικόνα ή μπορούμε να το εκπαιδεύσουμε δίνοντας του σύνολα εικόνων (**batches**) όπου μπορούμε να παρέχουμε σε κάθε κύκλο της εκπαίδευσης 10 εικόνες σαν είσοδο στο μοντέλο, ώστε κοιτώντας αυτές τις 10 εικόνες κάθε φορά, το μοντέλο μας να μπορεί να ξεχωρίσει τα κοινά πρότυπα-χαρακτηριστικά (πόδια, κεφάλι κ.λ.π.). Συμπερασματικά είναι σημαντικό να παρέχουμε στο μοντέλο μας το βέλτιστο **batch-size**, ώστε να μπορεί να μάθει τα χαρακτηριστικά των δειγμάτων κατά την εκπαίδευση.

Ο βέλτιστος αριθμός για το **batch size** φορά πρέπει να προσδιορίζεται μετά από πειράματα και αναλόγως το μέγεθος του dataset που χρησιμοποιείται. Για παράδειγμα για ένα dataset μερικών εκατοντάδων δειγμάτων ένα **batch-size** ίσο με 32 αρκεί. Άλλα συνηθισμένα μεγέθη για το **batch-size** είναι 64, 128, 256, 512. Σε προβλήματα πρόβλεψης ακολουθιών, συχνά είναι προτιμότερο να χρησιμοποιούμε μεγάλο **batch-size** κατά την εκπαίδευση.

Τα νευρωνικά δίκτυα εκπαιδεύονται χρησιμοποιώντας τον **stochastic gradient descent optimization** αλγόριθμο. Αυτό περιλαμβάνει την χρήση της τωρινής κατάστασης του μοντέλου για την εξαγωγή κάποιας πρόβλεψης, την σύγκριση της πρόβλεψης με την αναμενόμενη τιμή και την χρήση της διαφοράς τους για τον υπολογισμό του σφάλματος (**error gradient**). Το **error gradient** χρησιμοποιείται για να ενημερωθούν τα βάρη του μοντέλου και η παραπάνω διαδικασία επαναλαμβάνεται. Το **error gradient** είναι μια στατιστική εκτίμηση. Τις περισσότερες φορές όσο περισσότερα δείγματα χρησιμοποιήσουμε για την εκτίμηση, τόσο περισσότερο ακριβής είναι και είναι πιθανότερο τα βάρη του νευρωνικού να προσαρμοστούν έτσι, ώστε να βελτιώσουν την απόδοση του μοντέλου. Για να πάρουμε αυτή την βελτιωμένη εκτίμηση του **error gradient** και να ενημερωθούν τα βάρη, πρέπει να χρησιμοποιήσουμε το μοντέλο μας για να κάνουμε προβλέψεις αρκετές φορές πριν οδηγηθούμε σε αυτό το αποτέλεσμα.

Αντιθέτως χρησιμοποιώντας λιγότερα δείγματα οδηγεί, πολλές φορές σε λιγότερο ακριβή εκτίμηση του **error gradient** που είναι άμεσα εξαρτώμενο από αυτά. Αυτό οδηγεί σε μια εκτίμηση με θόρυβο που με την σειρά της οδηγεί σε επανεκτιμήσεις, των βαρών του μοντέλου, με θόρυβο. Παρόλαυτα, αυτές οι ‘’θορυβώδεις’’ επανεκτιμήσεις, μπορεί να οδηγήσουν κάποιες φορές σε μοντέλα που μαθαίνουν τα χαρακτηριστικά των δειγμάτων εκπαίδευσης γρηγορότερα. Ο αριθμός των δειγμάτων εκπαίδευσης που χρησιμοποιούνται για την εκτίμηση του **error gradient** είναι μια υπερπαράμετρος (**hyperparameter**) του αλγορίθμου εκμάθησης που καλείται **batch-size** ή **batch**.

Όταν δηλαδή έχουμε θέσει το **batch-size** ίσο, για παράδειγμα με 32, εννοούμε πως θα χρησιμοποιηθούν 32 δείγματα για την εκτίμηση του **error-gradient** πριν επανεκτιμηθούν τα βάρη. Ένας κύκλος εκπαίδευσης (**training epoch**) σημαίνει πως ο αλγόριθμος εκμάθησης έχει περάσει μία φορά από το **training dataset**, όπου τα δείγματα ήταν χωρισμένα, τυχαία σε ομάδες ίσες με το **batch-size**.

Άμα θέσουμε το **batch-size** σε οποιαδήποτε αριθμό μεγαλύτερο του 1 και μικρότερο από τον αριθμό των δειγμάτων στο σετ εκπαίδευσης , τότε ο αλγόριθμος εκμάθησης καλείται **minibatch gradient descent**. Αν το **batch-size** είναι ίσο με 1 τότε ο αλγόριθμος καλείται **stochastic gradient descent** και αν είναι ίσο με τον αριθμό των δειγμάτων στο σετ εκπαίδευσης τότε καλείται **batch gradient descent**.

Τα μικρά **batch-sizes** χρησιμοποιούνται για 2 κύριους λόγους:

- Έχουν θόρυβο και κάνουν κανονικοποίηση και προσφέρουν μικρότερο σφάλμα γενίκευσης (**generalization error**)
- Είναι καλύτερα για την μνήμη

3.1.5 Πλήθος των layers

Ακόμα πρέπει να γίνει πειραματισμός με τον αριθμό των στρωμάτων του μοντέλου που θέλουμε να κατασκευάσουμε. Η επιλογή των **hidden layers** έχει να κάνει με το αν τα δεδομένα μας είναι γραμμικώς διαχωρίσιμα. Στην περίπτωση που είναι δεν χρειαζόμαστε ενδιάμεσα στρώματα. Συνήθως συναντάμε σπάνια προβλήματα που απαιτούν δύο ενδιάμεσα στρώματα. Παρόλαυτα νευρωνικά δίκτυα με δύο ενδιάμεσα στρώματα μπορούν να αναπαραστήσουν συναρτήσεις σε οποιαδήποτε σχήμα. Θεωρητικά δεν υπάρχει κάποιος λόγος να χρησιμοποιούμε νευρωνικά δίκτυα με παραπάνω από δύο ενδιάμεσα στρώματα. Παρακάτω βλέπουμε έναν πίνακα που δείχνει περιληπτικά τις δυνατότητες των αρχιτεκτονικών των νευρωνικών δικτύων ανάλογα με των αριθμό των ενδιάμεσων στρωμάτων.

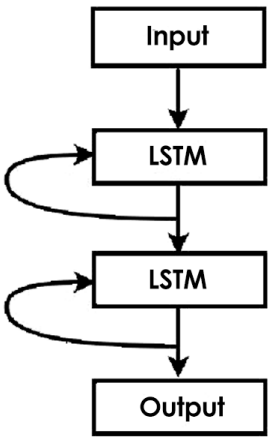
Αριθμός ενδιάμεσων στρωμάτων	**Αποτέλεσμα**
0	Ικανό να αναπαραστήσει μόνο γραμμικά διαχωρίσιμες συναρτήσεις
1	Μπορεί να προσεγγίσει οποιαδήποτε συνάρτηση που περιέχει συνεχή αντιστοίχηση από τον έναν πεπερασμένο χώρο στον άλλον
2	Μπορεί να αναπαραστήσει οποιαδήποτε αυθαίρετο όριο απόφασης σε αυθαίρετη ακρίβεια (υπολογισμός της ακρίβειας ανάλογα με την διαθέσιμη μνήμη) με την χρήση λογικών συναρτήσεων ενεργοποίησης και είναι ικανό να προσεγγίσει την συνάρτηση ομαλότητας της ακρίβειας

Όσον αφορά την μέτρηση των **layers** υπάρχουν αντικρουόμενες απόψεις. Οι διαφωνίες έχουν να κάνουνε με το αν πρέπει να λαμβάνουμε υπόψιν το στρώμα εισόδου. Αυτή η άποψη υποστηρίζεται από το επιχείρημα πως, ο λόγος που δεν πρέπει να μετράμε το **input layer** είναι γιατί οι είσοδοι (οι νευρώνες) δεν είναι ενεργοποιημένες, είναι απλά οι μεταβλητές εισόδου. Δηλαδή σύμφωνα με την παραπάνω άποψη ένα νευρωνικό δίκτυο με ένα **input layer**, ένα **hidden layer** και ένα **output layer**, είναι ένα νευρωνικό δίκτυο με δύο **layers**.

Ο λόγος που πολλές φορές χρησιμοποιούμε, για την επίλυση ορισμένων προβλημάτων παραπάνω από ένα **layer**, είναι όπως αναφέραμε και παραπάνω το

ότι ένα νευρωνικό δίκτυο με ένα **layer** μπορεί να χρησιμοποιηθεί μόνο για την αναπαράσταση γραμμικά διαχωριζόμενων συναρτήσεων. Αυτό σημαίνει πως μπορούμε να λύσουμε μόνο απλά προβλήματα κατηγοριοποίησης όπου για παράδειγμα δύο κλάσεις μπορούν να χωριστούν από μια ευθεία γραμμή. Ένα μοντέλο με παραπάνω από ένα **layer** χρησιμοποιείται για την αναπαράσταση κυρτών περιοχών. Δηλαδή για τον σχεδιασμό σχημάτων γύρω από γνωρίσματα που βρίσκονται σε πολλές διαστάσεις και έτσι να τα διαχωρίσουν και να τα κατηγοριοποιήσουν. Θεωρητικές μελέτες υποδεικνύουν πως μοντέλα με δύο ενδιάμεσα στρώματα είναι επαρκή για την δημιουργία περιοχών κατηγοριοποίησης οποιαδήποτε σχήματος. Ακόμα η επιτυχία των νευρωνικών δικτύων βαθιάς μάθησης έχει να κάνει με το ότι κάθε στρώμα επεξεργάζεται ένα μέρος του προβλήματος που θέλουμε να επιλύσουμε και το περνάει στο επόμενο στρώμα. Τα επιπλέον ενδιάμεσα στρώματα επανασυνθέτουν την αναπαράσταση των δεδομένων εισόδου που έχει ‘’μάθει’’ το προηγούμενο στρώμα και δημιουργούν επιπλέον αναπαραστάσεις με υψηλά επίπεδα αφαίρεσης. Για παράδειγμα από γραμμές, σε σχήματα και τελικά σε αντικείμενα. Ένα υποθετικά μεγάλο ενδιάμεσο στρώμα μπορεί να προσεγγίσει τις περισσότερες συναρτήσεις. Η αύξηση του ‘’βάθους’’ του νευρωνικού προσφέρει μια εναλλακτική λύση όπου έχουμε στρώματα με λιγότερους νευρώνες και γρηγορότερη εκπαίδευση του νευρωνικού.

Αφού τα **Lstm** λειτουργούν με ακολουθιακά δεδομένα, σημαίνει πως η πρόσθεση στρωμάτων προσθέτει επιπλέον επίπεδα αφαίρεσης των παρατηρήσεων εισόδου. Αυτό έχει σαν αποτέλεσμα με το πέρασμα των **epochs** να διαχωρίζουμε το αρχικό μας πρόβλημα σε μικρότερα κομμάτια. Η αρχιτεκτονική **Stacked Lstm** χρησιμοποιείται για την πρόβλεψη δύσκολων ακολουθιακών δεδομένων. Η **Stacked Lstm** αρχιτεκτονική μπορεί να οριστεί σαν ένα **Lstm** μοντέλο που αποτελείται από πολλά στρώματα **Lstm**. Το **Lstm layer** που βρίσκεται από πάνω παρέχει μια ακολουθία σαν έξοδο, η οποία είναι η είσοδος του **Lstm layer** που βρίσκεται από κάτω. Παρακάτω βλέπουμε την αρχιτεκτονική του **Stacked Lstm**.



Εικόνα 34 Stacked LSTM

3.1.6 Activation functions

Επίσης προσοχή πρέπει να δοθεί και στην επιλογή του κατάλληλου **Activation function**. Αυτό που κάνει επιγραμματικά ένας τεχνητός νευρώνας είναι να υπολογίζει το αθροιστικό βάρος, πολλαπλασιάζοντας κάθε τιμή του σετ με το βάρος της και στην συνέχεια προσθέτει τα γινόμενα και τα διαιρεί με το άθροισμα των βαρών (**weighted sum**), προσθέτει κάποια κλίση (**bias**) και στην συνέχεια αποφασίζει αν θα ενεργοποιηθεί ή όχι. Ας υποθέσουμε πως έχουμε έναν νευρώνα που περιγράφεται από τον παρακάτω τύπο:

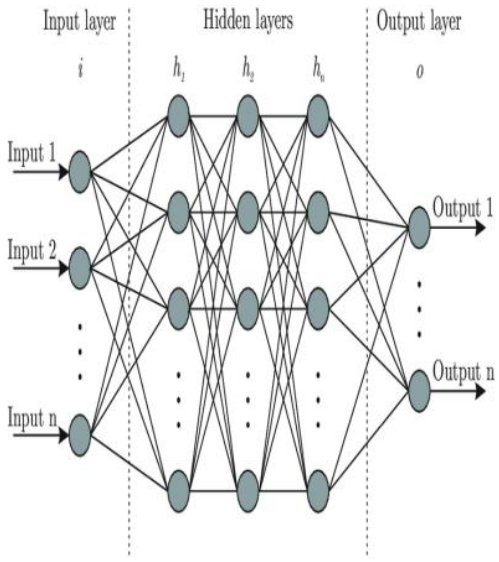
$$Y = \sum (weight * input) + bias$$

Εικόνα 35 Τύπος περιγραφής νευρώνα

Η τιμή του **Y** μπορεί να είναι οτιδήποτε μεταξύ -∞ έως +∞. Ο νευρώνας δεν γνωρίζει στην πραγματικότητα τα όρια αυτής της τιμής. Πως όμως αποφασίζουμε αν ο συγκεκριμένος νευρώνας πρέπει να ενεργοποιηθεί ή όχι; Χρησιμοποιούμε το **activation function** για αυτό τον σκοπό. Για να ελέγχουμε την τιμή του **Y** και να γνωρίζουμε αν οι εξωτερικές συνδέσεις οφείλουν να λαμβάνουν υπόψιν τους αυτόν τον νευρώνα ή όχι.

3.1.6.1 Γιατί τα νευρωνικά δίκτυα χρειάζονται Activation functions;

Ο λόγος χρήσης των **Activation functions** είναι γιατί Τα νευρωνικά δίκτυα είναι δίκτυα στρωμάτων που αποτελούνται από νευρώνες που σχηματίζουν κόμβους που χρησιμοποιούνται για την κατηγοριοποίηση και την πρόβλεψη δεδομένων από τα δεδομένα που έχουμε εκχωρήσει σαν είσοδο. Υπάρχει το **input layer**, ένα ή περισσότερα ενδιάμεσα στρώματα και το στρώμα εξόδου. Όλα τα στρώματα έχουν κόμβους και κάθε κόμβος έχει ένα βάρος που υπολογίζεται κατά την επεξεργασία πληροφοριών από το ένα στρώμα στο επόμενο.



Εικόνα 36 Παράδειγμα Νευρωνικού Δικτύου

Αν κάποιο **activation function** δεν χρησιμοποιηθεί στο νευρωνικό δίκτυο τότε το σήμα εξόδου θα είναι μια απλή γραμμική συνάρτηση που θα έχει πολυωνυμικό βαθμό 1. Παρόλαυτα μια γραμμική εξίσωση είναι απλή και εύκολη ως προς την επίλυση της, αλλά η πολυπλοκότητα της είναι περιορισμένη και δεν έχει την δυνατότητα να ‘’μάθει’’ και να αναγνωρίσει πολύπλοκες συσχετίσεις στα δεδομένα. Τα νευρωνικά δίκτυα που δεν χρησιμοποιούν **activation functions** συμπεριφέρονται σαν μοντέλα γραμμικής παλινδρόμησης με περιορισμένη απόδοση και ισχύ της περισσότερες φορές. Είναι επιθυμητό για ένα νευρωνικό δίκτυο, όχι μόνο να ‘’μαθαίνει’’ και να υπολογίζει γραμμικές συναρτήσεις αλλά να επιλύει πιο πολύπλοκα προβλήματα, όπως η μοντελοποίηση πολύπλοκων τύπων δεδομένων, όπως για παράδειγμα εικόνες, video, ήχος, ομιλία, κείμενο κ.τ.λ. Αυτός είναι ο λόγος χρήσης **activation function** σε τεχνικές όπως αυτή της Βαθιάς Μάθησης που αναπαριστά δεδομένα σε πολλές διαστάσεις, όπου το μοντέλο έχει πολλαπλά ενδιάμεσα στρώματα και πολύπλοκη αρχιτεκτονική για τον στόχο που δεν είναι άλλος από αυτόν της εξαγωγής γνώσης.

3.1.6.2 Η ανάγκη για μη-γραμμικότητα στα Νευρωνικά Δίκτυα

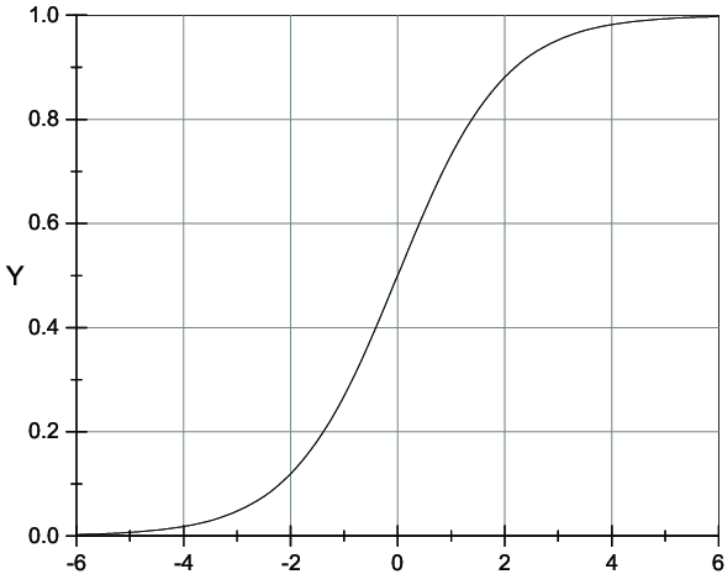
Η συναρτήσεις που έχουν βαθμό μεγαλύτερο του ένα και η γραφική τους παράσταση παρουσιάζει καμπυλότητα ονομάζονται Μη-Γραμμικές συναρτήσεις. Απαιτείται από ένα νευρωνικό δίκτυο να ‘’μάθει’’, να αναπαραστήσει και να επεξεργαστεί οποιαδήποτε δεδομένα και οποιαδήποτε πολύπλοκη συνάρτηση που αντιστοιχεί στην είσοδο και στην έξοδο. Τα νευρωνικά δίκτυα είναι επίσης γνωστά και σαν **Universal Function Approximators** διότι μπορούν να υπολογίσουν και να ‘’μάθουν’’ οποιαδήποτε συνάρτηση που δίνεται σε αυτά. Οποιαδήποτε διεργασία μπορεί να αναπαρασταθεί σαν συναρτησιακός υπολογισμός σε ένα νευρωνικό δίκτυο. Ως εκ τούτου χρειάζεται να εφαρμόζουμε **activation functions** για να κάνουμε το δίκτυο δυναμικό και να έχει την δυνατότητα να εξαγάγει πολύπλοκες πληροφορίες από τα δεδομένα εισόδου και να αναπαραστήσει μη γραμμικές, συνελικτικές, τυχαίες αντιστοιχίσεις μεταξύ της εισόδου και της εξόδου. Με αυτό τον τρόπο, προσθέτοντας μη γραμμικότητα με την βοήθεια μη γραμμικών **activation functions** στο δίκτυο, είμαστε ικανοί να επιτύχουμε μη γραμμικές αντιστοιχίσεις από τα δεδομένα εισόδου στα δεδομένα εξόδου. Ένα σημαντικό χαρακτηριστικό τον **activation functions** είναι το ότι πρέπει να είναι διαφοροποιήσημες ώστε να εκτελούν την στρατηγική της οπισθοδιάδοσης (**back propagation**) ώστε να υπολογιστούν τα σφάλματα των βαρών και τελικά να βελτιστοποιηθούν τα βάρη χρησιμοποιώντας κάποιον αλγόριθμο επαναληπτικής βελτιστοποίησης όπως ο **Gradient Descend**

3.1.6.3 Softmax

Η συνάρτηση Softmax αποτελεί μία γενίκευση της σιγμοειδούς συνάρτησης για μετασχηματισμό τιμών σε πιθανότητες, όταν υπάρχουν παραπάνω από δύο κατηγορίες (κλάσεις). Εφαρμόζεται στο δίανυσμα που προκύπτει από το στρώμα εξόδου και μετασχηματίζει τις τιμές σε πιθανότητες οι οποίες αθροίζουν στη μονάδα. Αν το διάνυσμα εξόδου είναι μήκους 2, η softmax ταυτίζεται με τη σιγμοειδή συνάρτηση. Η συνάρτηση softmax ορίζεται ως εξής:

$$a_i = \frac{e^{z_i}}{\sum_{k=1}^{m} e^{z_k}}$$

Η softmax χρησιμοποιείται ευρέως σε προβλήματα κατηγοριοποίησης, όπου μόνο μία κατηγορία είναι η σωστή. Για παράδειγμα, αν κάθε λέξη του λεξιλογίου θεωρηθεί ως κατηγορία, κατά τη Γλωσσική Μοντελοποίηση, ο στόχος είναι να μεγιστοποιηθεί η πιθανότητα που προκύπτει από τη softmax για την επόμενη λέξη / κατηγορία, δεδομένων όλων των προηγούμενων λέξεων. Ένα μείζον πρόβλημα κατά την εκπαίδευση μοντέλων τα οποία χρησιμοποιούν τη softmax είναι ότι είναι αρκετά χρονοβόρα, λόγω της κανονικοποίησης, ιδιαίτερα όταν εμπλέκονται εκατομμύρια ή δισεκατομμύρια κατηγορίες στα διανύσματα εξόδου.



Εικόνα 37 Διάγραμμα συνάρτησης Softmax

3.1.6.3 Sigmoid

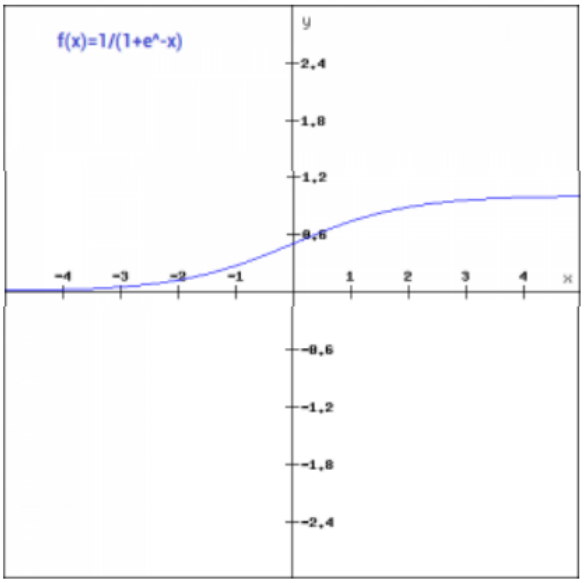
Είναι η περισσότερο ευρέως χρησιμοποιούμενη συνάρτηση ενεργοποίησης λόγω του ότι είναι μη γραμμική συνάρτηση. Η Σιγμοειδής συνάρτηση μετασχηματίζει τι τιμές μεταξύ του 0 και του 1. Μπορεί να οριστεί σαν:

$f(x) = 1/e^{-x}$

Η σιγμοειδής συνάρτηση είναι συνεχώς διαφοροποιήσιμη και μια καμπυλόγραμμη **S-shaped** συνάρτηση. Η παράγωγος της συνάρτησης είναι η παρακάτω:

f’(x) = 1-sigmoid(x)

Επίσης, η σιγμοειδής συνάρτηση δεν είναι συμμετρική στο 0, κάτι το οποίο σημαίνει πως τα πρόσημα όλων των τιμών της εξόδου των νευρώνων θα είναι ίδια. Αυτό το θέμα μπορεί να βελτιωθεί με την κλιμάκωση της σιγμοειδής συνάρτησης.



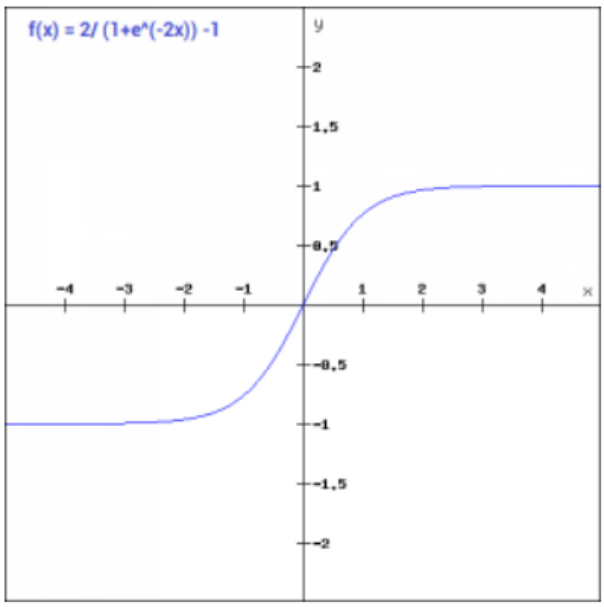
Εικόνα 38 Διάγραμμα Σιγμοειδούς συνάρτησης

3.1.6.4 Tanh

Η **Tanh** είναι μια υπερβολική εφαπτομενική συνάρτηση. Είναι παρόμοια με την σιγμοειδή συνάρτηση αλλά είναι συμμετρική στο κέντρο των αξόνων. Αυτό έχει ως αποτέλεσμα να έχουμε διαφορετικά πρόσημα στην έξοδο των προηγούμενων στρωμάτων και που θα δοθούν σαν είσοδοι στα επόμενα στρώματα. Μπορεί να οριστεί όπως παρακάτω:

f(x) = 2sigmoid(2x)-1

Η **Tanh** συνάρτηση είναι συνεχής και διαφοροποιήσιμη, οι τιμές τις είναι μεταξύ του -1 και 1. Σε σύγκριση με την σιγμοειδή η κλίση της είναι πιο απότομη. Η **Tanh** προτιμάται από την σιγμοειδή λόγω των κλίσεων της που επιτρέπουν την διαφοροποίηση προς κάποια κατεύθυνση και επίσης λόγω του ότι έχει σαν κέντρο το 0.



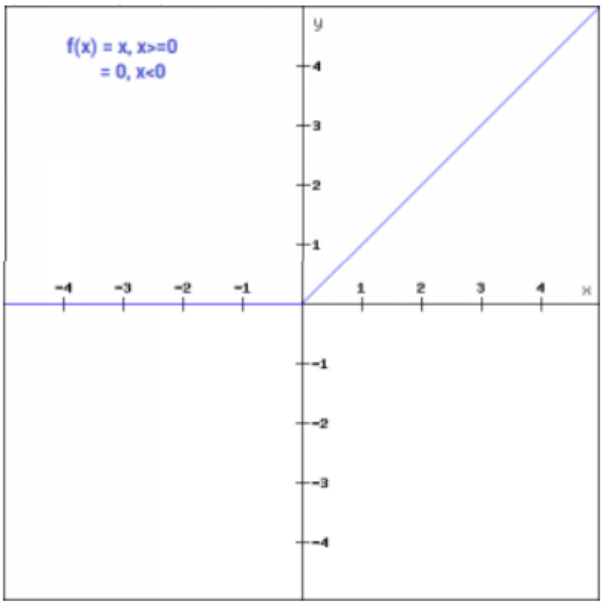
Εικόνα 39 Διάγραμμα συνάρτησης tanh

3.1.6.5 ReLU

Η συνάρτηση ανορθωτή **ReLU** (rectified liner unit) είναι μια μη γραμμική συνάρτηση ενεργοποίησης, ευρέως χρησιμοποιούμενη στα νευρωνικά δίκτυα. Το πλεονέκτημα της χρήσης της έχει να κάνει με το ότι όλοι οι νευρώνες δεν είναι ενεργοποιημένοι την ίδια χρονική στιγμή. Αυτό προϋποθέτει ότι οι νευρώνες θα είναι απενεργοποιημένοι μόνο όταν η έξοδος του γραμμικού μετασχηματισμού είναι 0. Μπορεί να οριστεί μαθηματικά όπως παρακάτω :

f(x) = max(0,x)

Η **ReLU** είναι πιο αποδοτική από τις άλλες συναρτήσεις ακριβώς λόγω του ότι δεν είναι όλοι οι νευρώνες ενεργοποιημένοι ταυτόχρονα αλλά μόνο ένας συγκεκριμένος αριθμός κάθε φορά. Σε κάποιες περιπτώσεις η τιμή της κλίσης είναι 0 λόγω του ότι τα βάρη και οι κλίσεις δεν ενημερώνονται κατά την οπισθοδιάδοση κατά την εκπαίδευση του νευρωνικού.



Εικόνα 40 Διάγραμμα συνάρτησης RELU

3.1.7 Dropout

Τα νευρωνικά δίκτυα βαθιάς μάθησης είναι πολύ πιθανό να παρουσιάσουν **overfitting** όταν τα εκπαιδεύσουμε με λίγα δείγματα στο **training dataset**. Ένα και μόνο μοντέλο, μπορεί να προσομοιώσει την ύπαρξη πολλών διαφορετικών αρχιτεκτονικών δικτύου απενεργοποιώντας τυχαία κόμβους κατά την εκπαίδευση του. Αυτή η τεχνική καλείται **dropout** και είναι μια υπολογιστικά φτηνή και αποτελεσματική μέθοδος κανονικοποίησης που αποσκοπεί στην μείωση του **overfitting** και στην βελτίωση του σφάλματος γενίκευσης.

3.1.7.1 Τυχαία απενεργοποίηση κόμβων

Το **dropout** είναι μια μέθοδος κανονικοποίησης που προσεγγίζει την εκπαίδευση πολλών αρχιτεκτονικών νευρωνικών δικτύων παράλληλα. Κατά την διάρκεια της εκπαίδευσης, κάποιος συγκεκριμένος αριθμός από τις εξόδους που παίρνουμε από ένα στρώμα του νευρωνικού, αγνοείται (**dropped out**). Αυτή η τεχνική έχει σαν αποτέλεσμα ένα εικονικό καινούργιο **layer** με διαφορετικό αριθμό νευρώνων και διαφορετική συνδεσμολογία με το **layer** που είχαμε αρχικά. Αυτό έχει σαν αποτέλεσμα, κάθε ενημέρωση σε κάποιο **layer** κατά την εκπαίδευση να πραγματοποιείται με διαφορετική ‘’όψη’’ του ίδιου **layer**.

Το **dropout** κάνει την διαδικασία εκπαίδευσης ‘’θορυβώδη’’, εξαναγκάζοντας τους κόμβους ενός **layer** να πάρουν πιθανολογικά, πολύ ή λιγότερη ευθύνη για τις εισόδους. Θεωρητικά το **dropout** διαχωρίζει τις καταστάσεις όπου τα στρώματα του δικτύου προσαρμόζονται σε λάθοι από προηγούμενα στρώματα και κάνει το μοντέλο πιο ισχυρό.

Η τυχαία δειγματοληψία και χρήση των νευρώνων των **layers** που χρησιμοποιούν **dropout** έχει ως αποτέλεσμα την μείωση της χωρητικότητας του δικτύου κατά την εκπαίδευση. Κατά συνέπεια ένα μεγαλύτερο δίκτυο με περισσότερους κόμβους μπορεί να είναι αποδοτικότερο όταν χρησιμοποιούμε **dropout**.

3.1.7.2 Τρόπος χρήσης dropout

Το **dropout** υλοποιείται ανά στρώμα στο νευρωνικό δίκτυο. Μπορεί να χρησιμοποιηθεί με τους περισσότερους τύπους στρωμάτων όπως πλήρως συνδεδεμένα στρώματα (**dense fully connected**), συνελικτικά τύπου στρώματα (**convolutional layers**) και αναδρομικά δίκτυα όπως αυτά του **LSTM**.

Το **dropout** πρέπει να υλοποιείται σε κανένα ή σε όλα τα ενδιάμεσα στρώματα ενός δικτύου, όπως και στο στρώμα εισόδου. Τέλος, δεν χρησιμοποιείται στο στρώμα εξόδου.

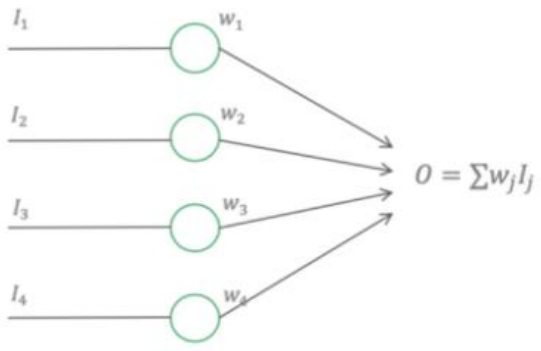
Μια νέα υπερπαράμετρος χρησιμοποιείται, η οποία ορίζει την πιθανότητα για το ποιοι έξοδοι-κόμβοι του κάθε στρώματος θα αγνοηθούν και ποιοι κόμβοι του κάθε στρώματος θα ληφθούν υπόψιν. Μια συνηθισμένη τιμή για την παραπάνω υπερπαράμετρο είναι το 0.5 για την διατήρηση της εξόδου κάθε κόμβου και η τιμές που πλησιάζουν το 1.0, όπως για παράδειγμα το 0.8, για την διατήρηση εισόδων εμφανές στρώμα (**input layer**).

Το **dropout** δεν χρησιμοποιείται μετά την εκπαίδευση κατά την διάρκεια της εξαγωγής πρόβλεψης από το δίκτυο. Τα βάρη του δικτύου θα είναι μεγαλύτερα ακριβώς λόγω του **dropout**. Συμπερασματικά, πριν την καθιέρωση της τελικής μορφής του δικτύου, τα βάρη κλιμακώνονται σύμφωνα με την τιμή του **dropout**.

Η κλιμάκωση των βαρών από την τιμή του **dropout**, μπορεί να συμβεί κατά την διάρκεια της εκπαίδευσης του δικτύου, μετά την φάση ενημέρωσης των βαρών στο τέλος κάθε τμήματος δειγμάτων εκπαίδευσης (**mini-batch**). Η παραπάνω μέθοδος καλείται ‘’**inverse dropout**’’ και δεν απαιτεί τροποποίηση των βαρών κατά την εκπαίδευση. Ο συγκεκριμένος τρόπος υλοποίησης του **dropout** είναι αυτός που χρησιμοποιούν και οι βιβλιοθήκες **Keras** και **PyTorch**.

3.1.7.3 Μαθηματικά πίσω από το Dropout

Ας θεωρήσουμε έναν γραμμικό κόμβο μονού στρώματος σε ένα δίκτυο όπως παρακάτω:



Εικόνα 41 Γραμμικός κόμβος layer

Καλείται γραμμικός λόγω της συνάρτησης ενεργοποίησης, f(x) = x. Όπως μπορούμε να παρατηρήσουμε στην παραπάνω εικόνα, η έξοδος του **layer** είναι ένα γραμμικά βεβαρημένο άθροισμα εισόδων. Για την εκτίμηση του μοντέλου θα

χρησιμοποιήσουμε συναρτήσεις απώλειας (**loss functions**). Για το γραμμικό μας **layer** θα χρησιμοποιήσουμε την μέθοδο του **least square loss**.

$$E_N = \frac{1}{2}\left(t - \sum_{i=1}^{n} w_i' I_i\right)^2 \quad (1)$$

$$E_D = \frac{1}{2}\left(t - \sum_{i=1}^{n} \delta_i w_i I_i\right)^2 \quad (2)$$

Εικόνα 42 Απώλεια δικτύου χωρίς και με dropout

(1): Δείχνει την απώλεια για ένα δίκτυο χωρίς **dropout**,

(2): Δείχνει την απώλεια με **dropout**. Η τιμή του **dropout** είναι **δ**, όπου **δ ~Bernoulli(p)**. Αυτό σημαίνει πως το **δ** ισούται με 1 με πιθανότητα **p** και 0 σε οποιαδήποτε άλλη περίπτωση.

Η οπισθοδιάδοση του δικτύου κατά την εκπαίδευση χρησιμοποιεί την προσέγγιση της μεθόδου επαναληπτικής βελτιστοποίησης **Gradient descent**. Κατά συνέπεια, πρώτα θα κοιτάξουμε την κλίση του **dropout** του δικτύου στην (2) και στην συνέχεια θα επανέλθουμε στο αρχικό δίκτυο της (1).

$$\frac{\partial E_D}{\partial w_i} = -t\delta_i I_i + w_i \delta_i^2 I_i^2 + \sum_{j=1, j\neq i}^{n} w_j \delta_i \delta_j I_i I_j \quad (3)$$

Εικόνα 43 Κλίση dropout δικτύου από (2)

Στην συνέχεια θα δείξουμε την σχέση μεταξύ κλίσης με **dropout** και κλίσης στο συνηθισμένο δίκτυο. Ας υποθέσουμε πως στην (1) έχουμε *w'* = *p*w*.

$$E_N = \frac{1}{2}\left(t - \sum_{i=1}^{n} p_i w_i I_i\right)^2 \quad (4)$$

Εικόνα 44 Απώλεια χωρίς dropout

Παίρνοντας την παράγωγο της (4) βρίσκουμε:

$$\frac{\partial E_N}{\partial w_i} = -tp_i I_i + w_i p_i^2 I_i^2 + \sum_{j=1, j\neq i}^{n} w_j p_i p_j I_i I_j \quad (5)$$

Εικόνα 45 Παράγωγος της απώλειας χωρίς dropout

Άμα πάρουμε την αναμενόμενη τιμή του παραπάνω έχουμε:

$$\begin{aligned} E\left[\frac{\partial E_D}{\partial w_i}\right] &= -tp_i I_i + w_i p_i^2 I_i^2 + w_i Var(\delta_i) I_i^2 + \sum_{j=1, j\neq i}^{n} w_j p_i p_j I_i I_j \\ &= \frac{\partial E_N}{\partial w_i} + w_i Var(\delta_i) I_i^2 \\ &= \frac{\partial E_N}{\partial w_i} + w_i p_i (1 - p_i) I_i^2 \end{aligned} \quad (6)$$

Εικόνα 46 Αναμενόμενη τιμή κλίσης με dropout

Άμα παρατηρήσουμε την (6), η αναμενόμενη τιμή της κλίσης με **dropout**, είναι ίση με την κλίση του κανονικοποιημένου δικτύου Εκ αν *w'* = *p*w.*

3.1.7.4 Dropout ανάλογο κανονικοποιημένου δικτύου

Αυτό σημαίνει πως η μείωση στο σφάλμα του **dropout** (2) είναι αντίστοιχη με το να ελαχιστοποιήσουμε το κανονικοποιημένο δίκτυο μας. Αυτό περιγράφεται από την παρακάτω εξίσωση:

$$E_R = \frac{1}{2}\left(t - \sum_{i=1}^{n} p_i w_i I_i\right)^2 + \sum_{i=1}^{n} p_i(1-p_i)w_i^2 I_i^2$$

Εικόνα 47 Η μείωση στο σφάλμα του dropout είναι αντίστοιχη με το να ελαχιστοποιήσουμε το κανονικοποιημένο δίκτυο μας

Δηλαδή άμα διαφοροποιήσουμε το κανονικοποιημένο δίκτυο της (7), θα πάρουμε την κλίση ενός δικτύου με **dropout** όπως στην εξίσωση (6).

3.1.7.5 Γιατί το dropout rate, p = 0.5 οδηγεί στην μέγιστη κανονικοποίηση ;

Το παραπάνω συμβαίνει διότι η παράμετρος κανονικοποίησης, **p(1 – p)** στην (7), είναι μέγιστη για **p = 0.5**.

3.1.7.6 Τι τιμές πρέπει να διαλέγουμε για το p για διαφορετικά layers ;

Στην βιβλιοθήκη **Keras**, το **dropout rate** είναι **(1 – p)**. Για τα ενδιάμεσα στρώματα, είναι ιδανικό, για ένα μεγάλο δίκτυο, να διαλέξουμε **(1 – p) = 0.5**. Για το στρώμα εισόδου, το **(1 – p)** πρέπει να παραμένει ίσο με **0.2** ή και χαμηλότερο. Αυτό διότι άμα ''πετάμε'' τα δεδομένα εισόδου, μπορεί να επηρεάσουμε την εκπαίδευση. Ένα **(1 – p) > 0.5** δεν συστήνεται διότι οδηγεί στον διαχωρισμό συνδέσεων χωρίς να αυξάνεται η κανονικοποίηση.

3.1.7.7 Γιατί κλιμακώνουμε τα βάρη ως προς p κατά την διάρκεια του testing ;

Διότι η αναμενόμενη τιμή για ένα δίκτυο με **dropout** είναι ίση με ένα κανονικό δίκτυο με τα βάρη του κλιμακωμένα με το **dropout rate p**. Αυτή η κλιμάκωση κάνει τα αποτελέσματα ενός δικτύου με **dropout** συγκρίσιμα με ένα δίκτυο με όλες του τις διασυνδέσεις και τους νευρώνες.

3.1.8 Optimizers

3.1.8.1 Τι είναι ο optimizer;

Κατά την διαδικασία της εκπαίδευσης, ενός νευρωνικού δικτύου, προσαρμόζουμε και αλλάζουμε τις παραμέτρους (**τα βάρη**) του μοντέλου μας και προσπαθούμε να ελαχιστοποιήσουμε την συνάρτηση σφάλματος (**loss function**, μέτρηση της ορθότητας των προβλέψεων μας) και να κάνουμε προβλέψεις οι οποίες είναι όσο το δυνατόν ακριβέστερες και βέλτιστες. Πως όμως εκτελείται η παραπάνω ενέργεια και πως, πόσο και κάθε πότε αλλάζουν οι παράμετροι του μοντέλου μας ;

Οι **optimizers** είναι το εργαλείο το οποίο χρησιμοποιούμε για να δώσουμε λύση στα παραπάνω ερωτήματα. Οι **optimizers** συνδυάζουν την συνάρτηση σφάλματος και τις παραμέτρους του μοντέλου μας, ενημερώνοντας το μοντέλο κατ αντιστοιχία κάθε φορά με την έξοδο που παίρνουμε από την συνάρτηση σφάλματος. Δηλαδή αυτό που κάνουν οι **optimizers** είναι να διαμορφώνουν το μοντέλο μας, ενημερώνοντας κάθε φορά τα βάρη, ώστε να πετυχαίνει τον μεγαλύτερο βαθμό ακρίβειας. Η

συνάρτηση σφάλματος ενημερώνει τον **optimizer** σχετικά με το αν λαμβάνει τις βέλτιστες αποφάσεις για την βελτίωση του μοντέλου.

3.1.8.2 Gradient descent

Ο πιο διάσημος **optimizer** είναι ο **Gradient descent**. Ο συγκεκριμένος αλγόριθμος χρησιμοποιείται σε πολλούς τύπου μηχανικής μάθησης (και άλλων μαθηματικών προβλημάτων) για βελτιστοποίηση. Είναι γρήγορος και ευέλικτος. Παρακάτω περιγράφουμε τον τρόπο λειτουργίας του:

1. Υπολογίζει την συνάρτηση σφάλματος σύμφωνα με μικρές αλλαγές του βάρους ξεχωριστά

2.Προσαρμόζει κάθε βάρος ξεχωριστά σύμφωνα με την κλίση (**gradient**)

3. Επαναλαμβάνει τα βήματα 1 και 2 μέχρι η συνάρτηση σφάλματος να φτάσει την ελάχιστη δυνατή τιμή της

Το δύσκολο μέρος του συγκεκριμένου αλγορίθμου (και τον **optimizers** γενικότερα) είναι η κατανόηση των **gradients**, τα οποία αναπαριστούν τι μπορεί να κάνει μια μικρή αλλαγή σε κάποιο βάρος ή σε κάποια παράμετρο στην συνάρτηση σφάλματος. Συνδέουν την συνάρτηση σφάλματος με τα βάρη και μας λένε τι ενέργειες πρέπει να κάνουμε με τα βάρη μας (πρόσθεση, αφαίρεση κ.τ.λ.) για να κάνουμε το μοντέλο μας ακριβέστερο.

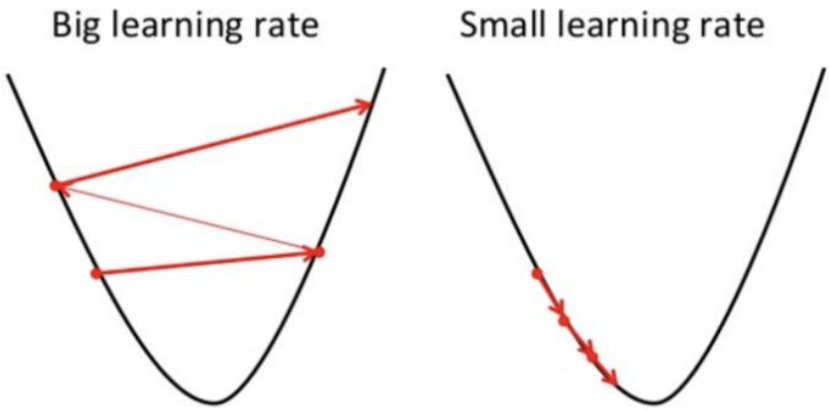
Ένα πρόβλημα που συναντάται κατά την βελτιστοποίηση είναι αυτό της προσκόλλησης σε κάποιο **τοπικό ελάχιστο**. Όταν έχουμε να κάνουμε με πολυδιάστατα **datasets** (με πολλές μεταβλητές), είναι πιθανό να βρούμε μια περιοχή όπου φαίνεται πως έχουμε φτάσει στην ελάχιστη δυνατή τιμή της συνάρτησης σφάλματος μας, αλλά στην πραγματικότητα βρισκόμαστε σε κάποιο τοπικό ελάχιστο. Για την αποφυγή του συγκεκριμένου προβλήματος πρέπει να σιγουρευτούμε πως χρησιμοποιούμε κατάλληλο ρυθμό εκμάθησης (**learning rate**).

3.1.8.3 Learning rate

Αλλάζοντας τα βάρη μας πολύ γρήγορα, προσθέτοντας και αφαιρώντας με μεγάλο ρυθμό, μπορεί να εμποδίσουμε την μείωση της συνάρτησης σφάλματος. Δεν θέλουμε να κάνουμε μεγάλες αλλαγές και να παραλείψουμε την βέλτιστη τιμή για κάποιο βάρος.

Για να διασφαλίσουμε πως δεν θα συμβεί κάτι τέτοιο, χρησιμοποιούμε μια μεταβλητή που καλείται **ρυθμός μάθησης** (**learning rate**). Αυτή η μεταβλητή είναι ένας πολύ μικρός αριθμός, συνήθως κάτι όπως **0.001**, με τον οποίο πολλαπλασιάζουμε τα **gradients** για να τα κλιμακώσουμε ανάλογα. Αυτό διασφαλίζει πως όποιες αλλαγές κάνουμε στα βάρη είναι μικρές και έτσι αποφεύγουμε να κάνουμε μεγάλες αλλαγές που δεν θα οδηγήσουν σε κάποια βέλτιστη τιμή.

Ταυτόχρονα, δεν θέλουμε να κάνουμε υπερβολικά μικρά βήματα, διότι και σε αυτή την περίπτωση μπορεί να μην καταλήξουμε με τις σωστές τιμές για τα βάρη μας. Τα πολύ μικρά βήματα μπορεί να οδηγήσουν στην σύγκλιση σε κάποιο τοπικό ελάχιστο.



Εικόνα 48 Διαγράμματα μεγάλου και μικρού learning rate αντίστοιχα

3.1.8.4 Κανονικοποίηση-Regularization

Στην μηχανική μάθηση, ένα κοινό πρόβλημα είναι το **overfitting**. Το **overfitting** σημαίνει πως το μοντέλο μας κάνει σωστές προβλέψεις σε δεδομένα με τα οποία το έχουμε εκπαιδεύσει αλλά λανθασμένες σε καινούργια δεδομένα τα οποία δεν έχει ξαναδεί. Αυτό μπορεί να συμβεί αν σε κάποια παράμετρο έχει δοθεί μεγάλο βάρος και καταλήγει να ελέγχει τις προβλέψεις. Η κανονικοποίηση βοηθάει την διαδικασία της βελτιστοποίησης στην αποφυγή του παραπάνω προβλήματος.

Κατά την κανονικοποίηση, προσθέτεται στην συνάρτηση σφάλματος μια μεταβλητή που αποκόπτει τις μεγάλες τιμές στα βάρη. Αυτό σημαίνει ότι εκτός από τις λάθος προβλέψεις και οι σωστές προβλέψεις με υπερβολικά μεγάλα βάρη αναγνωρίζονται σαν λάθος. Αυτή η διαδικασία διασφαλίζει πως τα βάρη έχουν μικρότερες τιμές και ως εκ τούτου κάνουν καλύτερη γενίκευση σε νέα δεδομένα.

3.1.8.5 Adam

Ο **Adam** (**Adaptive Moment estimation**) είναι ένας τρόπος χρήσης περασμένων κλίσεων (**gradients**) για τον υπολογισμό των τωρινών κλίσεων. Ο **Adam** επίσης χρησιμοποιεί την τεχνική του **momentum** προσθέτοντας κλάσματα προηγούμενων **gradients** στο τωρινό. Για παράδειγμα άμα το **momentum** είναι ίσο με 0.90, θα ακολουθήσουμε το 90% της έως τώρα κατεύθυνσης και 10% επιπρόσθετα της νέας κατεύθυνσης και θα ρυθμίσουμε τα βάρη αναλόγως πολλαπλασιάζοντας τον πίνακα κατεύθυνσης με το **learning rate**.

3.1.8.6 Adagrad

Ο **Adagrad** προσαρμόζει το **learning rate** σύμφωνα με ορισμένα χαρακτηριστικά. Αυτό σημαίνει πως κάποια βάρη στο **dataset** θα έχουν διαφορετικό **learning rate** από άλλα. Αυτή η τεχνική δουλεύει αρκετά καλά για αραιά **datasets** όπου λείπουν πολλά δεδομένα εισόδου. Ένα μεγάλο μειονέκτημα του **Adagrad** είναι πως το **learning rate** γίνεται υπερβολικά μικρό με το πέρασμα του χρόνου.

3.1.8.7 AdaMax

Ο **AdaMax optimizer** είναι μια παραλλαγή του **Adam** και χρησιμοποιεί το **infinity norm**. Ένα **norm** επιγραμματικά είναι μια συνάρτηση που αντιστοιχίζει μια είσοδο σε

κάποια έξοδο. Τα **norms** μας δίνουν πληροφορίες για το μήκος ενός διανύσματος σε κάποια διάσταση.

Υπάρχουν πολλά **norms** όπως το **L0 norm** (μας δίνει πληροφορίες για τον αριθμό των μη μηδενικών στοιχείων ενός διανύσματος), το **L1 norm** και το **L2 norm**.

Τα παραπάνω συνοψίζονται με το **p-norm** που υπολογίζει το **L-norm** για κάποιο **p** (π.χ. για p = 2 έχουμε το L2 norm). Όταν το **p** να πλησιάζει το άπειρο, έχουμε το **max norm** ή το **infinity norm**. Δεδομένου κάποιου διανύσματος **x = {x_1, x_2, …,x_n }**, το **infinity norm** μας δίνει την μέγιστη τιμή του διανύσματος.

Αυτή είναι η βασική διαφορά μεταξύ του **Adam** και του **AdaMax**, ο οποίος αντί για το **L2 norm** χρησιμοποιεί το **L-infinity norm**. Όταν ενημερώνουμε το **gradient** στον **AdaMax**, είναι το μέγιστο μεταξύ των προηγούμενων κλίσεων και των τωρινών κλίσεων.

Ο **AdaMax** είναι πιο χρήσιμος για **dataset** που περιέχουν περισσότερο θόρυβο (**outliers**) διότι λόγω των περασμένων ενημερώσεων του **gradient**, εμποδίζετε κάποια απότομη αλλαγή σε αυτό.

3.1.9 Συμπέρασμα

Από τα πειράματα που πραγματοποιήθηκαν στο **Κεφάλαιο 4**, προτείνουμε για κάποιο ανάλογο πρόβλημα πρόβλεψης εργασιών, την χρήση **10 epochs** (10 έως το πολύ 20) ώστε να επιτευχθεί η υψηλότερη τιμή ακρίβειας χωρίς να ξεπεραστεί το σημείο όπου το **accuracy** σταματάει να αυξάνεται και το μοντέλο μας αρχίζει απλώς να αποστηθίζει τα δεδομένα, χωρίς να βελτιώνεται η προβλεπτική του ικανότητα με νέα δεδομένα.

Όσον αφορά την περίπτωση των κλασσικών αλγορίθμων εξαγωγής διαδικασιών (Alpha, Heuristic, Inductive), αποδείχτηκε πως η πιο αξιόλογη και η πιο αξιόπιστη μέθοδος, είναι αυτή του **Heuristic miner**.

Η προτεινόμενη τιμή για το **batch-size** για datasets χιλιάδων δειγμάτων, αποφανθήκαμε πως είναι αυτή του **1**. Για datasets μερικών εκατοντάδων δειγμάτων είναι καλύτερα να χρησιμοποιούμε **batch-size** ίσο με **64**.

Ο αριθμός των **layers**, όταν αυξάνεται, φάνηκε πως οδηγεί σε ασταθή συμπεριφορά του μοντέλου μας και προκαλεί **overfitting**. Προτείνουμε την χρήση **1 layer** ώστε να κρατήσουμε απλούστερο το μοντέλο μας χωρίς να μειώσουμε την προβλεπτική του ικανότητα.

Για τον αριθμό των νευρώνων προτείνουμε για ανάλογα datasets την χρήση **30 νευρώνων**, ώστε να πετυχαίνουμε υψηλότερο **accuracy**.

Επιπλέον για την περίπτωση των **Activation functions** προτείνουμε και για datasets μεγαλύτερα σε μέγεθος αλλά και για μικρότερα την χρήση της **Softmax Activation function**, αφού με την συγκεκριμένη παίρνουμε τα λιγότερα λανθασμένα αποτελέσματα χωρίς να αυξάνουμε το **overfitting**.

Το **dropout** που θα πρέπει να επιλέγεται είναι μεταξύ του **0.2**.

Τέλος ο **Optimizer** με τον οποίο εξάγουμε τα βέλτιστα αποτελέσματα είναι ο **Adam**.

Κεφάλαιο 4

4.1 Πειράματα με process mining αλγορίθμους

4.1.1Πειράματα για το dataset της τράπεζας

Παρακάτω βλέπουμε τα **process models** που εξάχθηκαν για το **dataset** της τράπεζας από τους **process mining algorithms**. Όπως παρατηρούμε τα **Petri nets** που εξάχθηκαν δεν είναι ιδιαίτερα πολύπλοκα, κάτι που μας οδηγεί στο συμπέρασμα πως υπάρχουν καλά ορισμένες διεργασίες στο **dataset** μας και πως δεν θα χρειαστεί η κατασκευή κάποιου πολύπλοκου **LSTM** μοντέλου.

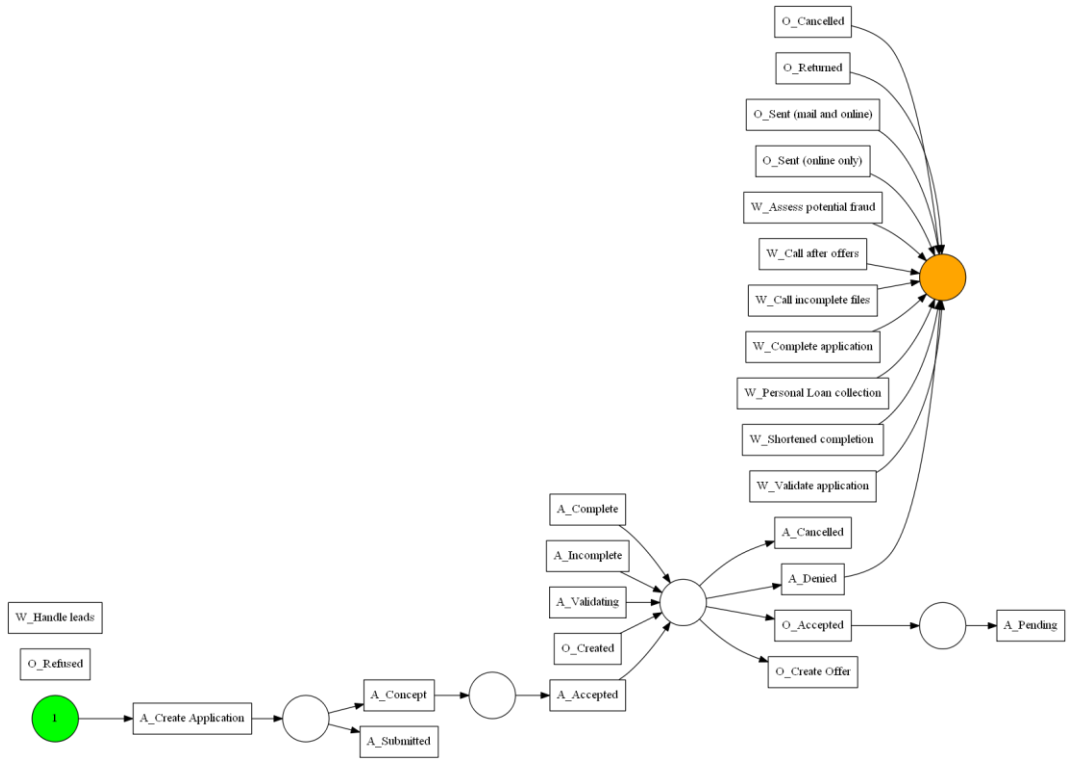
Αρχικά εξάγουμε το μοντέλο διεργασιών με την χρήση του **Alpha miner**. Ο **Alpha miner** λόγω του ότι είναι ο πρώτος αλγόριθμος που δημιουργήθηκε για την ανακάλυψη μοντέλων διεργασιών, είναι και ο λιγότερο πετυχημένος ως προς την περιγραφή του **event log** όπως θα δούμε και από τις μετρήσεις που παίρνουμε στην συνέχεια.. Άμα παρατηρήσουμε το παρακάτω **Petri net** θα δούμε πως ο αλγόριθμος δεν έχει αποτυπώσει και με ιδιαίτερη ακρίβεια τις ακολουθίες των γεγονότων που υπάρχουν στο **event log** μας. Για παράδειγμα μια από τις δύο αρχικές ενέργειες που βλέπουμε παρακάτω είναι το **O_Refused** δηλαδή η άρνηση της προσφοράς που έκανε η τράπεζα από τον πελάτη, κάτι που δεν βγάζει νόημα. Επιπλέον παρατηρούμε πως γενικότερα δεν έχουν αποτυπωθεί πολλές ακολουθίες γεγονότων και πολλές ενέργειες εκτελούνται μόνες τους, για παράδειγμα το **O_Cancelled** δηλαδή η ακύρωση της προσφοράς, η οποία λαμβάνει μέρος και ύστερα βλέπουμε πως φτάνουμε σε κατάσταση τερματισμού, κάτι που είναι λανθασμένο.

Στην συνέχεια βλέπουμε το **Petri net** που εξάγαμε με την χρήση του **Heuristic miner** αλγορίθμου και μπορούμε εύκολα να παρατηρήσουμε την μεγάλη διαφορά που υπάρχει με τον **Alpha miner** αλλά και την πληρέστερη και ακριβέστερη, όπως θα δούμε από τις μετρήσεις που κάνουμε παρακάτω, αποτύπωση των ακολουθιών των διεργασιών. Ο **Heuristic miner** είναι καλύτερος λόγω του ότι λαμβάνει υπόψιν του τις συχνότητες εμφάνισης των γεγονότων, εντοπίζει βρόγχους. Άμα αναζητήσουμε τις ενέργειες των ακολουθιών που βλέπουμε στην εικόνα, για παράδειγμα στο **dataset** της τράπεζας, θα δούμε πως όντως έχει αναπαραστήσει με ακρίβεια πολλές ακολουθίες γεγονότων. Για παράδειγμα αρχίζουμε από την δημιουργία της αίτησης **A_Create Application**, συνεχίζουμε με την υποβολή της **A_Submitted** και στην συνέχεια ακολουθούν οι ενέργειες **W_Handle leads**, **W_Complete application** και **A Concept**. Η συγκριμένη ακολουθία ενεργειών είναι επιτυχώς αποτυπωμένη στο **Petri net**, άμα πάρουμε όμως **A_Concept A_Accepted** θα δούμε πως δεν αποτυπώνεται πουθενά στο **Petri net** κάτι που είναι λανθασμένο γιατί μέσα στο **event log** υπάρχει η συγκεκριμένη αλληλουχία ενεργειών. Παρόλαυτα ο **Heuristic miner** είναι μια βελτίωση του **Alpha miner** αλγορίθμου και σίγουρα είναι πολύ πιο ακριβής στην εξόρυξη διεργασιών.

Έπειτα εξάγουμε μοντέλο διεργασιών με την χρήση του **Inductive miner** αλγορίθμου. Παρόλο που ο **Inductive miner** είναι βελτίωση του **Alpha miner** και του **Heuristic miner** μπορούμε εύκολα να συγκρίνουμε το **Petri net** με τα γεγονότα στο **event log** και να δούμε πως υπάρχουν αρκετές αστοχίες στην αποτύπωση των ακολουθιών για παράδειγμα η υποβολή της αίτησης δανείου στην τράπεζα **A_Submitted** δεν ακολουθείται ποτέ από την αποδοχή της **A_Accepted**. Επίσης ο

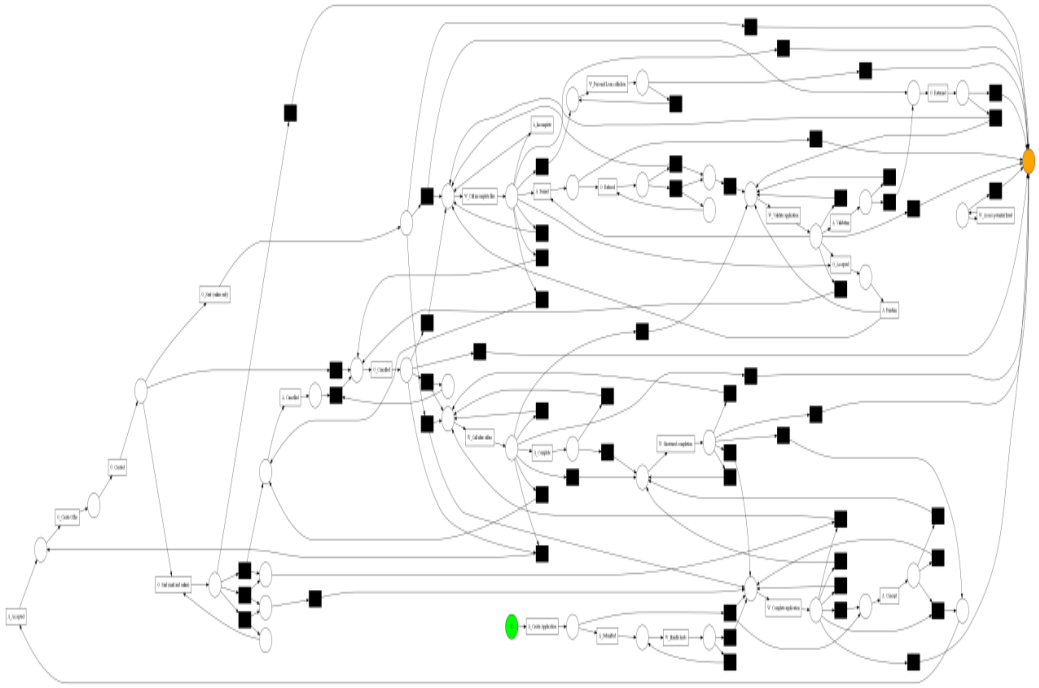
υπολογισμός του **Precision** για τον **Inductive miner** δεν τερμάτιζε, κάτι που εμποδίζει να κρίνουμε λεπτομερέστερα την επιτυχή λειτουργία του συγκεκριμένου αλγορίθμου.

4.1.1.1 Alpha miner



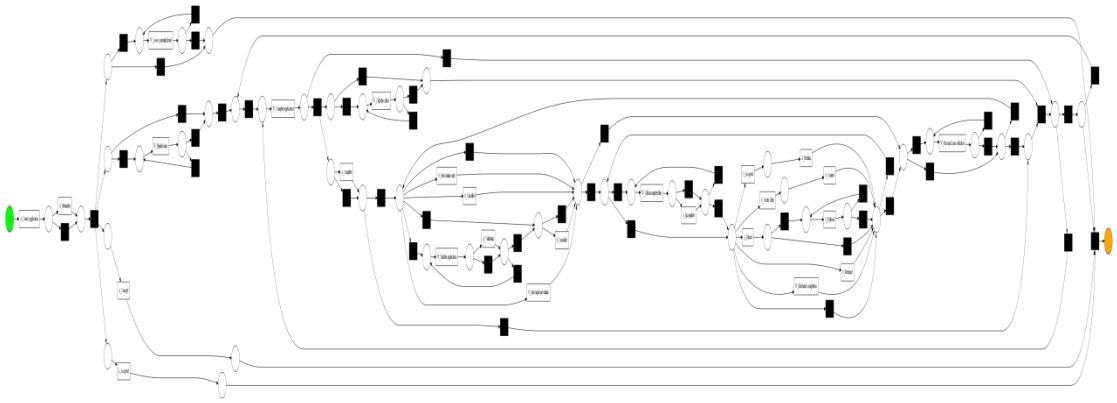
Εικόνα 49 Process model με Alpha miner

4.1.1.2 Heuristics miner



Εικόνα 50 Process model με Heuristic miner

4.1.1.3 Inductive miner



Εικόνα 51 Process model με Inductive miner

4.1.1.4 Fitness

Το **fitness** είναι ένα **metric** για την επαναληψιμότητα των **traces** που χρησιμοποιήθηκαν για να γίνει εξαγωγή του **process model**. Μια αξιολόγηση με βάση το **fitness** μας παρέχει:

-Τον μέσο όρο του **fitness** για το **log** με βάση το **model**, που είναι μια τιμή μεταξύ 0 και 1 και υποδεικνύει πόσο καλά το μοντέλο αναπαριστά την συμπεριφορά των **traces** στο **log**.

-Το ποσοστό επί της εκατό των **traces** στο **log** που ταιριάζουν απόλυτα με το εξαχθέν μοντέλο.

Παρακάτω βλέπουμε τα αποτελέσματα για το **fitness** της τράπεζας. Χρησιμοποιήθηκε το **alpha_petri** γιατί με τα άλλα **petri nets** ο υπολογισμός έπαιρνε πάνω από 5 ώρες.

```
In [8]: print("fitness_alpha=",fitness_alpha)
fitness_alpha= {'perc_fit_traces': 0.0, 'average_trace_fitness': 0.40591920920940966, 'log_fitness':
0.3956666023296359}
```

Εικόνα 52 Fitness με την χρήση του petrinet από τον Alpha miner

Καταλήγουμε στο συμπέρασμα πως το μοντέλο που εξάχθηκε με τον **alpha miner** δεν αναπαριστά πολύ καλά την συμπεριφορά των **traces** στο **log**.

4.1.1.5 Precision

Τα διαφορετικά **prefixes** στο **log** αποτυπώνονται όσο είναι δυνατό στο μοντέλο που έχει παραχθεί. Το **set** των μεταβάσεων (**transitions**) που είναι ενεργοποιημένες στο **process model** συγκρίνετε με το **set** των **activities** που ακολουθούν μετά το **prefix** του **log**. Όσο περισσότερο διαφορετικά είναι τα **set** τόσο χαμηλότερη τιμή έχει το **precision**. Όσο πιο όμοια είναι τα **set** τόσο μεγαλύτερη θα είναι η τιμή του **precision**.

Το παραπάνω δουλεύει μόνο αν υπάρχει επαναληψιμότητα του **prefix** στο **process model**. Αν το **replay** δεν παράγει αποτέλεσμα, το **prefix** δεν

λαμβάνεται υπόψιν για τον υπολογισμό του **precision**. Δηλαδή το **precision** θα υπολογίζεται με **unfit processes** κάτι το οποίο δεν έχει νόημα [33].

4.1.1.5.1 Precision με alpha miner

Το **precision** χρησιμοποιώντας το **petri net** του **alpha miner** για **10.000 traces** παρατηρούμε ότι είναι πολύ χαμηλό.

```
In [33]: alpha_prec
Out[33]: 0.09550735712946357
```

Εικόνα 53 Precision με Alpha miner

Άμα χρησιμοποιήσουμε 20.000 δείγματα βλέπουμε ότι το **precision** πέφτει. Άρα καταλήγουμε στο συμπέρασμα πως το **process model** που έχει εξαχθεί από τον **alpha miner** δεν είναι αξιόπιστο.

4.1.1.5.2 Precision με heuristic miner

Το **precision** χρησιμοποιώντας το **petri net** του **heuristic miner** και **10.000 traces** είναι το παρακάτω.

```
In [26]: heuristic_prec
Out[26]: 0.763625534178767
```

Εικόνα 54 Precision με Heuristic miner

Όπως παρατηρούμε πετυχαίνουμε αρκετά υψηλή τιμή **precision** αυτό σημαίνει πως το **transitions set** στο **process model** έχει μεγάλη ομοιότητα με το **activities set**, δηλαδή των **activities** που ακολουθούν μετά από κάθε **prefix**.

4.1.2 Πειράματα με το dataset του εργοστασίου

Από τα παρακάτω **Petri nets** παρατηρούμε πως έχουμε πιο πολύπλοκες διεργασίες για το **dataset** του εργοστασίου. Επίσης οδηγούμαστε στο συμπέρασμα πως θα έχουμε πιθανόν μικρότερο **accuracy** με την χρήση του συγκεκριμένου **dataset** λόγω όχι μόνο του μικρού μεγέθους του αλλά και λόγω των πολύπλοκων ακολουθιών που το απαρτίζουν.

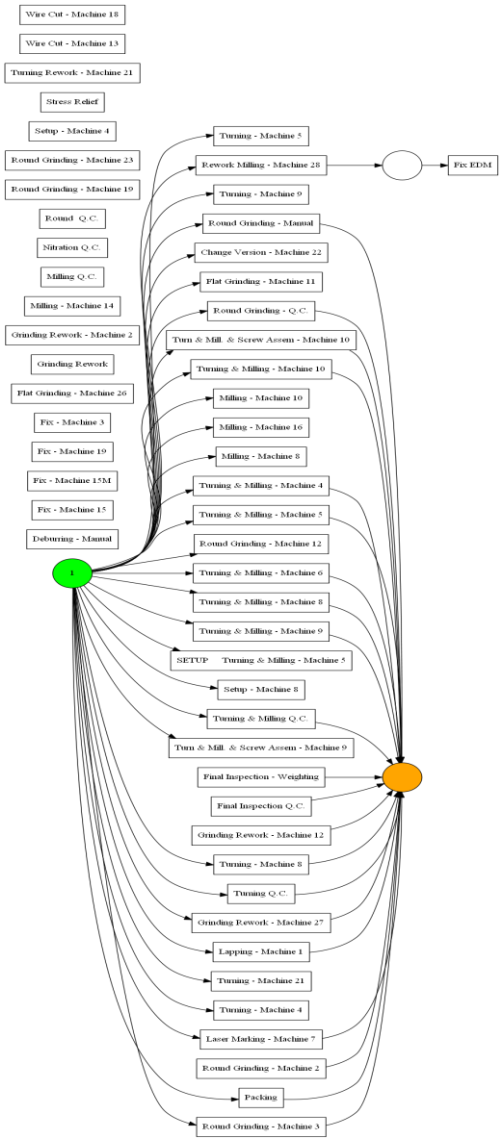
Αρχικά όπως και πριν κάνουμε χρήση του **Alpha miner** αλγορίθμου για την εξαγωγή μοντέλου διεργασιών. Παρατηρούμε πως πολλές ενέργειες μπορούν να θεωρηθούν αρχικές ενέργειες, όπως για παράδειγμα η χρήση του **Wire Cut – Machine 18**. Είναι εμφανές πως δεν μπορούμε εύκολα να ξεχωρίσουμε ποια ενέργεια έπεται της προηγούμενης. Όπως αναφέραμε και παραπάνω ο **Alpha miner** λόγω του ότι είναι ένας από τους πρώτους αλγορίθμους εξόρυξης διεργασιών που αναπτύχθηκαν δεν είναι και ο πλέον καταλληλότερος για την ανακάλυψη **process models**. Αυτό αποδεικνύεται και από την μέτρηση της ακρίβειας του εξαχθέντος μοντέλου, η οποία μας δίνει 0.

Στην συνέχεια ο **Heuristic miner** εξάγει ένα πολύ μεγαλύτερο **Petri net** από ότι ο **Alpha miner** και πολύ πιο δυσκολονόητο όπως παρατηρούμε από την εικόνα. Παρόλαυτα θα παρατηρήσουμε πως περιγράφει επιτυχώς τις συσχετίσεις των

ενεργειών. Για παράδειγμα η ενέργεια της μηχανής **FlatGrinding-Machine11** ακολουθείται όντως από την λειτουργία της μηχανής **Lapping-Machine1**. Η εξαγωγή επιτυχημένου μοντέλου διεργασιών αποδεικνύεται και από την μέτρηση του **precision** που βγαίνει ακριβώς 1.

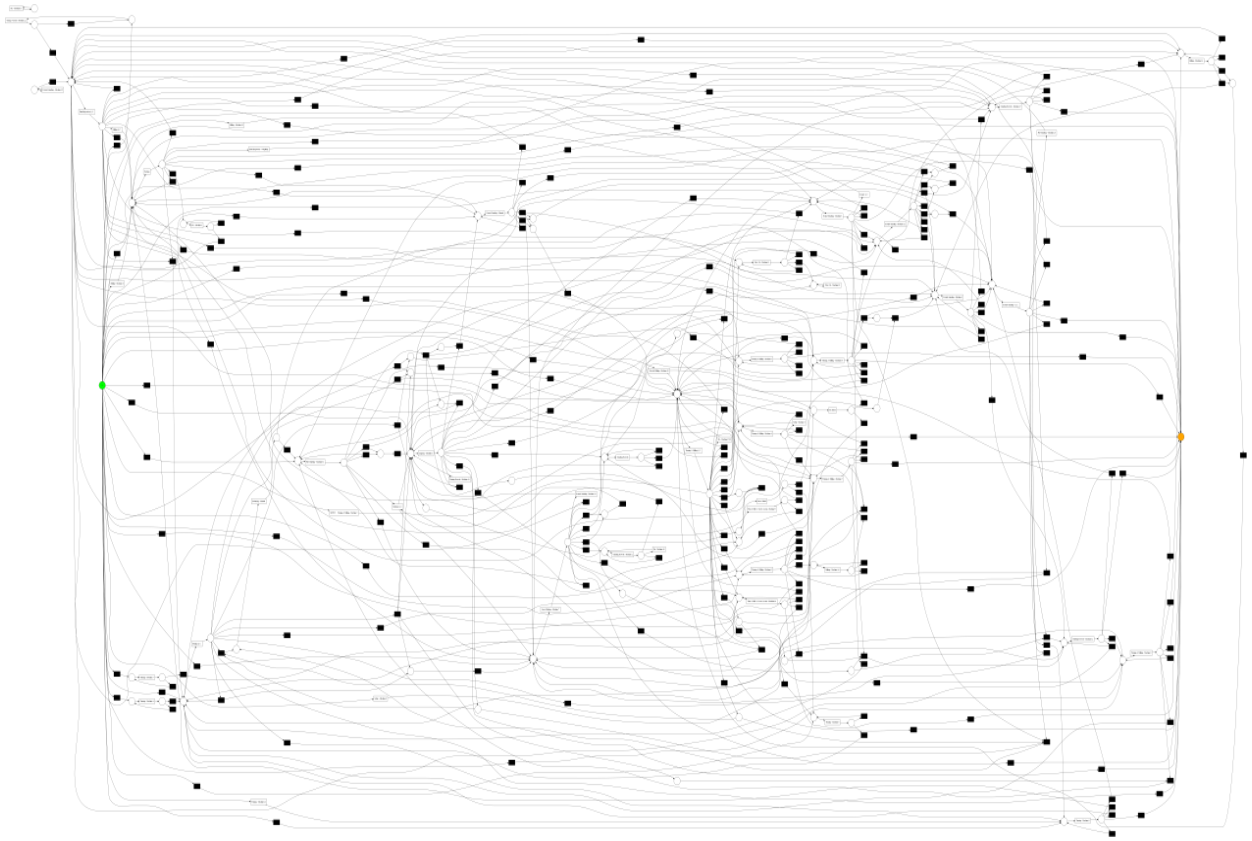
Το **Petri net** που παράγεται από τον **Inductive miner** παρατηρούμε πως είναι μικρότερο σε μέγεθος από αυτό του **Heuristic miner** και λιγότερο πολύπλοκο. Επίσης παρατηρούμε πως τα ίχνη του **event log** δεν αναπαριστούνται σωστά στο **Petri net**. Παρακάτω βλέπουμε πως ο συγκεκριμένος αλγόριθμος πετυχαίνει πολύ χαμηλό **precision**, κάτι που μας οδηγεί στο συμπέρασμα πως δεν είναι κατάλληλος για **process mining** στην περίπτωση που μελετάμε.

4.1.2.1 Alpha miner



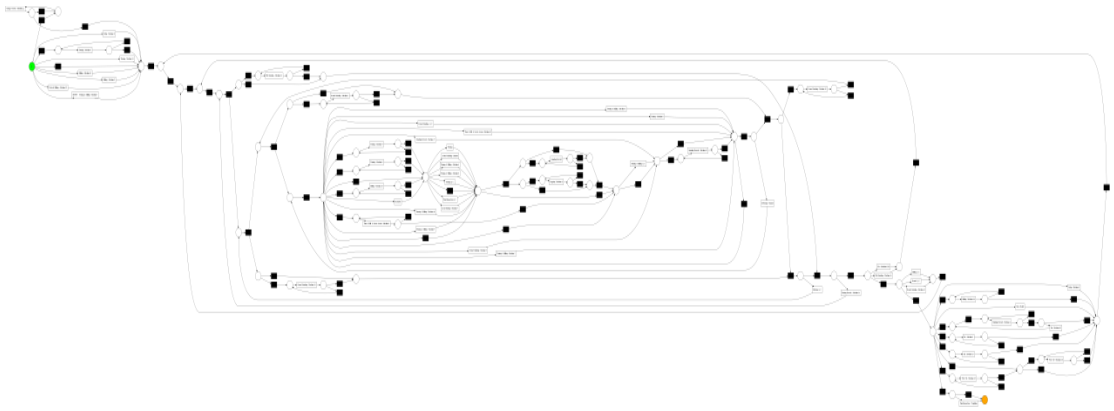
Εικόνα 55 Process model με Alpha miner

4.1.2.2 Heuristic miner



Εικόνα 56 Process model με Heuristic miner

4.1.2.3 Inductive miner



Εικόνα 57 Process model με Inductive miner

4.1.2.4 Fitness

4.1.2.4.1 Fitness με Alpha miner

```
In [16]: print("fitness_alpha=",fitness_alpha)
fitness_alpha= {'perc_fit_traces': 0.0, 'average_trace_fitness': 0.0009999999999999998, 'log_fitness':
0.0002439908135139733}
```

Εικόνα 58 Fitness με την χρήση του petrinet από τον Alpha miner

Ο υπολογισμός του **fitness** με **Heuristic miner** και **Inductive miner** δεν τερματίζει. Από τα παραπάνω αποτελέσματα από τον υπολογισμό του **fitness** με **Alpha miner** φαίνεται πως ο **Alpha miner** δεν είναι κατάλληλη μέθοδος για την εξαγωγή του **process model.**

4.1.2.5 Precision

4.1.2.5.1 Precision με Alpha miner

Το **precision** με την μέθοδο εξαγωγής μοντέλου **alpha miner** βγήκε 0.0 κάτι το οποίο μας οδηγεί στο συμπέρασμα πως η συγκεκριμένη μέθοδος δεν είναι αξιόπιστη και δεν έχει εξάγει σωστό μοντέλο.

```
In [8]: alpha_prec
Out[8]: 0.0
```

Εικόνα 59 Precision με Alpha miner

4.1.2.5.2 Precision με Heuristic miner

```
In [10]: heuristic_prec
Out[10]: 1.0
```

Εικόνα 60 Precision με Heuristic miner

4.1.2.5.3 Precision με Inductive miner

```
In [12]: inductive_prec
Out[12]: 0.0955376419241104
```

Εικόνα 61 Precision με Inductive miner

4.1.3 Συμπέρασμα

Με βάση τις παραπάνω εικόνες η βέλτιστη μέθοδος για την εξαγωγή μοντέλου από τα **dataset** μας είναι αυτή του **Heuristic miner**.

4.1.4 Σύγκριση του dataset της τράπεζας με το dataset του εργοστασίου.

Όπως παρατηρούμε και από τις γραφικές παραστάσεις του **accuracy** πετυχαίνουμε μεγαλύτερο βαθμό **accuracy** στην πρόβλεψη μας όταν κάνουμε προβλέψεις με το **dataset της τράπεζας** από όταν κάνουμε προβλέψεις με το **dataset του εργοστασίου**. Άμα κάνουμε εξαγωγή διεργασιών για τα δύο **datasets** παρατηρούμε πως έχουμε πολύ πιο πολύπλοκες διεργασίες στην περίπτωση του εργοστασίου. Αυτό αποδεικνύεται παίρνοντας το **simplicity** για κάθε dataset, όπως παρακάτω.

4.1.4.1 Simplicity εργοστασίου.

```
simplicity_alpha= 1.0
simplicity_inductive= 0.5837563451776651
simplicity_heuristic= 0.443956043956044
simplicity_heuristic_with_params= 0.45580589254766024
```

Εικόνα 62 Simplicity με Alpha, Inductive και Heuristic miner

4.1.4.2 Simplicity τράπεζας.

```
simplicity_alpha= 1.0
simplicity_inductive= 0.6274509803921569
simplicity_heuristic= 0.5186721991701245
simplicity_heuristic_with_params= 0.5146443514644352
```

Εικόνα 63 Simplicity με Alpha, Inductive και Heuristic miner

Διαπιστώνουμε πως το **simplicity του εργοστασίου** είναι μικρότερο από το **simplicity της τράπεζας**. Αυτός είναι και μια απόδειξη για τον λόγο που πετυχαίνουμε **μεγαλύτερο accuracy** με **το dataset της τράπεζας**. Παρόλαυτα αυτό δεν σημαίνει πως οι προβλέψεις που γίνονται με το **dataset του εργοστασίου** είναι λανθασμένες.

4.1.4.3 Τι είναι το simplicity ;

Το **simplicity** είναι η $4^{η}$ διάσταση ανάλυσης ενός **process model**. Σε αυτή την περίπτωση ορίζουμε το **simplicity** λαμβάνοντας υπόψιν μας μόνο το **Petri net** του μοντέλου μας. Το κριτήριο που χρησιμοποιούμε για το **simplicity** είναι το **inverse arc degree** όπως περιγράφουμε παρακάτω.

Καταρχήν παίρνουμε το **average degree** για ένα **place/transition** του **Petri net** το οποίο ορίζεται σαν το άθροισμα των **input arcs** και των **output arcs**. Αν όλα τα **places** του **Petri net** έχουν τουλάχιστον 1 **input arc** και 1 **output arc**, το **average degree** είναι τουλάχιστον 2. Επιλέγοντας έναν αριθμό **k** μεταξύ του 0 και του απείρου, το **simplicity** βασισμένο στο **inverse arc degree** ορίζεται σαν **1.0/(1.0 + max(mean_degree – k, 0))**.

4.2 Πειράματα με το μέγεθος του dataset.

4.2.1 Πειράματα με το dataset της τράπεζας

Πίνακας με τα activities και τα ακρωνύμια τους

Activity	Activity acronym
A_Create Application	ACRTAPP
A_Submitted	ASBMTD
W_Handle leads	WHNDL
W_Complete application	WCMPLT
A_Concept	ACNCPT
A_Accepted	AACCPTD
O_Create Offer	OCRTOFF
O_Created	OCRTD
O_Sent (mail and online)	OSNTMLON
W_Call after offers	WCAFO
W_Validate application	WVAPP
A_Validating	AVAL
O_Returned	ORET
W_Call incomplete files	WCINCFIL
A_Incomplete	AINC
A_Denied	ADEN
O_Refused	OREF
A_Pending	APEN

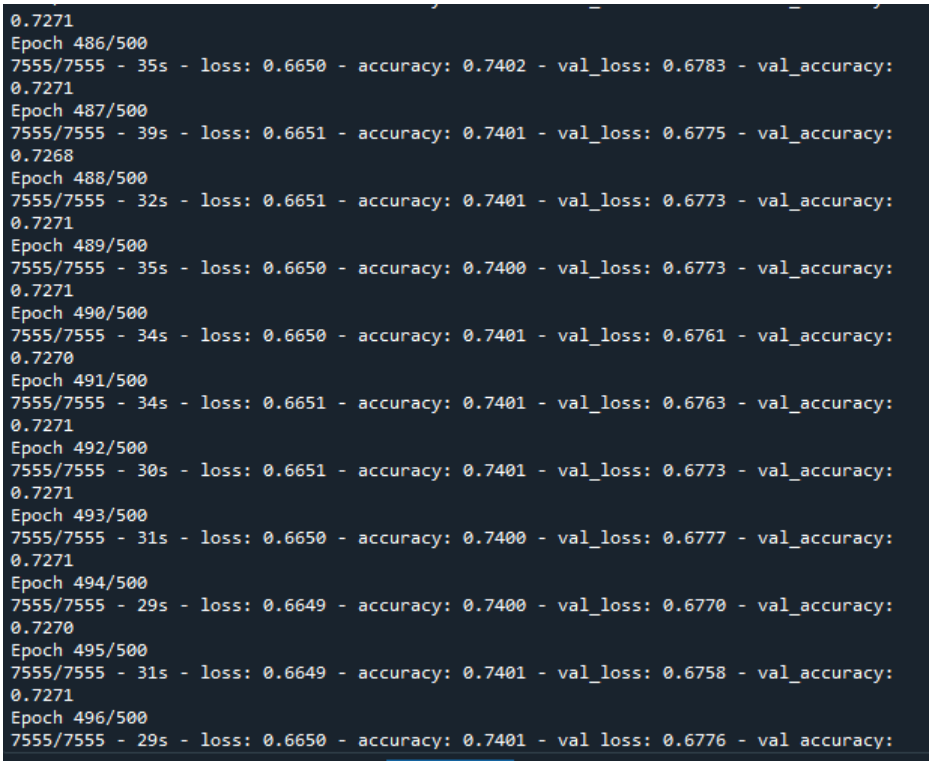
4.2.1.1 Πείραμα με ολόκληρο το Dataset (διάρκεια εκτέλεσης του προγράμματος πάνω από 4 ώρες)

4.2.1.2 Παρατηρήσεις:

Σε αυτή την δοκιμή που πήρα ολόκληρο το **dataset** το πρόγραμμα προέβλεψε σωστά το επόμενο **activity** εκτός από 2 περιπτώσεις, για το activity **AINC** και το activity **WHNDL**. Επίσης να σημειωθεί ότι παραπάνω από μια απαντήσεις – προβλέψεις μπορεί να είναι σωστές. Κάθε φορά αναλόγως με το πλήθος των data που περνάμε στο νευρωνικό παίρνουμε και άλλες προβλέψεις για τα επόμενα activities.

4.2.1.3 Γιατί συμβαίνει αυτό;

Αυτό συμβαίνει ως ένα βαθμό, διότι κάνουμε **overfitting** τα δεδομένα μας στο νευρωνικό μας δίκτυο, παρόλο που έχουμε πάρει όλο το **dataset** και θα περίμενε κανείς να παίρνουμε καλύτερα αποτελέσματα. Δηλαδή το NN έχει μάθει ακριβώς τα δεδομένα του **training dataset** και δεν είναι χρήσιμο στο να γενικεύει και να κάνει προβλέψεις με **inputs**. Η συγκεκριμένη συμπεριφορά οφείλεται στο ότι τα δεδομένα μας επαναλαμβάνονται πολλές φορές και υπάρχουν πολλά **traces** που περιέχουν ίδια **events** με άλλα **traces** αλλά και στο ότι έχουμε πολλά **features,** δηλαδή το μοντέλο μας μπορεί να αναγνωρίσει ένα **data point** από ένα **feature**.



Εικόνα 64 Έξοδος κατά το training του LSTM

4.2.1.4 train accuracy και train loss

Το νευρωνικό δίκτυο υπολογίζει το **train accuracy** και το **train loss** με τον εξής τρόπο. Αφού φορτώσουμε το **input/features** σε αυτό και αρχικοποιηθεί με τυχαία βάρη, το νευρωνικό δίκτυο "μαντεύει" το **label** του επόμενου **activity** και το συγκρίνουμε με το πραγματικό **label** του επόμενου **activity**. Το **error** (η διαφορά μεταξύ της "πρόβλεψης" και του πραγματικού **label**) υπολογίζεται και γίνεται **back propagate**. Δηλαδή για κάθε **batch** κατά το **training** το NN κάνει προβλέψεις και κάνει **update** τα βάρη. Η συγκεκριμένη διαδικασία επαναλαμβάνεται για όλα τα **batches**. Φυσικά η βιβλιοθήκη **Keras** διαχειρίζεται την συγκεκριμένη διαδικασία και βγάζει σαν έξοδο τα **metrics** (συνήθως **accuracy** και **loss**) για κάθε **batch** [34],[35].

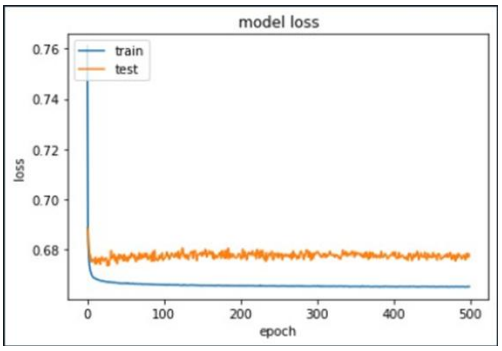
4.2.1.5 Γιατί χρειαζόμαστε το train accuracy και train loss ;

Δεν είναι βαρυσήμαντα **evaluation metrics** διότι ένα νευρωνικό δίκτυο με επαρκείς παραμέτρους μπορεί στην πραγματικότητα να απομνημονεύσει τα **labels**, στην περίπτωση μας τα επόμενα **activities** των **training data** και στην συνέχεια να προβλέπει τα **activities** πάνω σε άγνωστα δεδομένα που δεν έχει γίνει κάποιο **training**. Παρόλαυτα μπορεί να είναι χρήσιμο να παρακολουθούμε τα **accuracy** και **loss** έτσι ώστε να ξέρουμε άμα το **backend** λειτουργεί όπως θα περιμέναμε και αν το **training process** πρέπει να σταματήσει [34],[35].

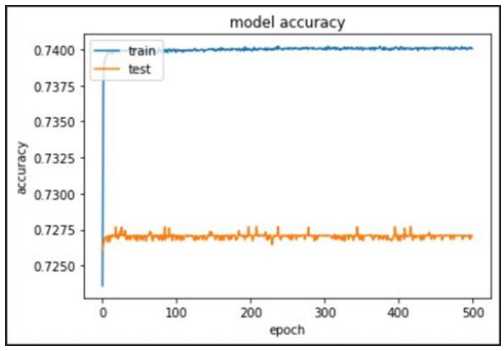
4.2.1.6 Τι είναι τα loss, accuracy, val_loss, val_accuracy ;

Τα **LSTM models** εκπαιδεύονται καλώντας την συνάρτηση **fit()**. Αυτή η συνάρτηση επιστρέφει μια μεταβλητή που λέγετε **history** και περιέχει ένα **trace** του **loss** και οποιαδήποτε άλλου **metric** έχουμε προσδιορίσει. Αυτές οι μετρήσεις καταγράφονται σε κάθε **epoch** και εμφανίζονται σαν "**loss"** και το **accuracy** σαν "**acc**".

Το **Keras** μας επιτρέπει να ορίσουμε ένα ξεχωριστό **validation dataset** ενώ κάνουμε **fit** το μοντέλο μας, το οποίο μπορεί να αξιολογηθεί χρησιμοποιώντας και αυτό τα ίδια **metrics**. Το παραπάνω το επιτυγχάνουμε βάζοντας στην παράμετρο **validation_split** του **fit()** ένα κλάσμα του **training data** ως **validation dataset**. Τα **metrics** που υπολογίζονται στο **validation dataset** φέρουν μπροστά τους το πρόθεμα "**val**_" [36].



Εικόνα 65 Train/Test loss με ολόκληρο το dataset dataset



Εικόνα 66 Train/Test accuracy με ολόκληρο το

Εμφάνιση των προβλέψεων.

```
Results:

acrtapp asbmtd whndl whndl
WCMPLT acncpt wcmplt wcmplt
OCRTOFF ocrtd osntmlon wcmplt
OCRTD wcmplt wcmplt wcmplt
OSNTMLON wcmplt wcafo wcafo
ACNCPT wcmplt wcmplt wcmplt
WVAPP wvapp wvapp wvapp
AACCPTD ocrtoff ocrtd osntmlon
AVAL wvapp wvapp wvapp
AINC aval wcincfil wcincfil
WCINCFIL wcincfil wcincfil wcincfil
ASBMTD whndl whndl wcmplt
WHNDL wcafo acncpt wcmplt
```

Εικόνα 67 Πρόβλεψη τριών επόμενων δραστηριοτήτων

Άμα κάνουμε αναζήτηση στο αρχείο **log_acron** θα παρατηρήσουμε πως το **LSTM** προβλέπει σωστά το επόμενο activity εκτός για το **AINC** και για το **WHNDL**. Άμα πάρουμε οποιοδήποτε άλλο συνδυασμό εισόδου και εξόδου και κάνουμε αναζήτηση στο text file θα πάρουμε σωστό αποτέλεσμα. Από τα παραπάνω βλέπουμε πως υπάρχει θόρυβος στα δύο διαγράμματα, κάτι που μας υποδεικνύει πως το μοντέλο μας απλά μαθαίνει τα δεδομένα τα οποία το εκπαιδεύουμε και δεν θα είναι ικανό να προβλέψει και καινούργια δεδομένα. Στην εμφάνιση των προβλέψεων βλέπουμε πως το **activity** A_Create Application ακολουθείται από το A_Submitted, το W_Complete application από το A_Concept, το O_Create Offer από το O_Created, το O_Created από το W_Complete application κ.τ.λ. Οι μόνες λάθος προβλέψεις είναι αυτές για το A_Incomplete για το οποίο προβλέπεται λανθασμένα ότι ακολουθείται από την επικύρωση αίτησης δανείου A_Validating και για το W_Handle leads για το οποίο προβλέπεται λανθασμένα ότι ακολουθείται από το **activity** W_Call after offers

4.2.1.7 Πείραμα με 20.000 δείγματα.

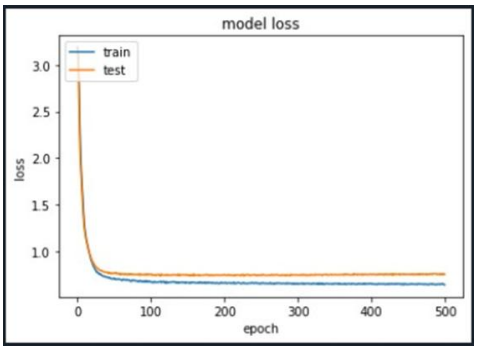
Παρατηρήσεις:

Αν κοιτάξουμε την γραφική παράσταση που δείχνει το **training** και **testing accuracy** θα παρατηρήσουμε πως υπάρχουν σημάδια **overfitting**. Παρόλαυτα παίρνουμε σωστές προβλέψεις. Όπως και στην προηγούμενη δοκιμή, περισσότερες από μια απαντήσεις είναι σωστές. Παρατηρούμε πως το **test accuracy** είναι πολύ ασταθές.

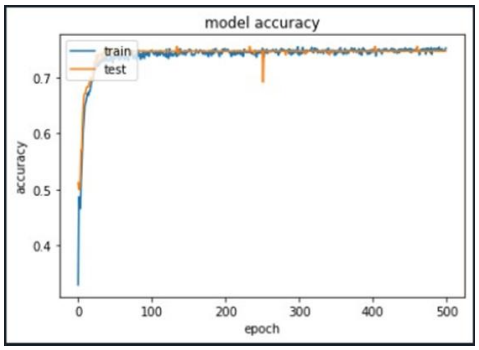
```
19/19 - 0s - loss: 0.6514 - accuracy: 0.7479 - val_loss: 0.7604 - val_accuracy: 0.7466
Epoch 483/500
19/19 - 0s - loss: 0.6454 - accuracy: 0.7508 - val_loss: 0.7553 - val_accuracy: 0.7466
Epoch 484/500
19/19 - 0s - loss: 0.6515 - accuracy: 0.7508 - val_loss: 0.7620 - val_accuracy: 0.7466
Epoch 485/500
19/19 - 0s - loss: 0.6479 - accuracy: 0.7513 - val_loss: 0.7572 - val_accuracy: 0.7466
Epoch 486/500
19/19 - 0s - loss: 0.6465 - accuracy: 0.7479 - val_loss: 0.7610 - val_accuracy: 0.7466
Epoch 487/500
19/19 - 0s - loss: 0.6490 - accuracy: 0.7504 - val_loss: 0.7611 - val_accuracy: 0.7466
Epoch 488/500
19/19 - 0s - loss: 0.6430 - accuracy: 0.7517 - val_loss: 0.7586 - val_accuracy: 0.7466
Epoch 489/500
19/19 - 0s - loss: 0.6468 - accuracy: 0.7500 - val_loss: 0.7579 - val_accuracy: 0.7466
Epoch 490/500
19/19 - 0s - loss: 0.6487 - accuracy: 0.7479 - val_loss: 0.7521 - val_accuracy: 0.7466
Epoch 491/500
19/19 - 0s - loss: 0.6449 - accuracy: 0.7538 - val_loss: 0.7569 - val_accuracy: 0.7466
Epoch 492/500
19/19 - 0s - loss: 0.6496 - accuracy: 0.7483 - val_loss: 0.7552 - val_accuracy: 0.7466
Epoch 493/500
19/19 - 0s - loss: 0.6476 - accuracy: 0.7479 - val_loss: 0.7543 - val_accuracy: 0.7466
Epoch 494/500
19/19 - 0s - loss: 0.6497 - accuracy: 0.7492 - val_loss: 0.7490 - val_accuracy: 0.7466
Epoch 495/500
19/19 - 0s - loss: 0.6457 - accuracy: 0.7508 - val_loss: 0.7614 - val_accuracy: 0.7466
Epoch 496/500
19/19 - 0s - loss: 0.6524 - accuracy: 0.7471 - val_loss: 0.7578 - val_accuracy: 0.7466
Epoch 497/500
19/19 - 0s - loss: 0.6496 - accuracy: 0.7462 - val_loss: 0.7517 - val_accuracy: 0.7466
Epoch 498/500
19/19 - 0s - loss: 0.6522 - accuracy: 0.7475 - val_loss: 0.7573 - val_accuracy: 0.7466
```

Εικόνα 68 Έξοδος κατά το training του LSTM

Παρακάτω βλέπουμε τα διαγράμματα για το **training** και **testing accuracy** και για το training και testing loss. Παρατηρούμε πως υπάρχουν σημάδια **overfitting**. Η γραφική παράσταση του **test** αντιγράφει τον θόρυβο του **train**.

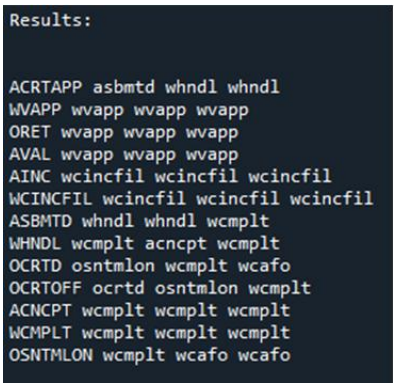


Εικόνα 69 Train/Test loss με 20.000 δείγματα



Εικόνα 70 Train/Test accuracy με 20.000 δείγματα

Εμφάνιση των προβλέψεων.



Εικόνα 71 Προβλέψεις για 20.000 δείγματα

Όλα τα αποτελέσματα-προβλέψεις είναι ακριβή. Παρατηρούμε από την γραφική παράσταση της **Εικόνας 33** πως το **test_loss** συνεχίζει να αυξάνεται όσο περνάνε τα **epochs**. Επίσης αν παρατηρήσουμε την **Εικόνα 32** θα δούμε πως ενώ το **test loss** αυξάνεται, το **test accuracy** παραμένει το ίδιο. Η συγκεκριμένη συμπεριφορά μπορεί να είναι αποτέλεσμα λάθος προβλέψεων από το μοντέλο μας στο **test set** κάτι το οποίο τελικά μας οδηγεί στο συμπέρασμα πως το μοντέλο μας κάνει **overfitting** στα δεδομένα εκπαίδευσης και δεν γενικοποιεί, όπως θα έπρεπε στα δεδομένα ελέγχου. Όλα τα αποτελέσματα παραπάνω παρόλο που είναι ακριβή, δεν υποδεικνύουν και την σωστή συμπεριφορά του μοντέλου μας σε καινούργια δεδομένα. Κάποια από τα **activity** για τα οποία έχει προβλεφτεί σωστά το επόμενο τους είναι το O_Create Offer που ακολουθείται από το O_Created, το W_Handle leads το οποίο έπεται από την δραστηριότητα W_Complete application κ.α.

4.2.1.8 Πείραμα με 50.000 δείγματα.

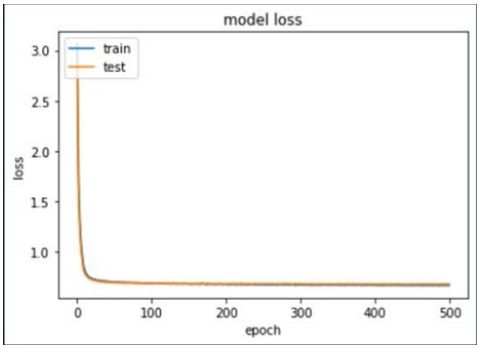
Παρατηρήσεις:

Άμα αυξήσουμε τον αριθμό των δειγμάτων που παίρνουμε για το **training** θα παρατηρήσουμε πως θα μειωθούν τα σημάδια του **overfitting**, όπως θα δούμε και από την γραφική παράσταση του **accuracy** για το **training** και το **testing**.

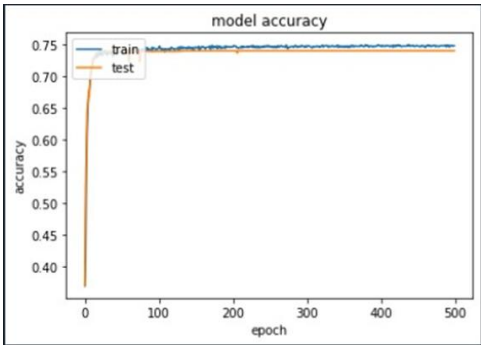
```
47/47 - 0s - loss: 0.6660 - accuracy: 0.7473 - val_loss: 0.6758 - val_accuracy: 0.7392
Epoch 485/500
47/47 - 0s - loss: 0.6621 - accuracy: 0.7473 - val_loss: 0.6768 - val_accuracy: 0.7392
Epoch 486/500
47/47 - 0s - loss: 0.6644 - accuracy: 0.7471 - val_loss: 0.6789 - val_accuracy: 0.7392
Epoch 487/500
47/47 - 0s - loss: 0.6620 - accuracy: 0.7471 - val_loss: 0.6757 - val_accuracy: 0.7392
Epoch 488/500
47/47 - 0s - loss: 0.6634 - accuracy: 0.7479 - val_loss: 0.6765 - val_accuracy: 0.7392
Epoch 489/500
47/47 - 0s - loss: 0.6660 - accuracy: 0.7456 - val_loss: 0.6763 - val_accuracy: 0.7392
Epoch 490/500
47/47 - 0s - loss: 0.6647 - accuracy: 0.7483 - val_loss: 0.6779 - val_accuracy: 0.7392
Epoch 491/500
47/47 - 0s - loss: 0.6638 - accuracy: 0.7478 - val_loss: 0.6787 - val_accuracy: 0.7392
Epoch 492/500
47/47 - 0s - loss: 0.6643 - accuracy: 0.7457 - val_loss: 0.6794 - val_accuracy: 0.7392
Epoch 493/500
47/47 - 0s - loss: 0.6653 - accuracy: 0.7481 - val_loss: 0.6770 - val_accuracy: 0.7392
Epoch 494/500
47/47 - 0s - loss: 0.6652 - accuracy: 0.7456 - val_loss: 0.6776 - val_accuracy: 0.7392
Epoch 495/500
47/47 - 0s - loss: 0.6651 - accuracy: 0.7464 - val_loss: 0.6769 - val_accuracy: 0.7392
Epoch 496/500
47/47 - 0s - loss: 0.6659 - accuracy: 0.7476 - val_loss: 0.6787 - val_accuracy: 0.7392
Epoch 497/500
47/47 - 0s - loss: 0.6663 - accuracy: 0.7478 - val_loss: 0.6778 - val_accuracy: 0.7392
Epoch 498/500
47/47 - 0s - loss: 0.6634 - accuracy: 0.7469 - val_loss: 0.6768 - val_accuracy: 0.7392
Epoch 499/500
47/47 - 0s - loss: 0.6641 - accuracy: 0.7466 - val_loss: 0.6781 - val_accuracy: 0.7392
Epoch 500/500
47/47 - 0s - loss: 0.6658 - accuracy: 0.7473 - val_loss: 0.6781 - val_accuracy: 0.7392
```

Εικόνα 72 Έξοδος κατά το training του LSTM

Παρακάτω βλέπουμε τα διαγράμματα για το **training** και **testing accuracy** και για το **training** και **testing loss**.



Εικόνα 74 Train/Test loss για 50.000 δείγματα



Εικόνα 75 Train/test accuracy για 50.000 δείγματα

Διαπιστώνουμε πως η γραμμή του **test** δεν είναι ακριβώς πάνω στην γραμμή του **train** και δεν έχει αντιγράψει τον θόρυβο του **train** επίσης.

Εμφάνιση των προβλέψεων.

```
Results:


ACRTAPP asbmtd whndl whndl
WVAPP wvapp wvapp wvapp
ORET wvapp wvapp wvapp
AVAL wvapp wvapp wvapp
AINC wcincfil wcincfil wcincfil
WCINCFIL wcincfil wcincfil wcincfil
ASBMTD whndl whndl wcmplt
WHNDL wcmplt acncpt wcmplt
OCRTD ocrtd osntmlon wcmplt
OCRTOFF ocrtd osntmlon wcmplt
ACNCPT wcmplt wcmplt wcmplt
WCMPLT wcmplt wcmplt wcmplt
OSNTMLON wcmplt wcafo wcafo
```

Εικόνα 76 Προβλέψεις για 50.000 δείγματα

Η μόνη λάθος πρόβλεψη είναι αυτή που για επόμενο **activity** του O_Created προβλέπει το ίδιο **activity** δηλαδή το O_Created ενώ το σωστό είναι το **activity** με όνομα O_Sent (mail and online). Για τις υπόλοιπες δραστηριότητες, τα **activities** που έχουν προβλεφτεί είναι σωστά. Για παράδειγμα μετά την δραστηριότητα A_Incomplete ακολουθεί όντως η δραστηριότητα W_Call incomplete files, μετά την δραστηριότητα A_Submitted ακολουθεί η δραστηριότητα W_Handle leads. Όπως και στο προηγούμενο πείραμα παρατηρούμε από την Εικόνα 36 πως το **test loss** αυξάνει όσο περνάνε **epochs**.

4.2.1.9 Πείραμα με 80.000 δείγματα.

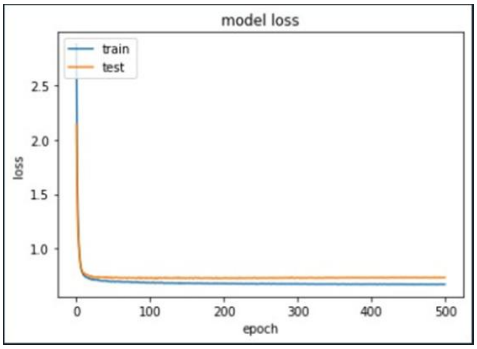
Παρατηρήσεις:

Για 80.000 δείγματα, όπως θα παρατηρήσουμε και από τα διαγράμματα του **accuracy** και του **loss**, μειώνονται τα σημάδια του **overfitting** περισσότερο και επιπλέον όλες οι προβλέψεις είναι σωστές. Αυτό το αποτέλεσμα μας οδηγεί στην υπόθεση πως όσο αυξάνουμε το πλήθος των δεδομένων που θα δίνουμε σαν είσοδο στο νευρωνικό μας δίκτυο τόσο καλύτερα μπορεί να εκπαιδευτεί και να κάνει σωστές προβλέψεις με καινούργια δεδομένα. Οδηγούμαστε σε αυτό το συμπέρασμα διότι όπως είπαμε και παραπάνω υπάρχουν κάποιες περιπτώσεις που περισσότερα **training data** οδηγούν σε ακριβέστερες προβλέψεις. Συγκεκριμένα αυτό συμβαίνει όταν έχουμε ιδιόμορφα **data**. Αντίθετα στην δικιά μας περίπτωση και στο **dataset** της τράπεζας αλλά και στο **dataset** του εργοστασίου τα δεδομένα μας είναι επαναλαμβανόμενα.

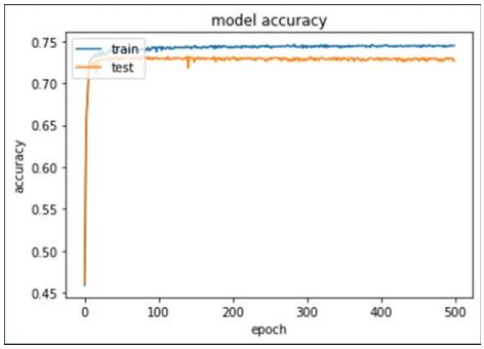
```
Epoch 485/500
75/75 - 0s - loss: 0.6674 - accuracy: 0.7450 - val_loss: 0.7318 - val_accuracy: 0.7268
Epoch 486/500
75/75 - 0s - loss: 0.6689 - accuracy: 0.7432 - val_loss: 0.7297 - val_accuracy: 0.7280
Epoch 487/500
75/75 - 0s - loss: 0.6688 - accuracy: 0.7433 - val_loss: 0.7289 - val_accuracy: 0.7280
Epoch 488/500
75/75 - 0s - loss: 0.6685 - accuracy: 0.7449 - val_loss: 0.7291 - val_accuracy: 0.7297
Epoch 489/500
75/75 - 0s - loss: 0.6671 - accuracy: 0.7442 - val_loss: 0.7294 - val_accuracy: 0.7297
Epoch 490/500
75/75 - 0s - loss: 0.6661 - accuracy: 0.7446 - val_loss: 0.7283 - val_accuracy: 0.7297
Epoch 491/500
75/75 - 0s - loss: 0.6684 - accuracy: 0.7439 - val_loss: 0.7300 - val_accuracy: 0.7297
Epoch 492/500
75/75 - 0s - loss: 0.6682 - accuracy: 0.7437 - val_loss: 0.7293 - val_accuracy: 0.7280
Epoch 493/500
75/75 - 0s - loss: 0.6676 - accuracy: 0.7443 - val_loss: 0.7292 - val_accuracy: 0.7297
Epoch 494/500
75/75 - 0s - loss: 0.6686 - accuracy: 0.7452 - val_loss: 0.7311 - val_accuracy: 0.7297
Epoch 495/500
75/75 - 0s - loss: 0.6701 - accuracy: 0.7445 - val_loss: 0.7294 - val_accuracy: 0.7297
Epoch 496/500
75/75 - 0s - loss: 0.6673 - accuracy: 0.7451 - val_loss: 0.7293 - val_accuracy: 0.7297
Epoch 497/500
75/75 - 0s - loss: 0.6688 - accuracy: 0.7444 - val_loss: 0.7296 - val_accuracy: 0.7297
Epoch 498/500
75/75 - 0s - loss: 0.6678 - accuracy: 0.7445 - val_loss: 0.7307 - val_accuracy: 0.7276
Epoch 499/500
75/75 - 0s - loss: 0.6684 - accuracy: 0.7448 - val_loss: 0.7307 - val_accuracy: 0.7289
Epoch 500/500
75/75 - 0s - loss: 0.6679 - accuracy: 0.7450 - val_loss: 0.7301 - val_accuracy: 0.7268
```

Εικόνα 77 Έξοδος κατά το training του LSTM

Παρακάτω βλέπουμε τα διαγράμματα για το **training** και **testing accuracy** και για το **training** και **testing loss**. Εξακολουθεί να υπάρχει **overfitting**, αυτό δεν σημαίνει όμως πως είναι συνέπεια του πλήθους των δεδομένων με το οποίο εκπαιδεύουμε το νευρωνικό μας δίκτυο. Και άλλοι παράμετροι του **LSTM**, όπως τα **epochs** παίζουν ρόλο στην ορθή λειτουργία του μοντέλου μας.



Εικόνα 78 Train/Test loss για 80.000 δείγματα



Εικόνα 79 Train/Test accuracy για 80.000 δείγματα

Εμφάνιση των προβλέψεων.

```
Results:


ACRTAPP asbmtd whndl whndl
WVAPP wvapp wvapp wvapp
ORET wvapp wvapp wvapp
AVAL wvapp wvapp wvapp
AINC wcincfil wcincfil wcincfil
WCINCFIL wcincfil wcincfil wcincfil
ASBMTD whndl whndl wcmplt
WHNDL wcmplt acncpt wcmplt
OCRTD osntmlon wcmplt wcafo
OCRTOFF ocrtd osntmlon wcmplt
ACNCPT wcmplt wcmplt wcmplt
WCMPLT wcmplt wcmplt wcmplt
OSNTMLON wcmplt wcafo wcafo
```

Εικόνα 80 Προβλέψεις για 80.000 δείγματα

Όλα τα αποτελέσματα-προβλέψεις είναι ακριβή. Για παράδειγμα η δραστηριότητα O_Created ακολουθείται από την O_Sent (mail and online), η O_Create Offer από την O_Created, η A_Concept από την W_Complete application κ.τ.λ.

4.3 Πειράματα με τον αριθμό των epochs.

4.3.1 Πειράματα με το dataset της τράπεζας

Παρακάτω βλέπουμε τον πίνακα με τις μέσες τιμές που πήραμε για το **accuracy**, το **test accuracy**, το **loss**, το **test loss** καθώς και το πόσες λάθος προβλέψεις έκανε το νευρωνικό μας δίκτυο.

Μπορούμε εύκολα να παρατηρήσουμε πως όσο αυξάνουμε τα **epochs** το **test accuracy** παραμένει σταθερό ενώ το **accuracy** αυξάνετε, έστω και ελάχιστα. Αυτό είναι απόδειξη πως το μοντέλο μας αντί να γενικοποιεί (**generalize**) τα δεδομένα μας έχει αρχίσει και τα απομνημονεύει (**overfit**). Γενικότερα αυτό που θα θέλαμε είναι το **test accuracy** να συνεχίσει να αυξάνετε.

Επιπρόσθετα άμα παρατηρήσουμε το **training loss** και το **validation loss** θα δούμε πως σε κάθε περίπτωση το **loss** είναι μικρότερο από το **validation loss**. Αυτό είναι μια ακόμα απόδειξη πως κάνουμε **overfit** το μοντέλο μας και πως αντί να είναι ικανό να κάνει προβλέψεις με καινούργια δεδομένα, απλά ‘’μαθαίνει’’ τα δεδομένα εκπαίδευσης.

Epochs	**Mean Test Accuracy**	**Mean Accuracy**	**Mean Test Loss**	**Mean Loss**	**Προβλέψεις**
100	0.7213871338963509	0.7296174055337906	0.768311607837677	0.7486629199981689	Όλες οι προβλέψεις είναι σωστές
200	0.7246574155 98631	0.7355491909 384727	0.7482277685 403824	0.7146643215 417862	Όλες οι προβλέψεις είναι

					σωστές
300	0.726324782371521	0.737962651153405 5	0.739731250206629 4	0.701287850141525 3	Όλες οι προβλέψεις είναι σωστές
400	0.726643543764948 9	0.740111677125096 3	0.738531585335731 5	0.692839180678129 2	Όλες οι προβλέψεις είναι σωστές
500	0.727476241827011 1	0.741042673408985 1	0.735074776291847 3	0.687797622799873 3	1 λάθος πρόβλεψη
600	0.726581190675497 1	0.741175109992424 6	0.734136823912461 6	0.685346744457880 7	1 λάθος πρόβλεψη
700	0.727681495164122	0.741965375627790 2	0.731255282248769 5	0.682709236826215 4	Όλες οι προβλέψεις είναι σωστές
800	0.727861488312482 9	0.742314353436231 6	0.735566273927688 6	0.680121630728244 8	Όλες οι προβλέψεις είναι σωστές
900	0.727952914966477 3	0.742598161101341 2	0.729587524268362 3	0.678668625619676 3	Όλες οι προβλέψεις είναι σωστές
1000	0.727984867185354 3	0.742858945339918 1	0.729738991081714 6	0.677655458390712 7	Όλες οι προβλέψεις είναι σωστές

4.3.2 Πειράματα με το dataset του εργοστασίου

Όπως παρατηρούμε από τον παρακάτω πίνακα με το **dataset** του εργοστασίου λόγω και του μικρού του μεγέθους δεν παίρνουμε και τόσο καλά αποτελέσματα. Όπως βλέπουμε πετυχαίνουμε **test accuracy** κάτω από το 50%, δηλαδή η προβλέψεις που κάνει το μοντέλο μας σε καινούργια δεδομένα είναι πάνω από τις μισές λανθασμένες.

Και στην περίπτωση του συγκεκριμένου **dataset** παρατηρούμε πως κάνουμε **overfit** το μοντέλο μας. Όσο προσθέτουμε **epochs** το **training accuracy** αυξάνεται, ενώ το **test accuracy** μεταβάλλεται ελάχιστα και παραμένει σταθερά μικρότερο από το **training accuracy**. Δηλαδή το μοντέλο μας δεν αποδίδει καλά στα καινούργια δεδομένα και απλά έχει απομνημονεύσει τα δείγματα του **training set**.

Επίσης όπως και την περίπτωση της τράπεζας παρατηρούμε πως το **training loss** είναι μικρότερο από το **val loss** και μικραίνει όσο προσθέτουμε **epochs**. Αντίθετα το **test loss** αυξάνεται, κάτι που μας αποδεικνύει πως κάνουμε **overfit**. Το μοντέλο μας ‘’μαθαίνει’’ πολύ καλά τα δεδομένα εκπαίδευσης αντί να εκπαιδευτεί στο να αναγνωρίζει τα μοτίβα των γεγονότων ώστε να κάνει προβλέψεις σε νέα δεδομένα.

Epochs	Mean Test Accuracy	Mean Accuracy	Mean Test Loss	Mean Loss	Προβλέψεις
100	0.4603300315141678	0.472268722653389	2.0165539717674257	1.7341770458221435	Όλες οι προβλέψεις είναι σωστές
200	0.4740594059973955	0.4969273122400045	2.0121624493598937	1.6108620780706406	Όλες οι προβλέψεις είναι σωστές
300	0.4735753564 5365717	0.5054854968686898	2.0238323815663657	1.5512152985 73176	Όλες οι προβλέψεις είναι σωστές
400	0.4758470846712589	0.5116671251133085	2.0660510167479513	1.5202170661091805	Όλες οι προβλέψεις είναι σωστές
500	0.4767832773 3278275	0.5151651971 34018	2.0994630885124206	1.4903638081550599	Όλες οι προβλέψεις είναι σωστές
600	0.4806343960 2653185	0.5171576720724503	2.1108089242 37887	1.4750303576 38995	Όλες οι προβλέψεις είναι σωστές

700	0.4748436273740871	0.5193089197150299	2.1410712942055294	1.4575200756958553	Όλες οι προβλέψεις είναι σωστές
800	0.47480197947472336	0.520757502540946	2.136001325249672	1.4480325144529342	Όλες οι προβλέψεις είναι σωστές
900	0.4740850741995706	0.5217226500146919	2.145725140571594	1.441628613471985	Όλες οι προβλέψεις είναι σωστές
1000	0.4755819578319788	0.5222304514795542	2.165612400531769	1.432954288442474	Όλες οι προβλέψεις είναι σωστές

4.4 Πειράματα με το batch size

Κάνουμε δοκιμές με τον αριθμό του **batch size**. Εκπαιδεύουμε το NN μας με τις τιμές που περιέχονται στη λίστα **batch_sizes = [1, 32, 64, 128, 256, 450]**. Αφού κάνουμε **fit** το μοντέλο μας, σχεδιάζουμε το **train** και **test loss** και **accuracy**. Κάθε **line curve** που σχεδιάζεται μας λέει τρία πράγματα. Πόσο γρήγορα το μοντέλο μαθαίνει το πρόβλημα (τα δεδομένα μας), πόσο καλά έχει καταλάβει το πρόβλημα και το πόσο θόρυβο είχαμε σε κάθε **update** των βαρών κατά το **training**. Με αυτά τα πειράματα έχουμε σκοπό να βρούμε το βέλτιστο μέγεθος του **batch size** αλλά και να επαναπροσδιορίσουμε τον αριθμό των **epochs**.

Κάθε **batch** χρησιμοποιείται για την εκπαίδευση του νευρωνικού μας δικτύου, κατά συνέπεια η επιλογή του βέλτιστου **batch-size** είναι μεγάλης σημασίας λόγω του ότι μπορεί να έχει αντίκτυπο στην απόδοση του μοντέλου μας (στην ακρίβεια των προβλέψεων).

4.4.1 Πειράματα για το dataset της τράπεζας

Όπως παρατηρούμε από τον παρακάτω πίνακα το μοντέλο μας αποδίδει καλύτερα όταν χρησιμοποιούμε **Batch Gradient Descent** (1 < batch size < len(dataset)) από όταν χρησιμοποιούμε **Stochastic Gradient Descent** (batch size = 1). Όταν χρησιμοποιούμε **Stochastic Gradient Descent** το **training process** γίνεται σημαντικά πιο αργό από όταν χρησιμοποιήσαμε **Batch Gradient Descent**. Επίσης παίρνουμε μια λάθος πρόβλεψη, αντίθετα με τα

άλλα πειράματα. Θα επιλέξουμε το 32 για την τιμή του **batch size** διότι δεν παίρνουμε καμία λάθος πρόβλεψη, ο χρόνος εκπαίδευσης του NN μειώνεται σημαντικά και το **training loss** είναι περίπου ίσο με το **validation loss** κάτι που μας υποδεικνύει πως δεν κάνουμε **overfitting** [30].

Batch size	**Accuracy**	**Val_Accuracy**	**Loss**	**Val_Loss**	**Λάθος Προβλέψεις**
1	0.724	0.727	0.7520	0.7526	1
32	0.703	0.712	0.895	0.828	Όλες οι προβλέψεις είναι σωστές
64	0.686	0.691	1.008	0.909	Όλες οι προβλέψεις είναι σωστές
128	0.656	0.663	1.206	1.095	Όλες οι προβλέψεις είναι σωστές
256	0.605	0.616	1.529	1.428	Όλες οι προβλέψεις είναι σωστές

4.4.2 Πειράματα με το dataset του εργοστασίου

Όπως διαπιστώνουμε από τον παρακάτω πίνακα ο βέλτιστος αριθμός για το **batch size** είναι **32**. Καταλήγουμε σε αυτό το συμπέρασμα διότι παρατηρούμε πως μόνο χρησιμοποιώντας την συγκεκριμένη τιμή για το **batch size** το **training loss** είναι περίπου ίσο με το **validation loss** και δηλαδή δεν υπάρχει ούτε **underfitting** ούτε **overfitting.** Οι μικρές τιμές που πετυχαίνουμε στο **training accuracy** αλλά και στο **testing accuracy** είναι λόγω του μικρού μεγέθους του **dataset**.

Batch size	**Accuracy**	**Val_Accuracy**	**Loss**	**Val_Loss**	**Λάθος προβλέψεις**
1	0.466	0.471	1.764	1.964	1
32	0.403	0.416	2.107	2.134	2
64	0.361	0.372	2.332	2.336	1
128	0.316	0.319	2.637	2.646	2
256	0.221	0.190	3.131	3.149	3

4.5 Πειράματα με layers και epochs

Το **πρώτο layer** του NN πρέπει να προσδιορίζει τον αριθμό των inputs που περιμένει. Το **input** πρέπει να είναι **3 διαστάσεων** και να αποτελείται από **samples**, **timesteps** και **features**.

- Samples: Οι γραμμές των δεδομένων μας.

- Timesteps: Οι προηγούμενες παρατηρήσεις (observations) για ένα feature.
- Features: Οι στήλες των δεδομένων μας.

Τα επόμενα layers μπορούν να είναι **stacked** προσθέτοντας τα στο Sequential model. Όταν προσθέτουμε layers το κάθε layer πρέπει να έχει σαν output ένα sequence ώστε τα επόμενα layers να έχουν την 3D είσοδο που απαιτείται.

4.5.1 Input layer

Κάθε νευρωνικό δίκτυο έχει το στρώμα-επίπεδο (**layer**) εισόδου. Όσον αφορά τον αριθμό των νευρώνων που απαρτίζουν το συγκεκριμένο **layer**, αυτή η παράμετρος μπορεί να οριστεί όταν γνωρίζουμε το σχήμα τον δεδομένων εκπαίδευσης του νευρωνικού. Συγκεκριμένα ο αριθμός των νευρώνων του **input layer** είναι ίσος με τον αριθμό τον χαρακτηριστικών (**features**) στα δεδομένα μας.

4.5.2 Πειράματα με το dataset της τράπεζας

Παρακάτω βλέπουμε τον πίνακα με τα αποτελέσματα που μας έδωσε το μοντέλο μας ανά τον αριθμό των στρωμάτων τον οποίο χρησιμοποίησε.

Όπως παρατηρούμε όσο αυξάνουμε τον αριθμό των στρωμάτων στο νευρωνικό μας το **training accuracy** ελαττώνεται. Το ίδιο συμβαίνει επίσης και με το **validation accuracy**. Επίσης το ίδιο μοτίβο ακολουθούν και τα **training loss** και **validation loss** (αυξάνονται). Το παρακάτω είναι κλασσικό παράδειγμα **overfitting**.

Layers και Epochs	**Accuracy**	**Val_Accuracy**	**Loss**	**Val_Loss**	**Λάθος προβλέψεις**
2 layers και 10 epochs	0.618	0.629	1.268	1.170	1
3 layers και 10 epochs	0.567	0.596	1.416	1.324	2
4 layers και 10 epochs	0.535	0.571	1.598	1.519	6
5 layers και 10 epochs	0.504	0.514	1.670	1.576	1
6 layers και 10 epochs	0.492	0.509	1.721	1.639	3

4.5.2.6 Συμπέρασμα για το dataset της τράπεζας

Εικόνα 81 Σχεδιάγραμμα που φαίνεται πότε αρχίζουμε να κάνουμε overfitting

Τελικά διαπιστώνουμε πως επιτρέποντας στο μοντέλο να συνεχίσει το **training** (για περισσότερα **epochs**) αυξάνεται η πιθανότητα τα **weights** και τα **biases** να ρυθμιστούν (**tuned**) σε τέτοιο βαθμό που το μοντέλο να μην αποδίδει καλά σε άγνωστα δεδομένα (**test/validation data**). Το μοντέλο απλά απομνημονεύει το **training set** αντί να ανακαλύψει τις συσχετίσεις και τα μοτίβα που υπάρχουν μεταξύ των **events** στο **dataset.**

Ο μεγάλος αριθμός **epochs** μπορεί να αυξήσει το **training accuracy** αλλά αυτό δεν σημαίνει απαραίτητα πως οι προβλέψεις του μοντέλου πάνω σε καινούργια δεδομένα θα είναι ακριβείς, αντιθέτως πολύ συχνά οι προβλέψεις γίνονται χειρότερες. Για να το αποτρέψουμε αυτό χρησιμοποιούμε ένα **test data set** και παρακολουθούμε το **test accuracy** κατά το **training**. Αυτό μας επιτρέπει στο να συμπεράνουμε αν το μοντέλο μας κάνει ακριβείς προβλέψεις σε καινούργια δεδομένα.

Μπορούμε να χρησιμοποιήσουμε το **early stopping** το οποίο σταματάει το **training** του μοντέλου όταν το **test accuracy** έχει σταματήσει να αυξάνετε μετά από έναν μικρό αριθμό **epochs**. Το **early stopping** μπορεί να θεωρηθεί σαν μια ακόμα τεχνική κανονικοποίησης.

Όσον αφορά τον αριθμό των **layers** στην περίπτωση του **dataset** της τράπεζας παρατηρούμε πως με **ένα layer** όλες οι προβλέψεις είναι σωστές. Όταν αυξάνουμε τον αριθμό των **layers** ο αριθμός των λάθος προβλέψεων αυξομειώνεται. Τα **stacked LSTM NN** πολλές φορές μπορεί να είναι δύσκολο να εκπαιδευτούν για να μας δίνουν σωστά αποτελέσματα. Με την αφαίρεση ενός layer μπορούμε να κάνουμε το μοντέλο μας απλούστερο, να αποφύγουμε το **overfitting** και να πετύχουμε καλύτερα αποτελέσματα.

4.5.3 Πειράματα με το dataset του εργοστασίου

Όπως παρατηρούμε από τον παρακάτω πίνακα όσο προσθέτουμε στρώματα στο μοντέλο μας, το **training accuracy** και το **validation accuracy** μειώνονται. Αντίθετα

το **training loss** και το **validation loss** αυξάνονται, δηλαδή όπως βλέπουμε από το **validation loss** το μοντέλο μας δεν αποδίδει καλά στα καινούργια δεδομένα. Είναι φανερό πως κάνουμε **overfit** το μοντέλο μας.

Layers και Epochs	Accuracy	Val_Accuracy	Loss	Val_Loss	Λάθος προβλέψεις
2 layers και 10 epochs	0.391	0.411	2.108	2.202	1
3 layers και 10 epochs	0.275	0.288	2.427	2.447	2
4 layers και 10 epochs	0.243	0.213	2.496	2.572	5
5 layers και 10 epochs	0.228	0.197	2.559	2.624	5
6 layers και 10 epochs	0.230	0.187	2.519	2.645	3

4.5.3.8 Συμπέρασμα για το dataset του εργοστασίου

Παρατηρούμε από τα παραπάνω αποτελέσματα πως η προσθήκη επιπλέον **hidden layers** σε συνδυασμό με το μικρό μέγεθος του **dataset** μας προκαλεί **overfitting** στο μοντέλο μας και αυξάνει το πλήθος των λανθασμένων προβλέψεων αντί να βελτιώνει την απόδοση του νευρωνικού δικτύου μας.

4.6 Πειράματα με το πλήθος των Neurons

4.6.1 Επιλογή του βέλτιστου πλήθους νευρώνων

Δεν υπάρχει κάποιος καθιερωμένος κανόνας για το πλήθος των νευρώνων που πρέπει να χρησιμοποιούμε. Τις περισσότερες φορές αυτό το πλήθος καθορίζεται μέσα από τον πειραματισμό και την εξαγωγή αποτελεσμάτων. Η συνήθης τεχνική για τον πειραματισμό για την λήψη αποτελεσμάτων ώστε να προσδιορίσουμε παραμέτρους όπως των αριθμό των νευρώνων είναι το **k-fold-cross-validation**. Ένας ενδεικτικός τύπος για την επιλογή αυτού του αριθμού είναι ο παρακάτω:

$$N_h = \frac{N_s}{(\alpha * (N_i + N_o))}$$

Εικόνα 82 Τύπος για τον κατάλληλο αριθμό νευρώνων

Το N_i είναι ο αριθμός των νευρώνων εισόδου, το N_o είναι ο αριθμός των νευρώνων εξόδου, το N είναι ο αριθμός των δειγμάτων στα δεδομένα εκπαίδευσης και το **α** αναπαριστά έναν παράγοντα κλιμάκωσης συνήθως μεταξύ του 2 και του 10. Άμα το πρόβλημα είναι απλό και ο χρόνος πρέπει να λαμβάνεται υπόψιν, υπάρχουν διάφοροι άλλοι κανόνες για τον προσδιορισμό του αριθμού των νευρώνων, οι οποίοι επικεντρώνονται στους νευρώνες εισόδου και εξόδου. Παρόλο που αυτοί οι κανόνες είναι εύκολοι στην χρήση τους, σπάνια θα οδηγήσουν σε βέλτιστα αποτελέσματα. Παρακάτω βλέπουμε έναν τέτοιο κανόνα:

$$N_h = \frac{2}{3} * (N_i + N_o)$$

Εικόνα 83 Τύπος για τον κατάλληλο αριθμό νευρώνων

4.6.2 Αριθμός των νευρώνων στα ενδιάμεσα layers

Η επιλογή του σωστού αριθμού νευρώνων στα ενδιάμεσα **layers** είναι πολύ σημαντικό μέρος της συνολικής αρχιτεκτονικής τους νευρωνικού μας δικτύου. Παρόλο που αυτά τα στρώματα δεν επικοινωνούν απευθείας με το εξωτερικό περιβάλλον, έχουν μεγάλη συνεισφορά στο τελικό αποτέλεσμα. Η χρήση πολύ λίγων νευρώνων στα ενδιάμεσα στρώματα θα οδηγήσει σε **underfitting**. Το **underfitting** συμβαίνει όταν λόγω του μικρού αριθμού νευρώνων δεν είναι δυνατόν να εντοπιστούν επαρκώς τα πρότυπα που υπάρχουν μέσα σε ένα περίπλοκο **dataset**. Αντίθετα η χρήση υπερβολικά πολλών νευρώνων μπορεί να οδηγήσει σε **overfitting**. Το **overfitting** συμβαίνει όταν το νευρωνικό δίκτυο έχει τόσο μεγάλη χωρητικότητα για την επεξεργασία πληροφοριών που ο περιορισμένος αριθμός πληροφορίας που περιέχεται στο σετ εκπαίδευσης δεν είναι επαρκής για να εκπαιδευτούν όλοι οι νευρώνες στα ενδιάμεσα στρώματα. Ένα ακόμα πρόβλημα που μπορεί να εμφανιστεί είναι η υπερβολική αύξηση του χρόνου εκπαίδευσης του δικτύου. Ο χρόνος εκπαίδευσης μπορεί να αυξηθεί μέχρι το σημείο που είναι αδύνατο να εκπαιδευτεί επαρκώς το νευρωνικό μας δίκτυο. Υπάρχουν κάποιοι κανόνες για την επιλογή του αριθμού των νευρώνων στα ενδιάμεσα **layer** όπως οι ακόλουθοι:

- Ο αριθμός των κρυμμένων νευρώνων πρέπει να είναι μεταξύ του αριθμού στο **input layer** και του αριθμού στο **output layer**.
- Ο αριθμός των κρυμμένων νευρώνων πρέπει να είναι τα 2/3 του αριθμού στο **input layer** συν τον αριθμό στο **output layer**.
- Ο αριθμός των κρυμμένων νευρώνων πρέπει να είναι λιγότερος από το διπλάσιο του αριθμού στο **input layer**.

4.6.3 Πειράματα με το dataset της τράπεζας

Παρακάτω βλέπουμε τον πίνακα με τις τιμές που παίρνουν τα **training accuracy**, **testing accuracy**, **loss** και **testing loss** ανάλογα με τον αριθμό των νευρώνων στο **LSTM**. Διαπιστώνουμε εύκολα πως όσο αυξάνετε ο αριθμός των νευρώνων στο μοντέλο μας πετυχαίνουμε καλύτερες τιμές.

Το βέλτιστο πλήθος νευρώνων για το νευρωνικό μας δίκτυο είναι, όπως βλέπουμε **30**. Παρατηρούμε πως το **training accuracy** είναι ελάχιστα μικρότερο από το **testing accuracy**. Αυτό συμβαίνει λόγω του ότι κατά την περίοδο του **training** χρησιμοποιούμε **dropout** άρα δεν χρησιμοποιούμε το μοντέλο μας στο μέγιστο των δυνατοτήτων του, ενώ κατά την περίοδο του **testing** χρησιμοποιούμε ολόκληρο το νευρωνικό μας δίκτυο με όλους τους νευρώνες (χωρίς **dropout**).

Το **training loss** και το **testing loss** για 30 νευρώνες παρατηρούμε πως βρίσκονται πολύ κοντά. Αυτό μας διασφαλίζει πως δεν κάνουμε **overfit** το μοντέλο μας και πως είναι ικανό να κάνει σωστές προβλέψεις.

Neurons	Accuracy	Val_Accuracy	Loss	Val_Loss	Λάθος προβλέψεις
10	0.711	0.720	0.868	0.805	Όλες οι προβλέψεις είναι σωστές
15	0.717	0.726	0.810	0.770	Όλες οι προβλέψεις είναι σωστές
20	0.719	0.720	0.793	0.762	Όλες οι προβλέψεις είναι σωστές
30	0.720	0.724	0.770	0.755	Όλες οι προβλέψεις είναι σωστές

4.6.4 Πειράματα με το dataset του εργοστασίου

Παρακάτω βλέπουμε τον πίνακα με τα πειράματα με το πλήθος των νευρώνων για το **dataset** του εργοστασίου. Με το συγκεκριμένο **dataset** όπως έχουμε ξανά αναφέρει πετυχαίνουμε μικρότερες τιμές από αυτές που πετυχαίνουμε με το **dataset** της τράπεζας λόγω του μικρού του μεγέθους.

Το βέλτιστο πλήθος νευρώνων όπως παρατηρούμε είναι **30**. Και σε αυτή την περίπτωση καταλαβαίνουμε πως δεν κάνουμε **overfit** το μοντέλο μας αφού **training loss** και **testing loss** είναι περίπου ίσα.

Neurons	Accuracy	Val_Accuracy	Loss	Val_Loss	Λάθος προβλέψεις
10	0.250	0.235	2.852	2.861	3
15	0.268	0.276	2.735	2.756	2
20	0.317	0.320	2.575	2.584	2
30	0.345	0.350	2.465	2.459	3

4.7 Πειράματα με Activation Functions

Παρακάτω κάνουμε πειράματα με διαφορετικά **Activation Functions** για κάθε **dataset**. Παρατηρούμε πως στις πρώτες δύο (**Softmax**, **Sigmoid**) δεν παίρνουμε μεγάλες διαφορές στα αποτελέσματα. Στην συνέχεια όταν χρησιμοποιούμε την **Tanh** βλέπουμε πως το μοντέλο μας δεν αποδίδει καλά παίρνοντας πολύ χαμηλές τιμές για τα **training accuracy**, **testing accuracy** και πολύ υψηλές για τα **training loss** και **testing loss**. Τέλος διαπιστώνουμε πως με την χρήση της συνάρτησης **ReLU** το μοντέλο μας δεν ήταν καν σε θέση να κάνει προβλέψεις.

4.7.6 Πειράματα με το dataset της τράπεζας

Όπως παρατηρούμε από τον παρακάτω πίνακα με τις τιμές που παίρνουμε για το **training-testing accuracy**, το **training-testing loss** και το πλήθος των λάθος προβλέψεων για το **dataset** της τράπεζας, μπορούμε να επιλέξουμε για **activation function** είτε την **softmax** είτε την **sigmoid**. Θα επιλέξουμε την **softmax**, διότι το **training loss** είναι πολύ κοντά στο **testing loss** και έτσι δεν υπάρχει **overfitting**.

Με την χρήση της συνάρτησης **tanh** παρατηρούμε πως αυξάνεται η διαφορά μεταξύ **training loss** και **validation loss** και άρα αυξάνεται το **overfitting**. Επίσης οι περισσότερες προβλέψεις που κάνει το μοντέλο μας είναι λανθασμένες, κάτι που ήταν αναμενόμενο από το πού μικρό **testing accuracy** που πέτυχε το νευρωνικό μας.

Με την συνάρτηση **ReLU** το μοντέλο δεν ήταν σε θέση να κάνει καθόλου προβλέψεις. Οι τιμές για τα **training-testing loss** είναι κενές. Η κακιά επίδοση του μοντέλου μας ευθύνεται στο ότι τα **Relu units** είναι ‘’ευαίσθητα’’ κατά το **training** και μπορεί να ‘’πεθάνουν – καταστραφούν’’. Για παράδειγμα έχοντας μεγάλο **gradient** το οποίο περνάει μέσα από ένα **Relu neuron** μπορεί να προκαλέσει ένα **update** των **weights** το οποίο θα οδηγήσει τον νευρώνα να μην ενεργοποιηθεί για κανένα **datapoint** ξανά. Αν κάτι τέτοιο συμβεί τότε το **gradient** που θα περνάει μέσα από το **unit** θα είναι πάντα 0 από αυτό το σημείο και έπειτα. Έτσι το **Relu unit** ‘’πεθαίνει’’ κατά την εκπαίδευση. Για παράδειγμα θέτοντας υψηλό **learning rate** μπορεί να διαπιστώσουμε πως μέχρι και το **40%** του δικτύου μας είναι ‘’νεκρό’’. Με το κατάλληλο **learning rate** αυτό είναι λιγότερο πιθανό να συμβεί.

Activation Function	**Accuracy**	**Val_Accuracy**	**Loss**	**Val_Loss**	**Λάθος Προβλέψεις**
Softmax	0.7238	0.7230	0.765	0.756	1
Sigmoid	0.722	0.720	0.794	0.760	Όλες οι προβλέψεις είναι σωστές
Tanh	0.210	0.198	9.598	8.751	6
ReLU	0.013	0.0	nan	nan	Το μοντέλο δεν έκανε καθόλου προβλέψεις

4.7.7 Πειράματα με το dataset του εργοστασίου

Για το **dataset** του εργοστασίου παρατηρούμε από τον παρακάτω πίνακα, για ακόμη μια φορά τις χαμηλές τιμές που πετυχαίνουμε όσον αφορά το **training-testing accuracy**, λόγω του μεγέθους του **dataset**.

Διαπιστώνουμε πως τις μικρότερες τιμές τις πετυχαίνουμε με την χρήση των συναρτήσεων **sigmoid** και **tanh**, άρα κρίνουμε πως δεν είναι κατάλληλες για να κάνουμε προβλέψεις με το μοντέλο μας για το συγκεκριμένο **dataset**. Επίσης με την **tanh** πετυχαίνουμε μεγάλο **training-testing loss** κάτι το οποίο φαίνεται από τις προβλέψεις μας οι οποίες πολλές να είναι λανθασμένες.

Τον μεγαλύτερο βαθμό **testing accuracy** τον πετυχαίνει η συνάρτηση **relu**, παρόλαυτα θα επιλέξουμε την **softmax** διότι παίρνουμε μικρότερο **loss** και παρατηρούμε πως υπάρχει **overfitting** σε μικρότερο βαθμό από ότι υπάρχει στην **relu**.

Activation Function	Accuracy	Val_Accuracy	Loss	Val_Loss	Λάθος Προβλέψεις
Softmax	0.330	0.339	2.480	2.464	3
Sigmoid	0.169	0.126	2.894	2.911	3
Tanh	0.132	0.131	9.001	9.034	4
ReLU	0.356	0.365	4.520	4.870	2

4.8 Πειράματα με Dropout

Σε αυτή την ενότητα βλέπουμε τους πίνακες με τις τιμές που πήραμε για το **mean training-testing accuracy**, **mean training-testing loss** καθώς και το πλήθος των λάθος προβλέψεων που παίρνουμε ανάλογα με το πόσο **dropout** χρησιμοποιούμε.

Το **dropout** όπως αναφέραμε και σε προηγούμενο κεφάλαιο είναι η τυχαία απενεργοποίηση κόμβων, ώστε να αποφευχθεί το **overfitting**, ειδικότερα όταν έχουμε και λίγα δείγματα για να εκπαιδεύσουμε το νευρωνικό μας όπως στην περίπτωση του εργοστασίου.

Με τις ακόλουθες τιμές που πήραμε από τα πειράματα που κάναμε με τις τιμές του **dropout** αποσκοπούμε στο να βρούμε την βέλτιστη τιμή ώστε να μειώσουμε το **overfitting**, να πάρουμε την μέγιστη δυνατή ακρίβεια και να πετύχουμε σωστές προβλέψεις με το νευρωνικό μας.

4.8.9 Πειράματα για το dataset της τράπεζας

Παρακάτω βλέπουμε τον πίνακα με τις τιμές που πήραμε για το **dataset** της τράπεζας, για κάθε δοκιμή που κάναμε με το **dropout**.

Όπως παρατηρούμε πάντα το **test accuracy** είναι μεγαλύτερο από το **training accuracy**. Αυτό συμβαίνει διότι κατά το **testing** χρησιμοποιούμε ολόκληρο το

νευρωνικό μας δίκτυο για να κάνουμε προβλέψεις. Αντίθετα κατά την εκπαίδευση του δικτύου μας λόγω ακριβώς του **dropout** κάποιοι νευρώνες απενεργοποιούνται, έτσι όπως είναι λογικό το δίκτυο μας δεν έχει την ίδια προβλεπτική ικανότητα.

Από τις παρακάτω τιμές με τις οποίες πραγματοποιήσαμε πειράματα, θα επιλέξουμε για την τιμή του **dropout** αυτή του **0.2** διότι πετυχαίνουμε μεγάλο **test accuracy** και η διαφορά μεταξύ **train** – **test loss** είναι η μικρότερη κάτι που μας υποδεικνύει πως δεν έχουμε μεγάλο ποσοστό **overfitting**.

Dropout	**Accuracy**	**Val_Accuracy**	**Loss**	**Val_Loss**	**Λάθος προβλέψεις**
0.2	0.721	0.736	0.783	0.800	Όλες οι προβλέψεις είναι σωστές
0.4	0.714	0.734	0.831	0.810	Όλες οι προβλέψεις είναι σωστές
0.6	0.698	0.737	0.927	0.831	Όλες οι προβλέψεις είναι σωστές

4.8.10 Πειράματα με το dataset του εργοστασίου

Παρακάτω βλέπουμε τον πίνακα με τα αποτελέσματα που πήραμε, από τα πειράματα που κάναμε με το **dropout** για το **dataset** του εργοστασίου.

Όπως παρατηρούμε και για αυτό το **dataset** ή βέλτιστη τιμή για να χρησιμοποιήσουμε για το **dropout** είναι **0.4**, διότι πετυχαίνουμε το μεγαλύτερο **accuracy**.

Dropout	**Accuracy**	**Val_Accuracy**	**Loss**	**Val_Loss**	**Λάθος προβλέψεις**
0.2	0.330	0.346	2.465	2.437	2
0.4	0.315	0.349	2.536	2.448	3
0.6	0.276	0.331	2.653	2.550	3

4.9 Πειράματα με Optimizers

Όπως αναφέραμε και στο προηγούμενο κεφάλαιο, οι **optimizers** διαμορφώνουν το μοντέλο μας, ενημερώνοντας κάθε φορά τα βάρη, ώστε να πετυχαίνουμε τον μεγαλύτερο βαθμό ακρίβειας.

Παρακάτω βλέπουμε τα αποτελέσματα που πήραμε για διάφορους **optimizers** για το **dataset** της τράπεζας και για το **dataset** του εργοστασίου. Όπως παρατηρούμε υπάρχουν μεταβολές στην ακρίβεια αλλά και στο **loss** που πετυχαίνει το μοντέλο μας ανάλογα με τον ποιον **optimizer** χρησιμοποιούμε. Με τα παρακάτω πειράματα αποσκοπούμε στο να βρούμε τον **optimizer** που θα μεγιστοποιεί το **test accuracy**, θα ελαχιστοποιεί το **test loss** και το νευρωνικό μας δίκτυο δεν θα παρουσιάζει **overfitting**.

4.9.8 Πειράματα με το dataset της τράπεζας

Όπως βλέπουμε από τον πίνακα που ακολουθεί η καλύτερη επιλογή σχετικά με τον **optimizer** για το **dataset** της τράπεζας είναι αυτή του **adam**.

Παρατηρούμε από το **training-testing loss** πως δεν υπάρχουν σημάδια **overfitting**, αφού δεν έχουν μεγάλη διαφορά μεταξύ τους. Αντίθετα με τους άλλους **optimizers** βλέπουμε πως αυξάνεται η διαφορά μεταξύ των δύο **loss**. Επίσης με την χρήση του **adagrad** παρατηρούμε πως και το **training** αλλά και το **testing loss** αυξάνονται, με αποτέλεσμα να έχουμε αρκετές λανθασμένες προβλέψεις.

Επιπλέον με την χρήση του **adam** πετυχαίνουμε το μεγαλύτερο **accuracy** από τους άλλους δύο **optimizers** κάτι το οποίο μας λέει πως είναι καλύτερος **optimizer** για την διαμόρφωση των βαρών του μοντέλου μας.

Optimizer	**Accuracy**	**Val_Accuracy**	**Loss**	**Val_Loss**	**Λάθος προβλέψεις**
Adam	0.719	0.738	0.784	0.800	Όλες οι προβλέψεις είναι σωστές
Adamax	0.682	0.713	1.006	0.926	Όλες οι προβλέψεις είναι σωστές
Adagrad	0.436	0.494	2.339	2.221	5

4.9.9 Πειράματα για το dataset του εργοστασίου

Από τον παρακάτω πίνακα διαπιστώνουμε πως και για αυτή την περίπτωση, ο καλύτερος **optimizer** για το συγκεκριμένο **dataset** είναι ο **adam**.

Όπως βλέπουμε πετυχαίνουμε μεγαλύτερο **accuracy** από τις άλλες δύο περιπτώσεις. Στην περίπτωση του **adagrad** παρατηρούμε πως έχουμε πολύ μικρές τιμές για το **accuracy**. Αυτό οφείλεται πως στο πέρασμα των **epochs** το **learning rate** (παράμετρος που ελέγχει το πόσο θα αλλάξει το μοντέλο, δηλαδή το κατά πόσο θα αλλάξουν τα βάρη) θα μειωθεί σε τέτοιο βαθμό έτσι ώστε δεν θα γίνονται πια αλλαγές στα βάρη [28],[29].

Ακόμα με τον **adam** πετυχαίνουμε το χαμηλότερο **training error** και παίρνουμε τις περισσότερες σωστές προβλέψεις.

Optimizer	**Accuracy**	**Val_Accuracy**	**Loss**	**Val_Loss**	**Λάθος προβλέψεις**
Adam	0.336	0.342	2.463	2.450	2
Adamax	0.218	0.203	3.040	3.049	4
Adagrad	0.068	0.061	4.002	4.002	5

Κεφάλαιο 5

5.1 Γενικά συμπεράσματα

Το **LSTM** όπως τελικά είδαμε από τα πειράματα μας, αποδείχθηκε μια ιδιαίτερα καλή και ακριβής προσέγγιση από τον τομέα του **Deep learning** με την οποία μπορούμε να κάνουμε προβλέψεις με μεγάλη ακρίβεια (**accuracy** που φτάνει και πάνω από 70 %) σε **event logs** με πολλά (χιλιάδες) αλλά και με λίγα (μερικές εκατοντάδες) δείγματα. Με το σωστό **configuration** του **LSTM** μοντέλου, είμαστε σε θέση να προβλέψουμε με υψηλή ακρίβεια τα επόμενα **activities** σε ένα **Business process**. Όπως αποδείχθηκε, το **LSTM** μοντέλο δεν χρειάζεται να είναι ιδιαίτερα πολύπλοκο, δηλαδή δεν χρειάζεται να έχει πολλούς **νευρώνες** και πολλά **στρώματα**. Συγκεκριμένα ένα νευρωνικό δίκτυο με ένα **layer** και 30 νευρώνες, είναι ικανό να κάνει προβλέψεις με μεγάλη ακρίβεια, μικρό σφάλμα και χωρίς να γίνεται **overfitting**.

Σε σύγκριση με τους κλασσικούς αλγόριθμους **Process mining** (**Alpha miner, Heuristic, Inductive**) οι οποίοι είναι ντετερμινιστικοί, το **LSTM** μπορεί να διαχειριστεί καλύτερα τον θόρυβο στα δεδομένα εισόδου , την αβεβαιότητα, τα καινούργια δεδομένα και να υπολογίσει με μεγάλη ακρίβεια τα επόμενα βήματα μιας διαδικασίας σε πραγματικό χρόνο κάτι που το κάνει ιδιαίτερα εύχρηστο και συμβάλλει στην λήψη καλύτερων αποφάσεων όσον αφορά την διεξαγωγή αλλά και την αναδιαμόρφωση μιας διαδικασίας. Επιπροσθέτως είδαμε πως όσο μικρότερο είναι το **simplicity** ενός **process model** τόσο χαμηλότερη ακρίβεια θα είναι το **LSTM model** μας να πετύχει, κάτι στο οποίο μας οδηγεί στο συμπέρασμα πως η διαδικασίες πρέπει να είναι καλά ορισμένες.

Μέσω των πειραμάτων παρατηρήθηκε πως με το να κάνουμε το **LSTM** μοντέλο μας πιο πολύπλοκο δεν θα οδηγηθούμε με σιγουριά σε περισσότερο ακριβής προβλέψεις. Αντίθετα η συμπεριφορά του μοντέλου μας μπορεί να γίνει και πιο ασταθής.

Για να οδηγούμαστε σε σωστές προβλέψεις για κάποιο **dataset** και για να μπορεί το εκάστοτε μοντέλο που έχουμε δημιουργήσει κάθε φορά να γενικεύει σωστά, ώστε να είναι ικανό να κάνει προβλέψεις με καινούργια δεδομένα, αρκεί να χρησιμοποιήσουμε λίγα **epochs**, ώστε να αποφύγουμε το μοντέλο μας να απομνημονεύσει απλά τα δεδομένα εκπαίδευσης.

Επίσης μια παράμετρος που είναι σημαντική είναι το **batch size** αφού είναι ο αριθμός των δειγμάτων που εκπαιδεύεται το δίκτυο μας κάθε φορά, έτσι είναι σημαντικό να επιλέξουμε το κατάλληλο μέγεθος για να αποφύγουμε το **overfitting**. Για **datasets** μερικών εκατοντάδων δειγμάτων αλλά και για **datasets** χιλιάδων δειγμάτων ένα **batch size** μεταξύ **32** και **64** μας δίνει ικανοποιητικά αποτελέσματα.

Για τα **activation functions** αποδείχθηκε πως και η συνάρτηση **softmax** αλλά και η συνάρτηση **sigmoid**, η οποία είναι η πιο ευρέως χρησιμοποιούμενη δίνουν καλά αποτελέσματα και μπορούν να χρησιμοποιηθούν και οι δύο ανάλογα με το πρόβλημα που αντιμετωπίζετε κάθε φορά.

Επιπλέον από τα πειράματα που γίνανε με το **dropout**, είδαμε πως θέτοντας το, ίσο με **0.4** για μικρά **datasets** και ίσο με **0.2** για μεγάλα **datasets** αποφεύγουμε το

overfitting και παίρνουμε τις περισσότερες σωστές προβλέψεις, ενώ εξασφαλίζουμε την σωστή εκπαίδευση του νευρωνικού μας.

Από τα πειράματα που γίνανε με τους **optimizers** είδαμε πως ο βέλτιστος **optimizer** είναι ο **adam**, αφού πετυχαίνει με διαφορά την μεγαλύτερη ακρίβεια και κάνει τις περισσότερες σωστές προβλέψεις.

5.2 Μελλοντική εργασία

Η συγκεκριμένη διπλωματική εργασία μπορεί να μας δώσει το έναυσμα ώστε να επεκτείνουμε τις δυνατότητες του **LSTM** μοντέλου που έχουμε κατασκευάσει και να εργαστούμε ως προς τις περισσότερο μακροπρόθεσμες προβλέψεις, για παράδειγμα να προβλέπουμε όλα τα **events** ενός **trace**. Επίσης μια ακόμα ενδιαφέρουσα πτυχή είναι αυτή του χρόνου στον οποίο εκτελείται κάθε ενέργεια κάτι το οποίο θα θέλαμε να προβλέπουμε ώστε να κάνουμε τις προβλέψεις μας πιο στοχευμένες.

Αναφορές

[1] ΕΠΑΝΑΛΑΜΒΑΝΟΜΕΝΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΚΑΙ ΔΗΜΙΟΥΡΓΙΚΗ ΓΡΑΦΗ: ΜΙΑ ΕΚΠΑΙΔΕΥΤΙΚΗ ΣΥΖΕΥΞΗ
https://www.researchgate.net/publication/335911293_EPANALAMBANOMENA_NEURONIKA_DIKTYA_KAI_DEMIOURGIKE_GRAPHE_MIA_EKPAIDEUTIKE_SYZEUXE

[2] https://en.wikipedia.org/wiki/Long_short-term_memory

[3] Ν. Παναγιώτου, Ν. Ευαγγελόπουλος, Π. Κατημερτζόγλου και Σ. Γκαγιαλής, Διαχείριση Επιχειρησιακών Διαδικασιών, Αθήνα: Κλειδάριθμος, 2013.

[4] Θ. Μητάκος, «Κεφάλαιο 3, Διαδικασίες,» σε Πληροφοριακά Συστήματα Διοίκησης - Μελέτη, ανάλυση και διαχείριση, Ελληνικά Ακαδημαϊκά Ηλεκτρονικά Συγγράμματα και Βοηθήματα / Κάλλιπος, 2015, pp. 63-82.

[5]http://okeanis.lib2.uniwa.gr/xmlui/bitstream/handle/123456789/3630/tex_10305.pdf?sequence=1&isAllowed=y

[6] J. Jeston και J. Nelis, Business Process Management - Practical Guidelines to Successful Implementations, 2nd Edition, Amsterdam: Elsevier, 2008.

[7] T. Panagacos, The Ultimate Guide to Business Process Management, 2012.

[8] B. Vanhecke, «BPTrends,» 04 2013. [Ηλεκτρονικό]. Available: http://www.bptrends.com/how-managers-can-make-a-success-of-business-processmanagement/. [Πρόσβαση 31 01 2017].

[9] J. v. Brocke, T. Schmiedel, J. Recker, P. Trkman, W. Mertens και S. Viaene, «Ten principles of good business process management,» Business Process Management Journal, τόμ. vol. 20, αρ. Iss: 4, pp. 530 - 548, 2014.

[10] S. Tonchia και A. Tramontano, Process Management for the Extended Enterpris. Organizational and ICT network, Berlin: Springer, 2004.

[11] M. Weske, Business Process Management - Concept Language Architecture, Berlin: Springer, 2007.

[12] Wikipedia, «Wikipedia,» 2017. [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Business_process_management. [Πρόσβαση 25 01 2017].

[13]https://el.wikipedia.org/wiki/%CE%95%CE%BE%CF%8C%CF%81%CF%85%CE%BE%CE%B7_%CE%B4%CE%B9%CE%B5%CF%81%CE%B3%CE%B1%CF%83%CE%B9%CF%8E%CE%BD

[14] W. van der Aalst, "Extracting Event Data from Databases to Unleash Process Mining", Management for Professionals, pp. 105-128, 2015.

[15] W. van der Aalst, A. Adriansyah, A. de Medeiros, F. Arcieri, T. Baier, T. Blickle, J. Bose, P. van den Brand, R. Brandtjen, J. Buijs, A. Burattin, J. Carmona, M.

Castellanos, J. Claes, J. Cook, N. Costantini, F. Curbera, E. Damiani, M. de Leoni, P. Delias, B. van Dongen, M. Dumas, S. Dustdar, D. Fahland, D. Ferreira, W. Gaaloul, F. van Geffen, S. Goel, C. Günther, A. Guzzo, P. Harmon, A. ter Hofstede, J. Hoogland, J. Ingvaldsen, K. Kato, R. Kuhn, A. Kumar, M. La Rosa, F. Maggi, D. Malerba, R. Mans, A. Manuel, M. McCreesh, P. Mello, J. Mendling, M. Montali, H. Motahari-Nezhad, M. zur Muehlen, J. Munoz-Gama, L. Pontieri, J. Ribeiro, A. Rozinat, H. Seguel Pérez, R. Seguel Pérez, M. Sepúlveda, J. Sinur, P. Soffer, M. Song, A. Sperduti, G. Stilo, C. Stoel, K. Swenson, M. Talamo, W. Tan, C. Turner, J. Vanthienen, G. Varvaressos, E. Verbeek, M. Verdonk, R. Vigo, J. Wang, B. Weber, M. Weidlich, T. Weijters, L. Wen, M. Westergaard and M. Wynn, "Process Mining Manifesto", Business Process Management Workshops, pp. 169-194, 2012.

[16] W. van der Aalst, "Process Mining", ACM Transactions on Management Information Systems, vol. 3, no. 2, pp. 1-17, 2012.

[17] W. van der Aalst and S. Dustdar, "Process Mining Put into Context", IEEE Internet Computing, vol. 16, no. 1, pp. 82-86, 2012.

[18] E. Kindler, V. Rubin and W. Schäfer, "Process Mining and Petri Net Synthesis", Business Process Management Workshops, pp. 105-116, 2006.

[19] W. van der Aalst, "Business Process Management Demystified: A Tutorial on Models, Systems and Standards for Workflow Management", Lectures on Concurrency and Petri Nets, pp. 1-65, 2004.

[20] W. Aalst, Process mining. Berlin: Springer-Verlag, 2011.

[21] E. Kindler, V. Rubin and W. Schäfer, "Process Mining and Petri Net Synthesis", Business Process Management Workshops, pp. 105-116, 2006.

[22] W. van der Aalst, M. Netjes and H. Reijers, "Supporting the Full BPM Life-Cycle Using Process Mining and Intelligent Redesign", Advances in Database Research, pp. 100-132.

[23] W. van der Aalst, J. L. Zhao, H. J. Wang, "Editorial: Business Process Intelligence: Connecting Data and Processes", ACM Transactions on Management Information Systems, vol. 5, no. 4, pp. 1-7, 2015.

[24]https://nemertes.lis.upatras.gr/jspui/bitstream/10889/14408/1/_____%20%CE%B1%CF%84%CE%B9%CE%BA%CE%B7%CC%8115_3_2018.pdf%20-%20page=35&zoom=100,116,97

[25]https://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/10632/Lepenioti_Aikaterini.pdf?sequence=1&isAllowed=y

[26] ΒΑΤΙΚΙΩΤΗΣ ΦΩΤΙΟΣ Υπολογιστική ανάλυση και μοντελοποίηση της συμπεριφοράς καταγραφής συμβάντων ιστοσελίδων του διαδικτύου με τεχνικές εξόρυξης δεδομένων (Data Mining). https://dspace.lib.ntua.gr/xmlui/bitstream/handle/123456789/40750/diplomatiki_vatikiotis.pdf?locale-attribute=el

[27]A.J.M.M. Weijters, W.M.P. van der Aalst, and A.K. Alves de Medeiros Process Mining with the HeuristicsMiner Algorithm

[28] https://towardsdatascience.com/learning-parameters-part-5-65a2f3583f7d

[29] https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/

[30] https://forums.fast.ai/t/determining-when-you-are-overfitting-underfitting-or-just-right/7732

[31] https://stackoverflow.com/questions/55770578/understanding-epoch-batch-size-accuracy-and-performance-gain-in-lstm-forecasti

[32] https://datascience.stackexchange.com/questions/27561/can-the-number-of-epochs-influence-overfitting

[33] https://pm4py.fit.fraunhofer.de/documentation

[34] https://stackoverflow.com/questions/65725287/what-is-training-accuracy-and-training-loss-and-why-we-need-to-compute-them/65725496#65725496

[35] https://stackoverflow.com/questions/51335133/keras-how-come-accuracy-is-higher-than-val-acc

[36] https://machinelearningmastery.com/diagnose-overfitting-underfitting-lstm-models/

[37] https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/

[38] https://towardsdatascience.com/simplified-math-behind-dropout-in-deep-learning-6d50f3f47275