



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Κατανεμημένη Μάθηση Ορισμών Γεγονότων σε
πραγματικό χρόνο με τη χρήση Δραστών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΑΛΕΞΑΝΔΡΟΥ Χ. ΦΕΡΛΕ

Επιβλέπων: Γεώργιος Στάμου

Αναπληρωτής Καθηγητής Ε.Μ.Π.

ΕΡΓΑΣΤΗΡΙΟ ΕΠΕΞΕΡΓΑΣΙΑΣ ΕΙΚΟΝΑΣ, ΒΙΝΤΕΟ ΚΑΙ ΠΟΛΥΜΕΣΩΝ

Αθήνα, Ιούλιος 2017

Αυτή η σελίδα έχει σκοπίμως αφεθεί κενή.



Εθνικό Μετσόβιο Πολυτεχνείο

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Εργαστήριο Επεξεργασίας Εικόνας, Βίντεο και Πολυμέσων

Κατανεμημένη Μάθηση Ορισμών Γεγονότων σε πραγματικό χρόνο με τη χρήση Δραστών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΑΛΕΞΑΝΔΡΟΥ Χ. ΦΕΡΛΕ

Επιβλέπων: Γεώργιος Στάμου

Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 20η Ιουλίου 2017.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....

Γεώργιος Στάμου

Αν. Καθηγητής Ε.Μ.Π

.....

Ανδρέας-Γεώργιος

Σταφυλοπάτης

Καθηγητής Ε.Μ.Π.

.....

Δημήτρης Φωτάκης

Επ. Καθηγητής Ε.Μ.Π

Αθήνα, Ιούλιος 2017

(Υπογραφή)

.....

ΑΛΕΞΑΝΔΡΟΣ Χ. ΦΕΡΛΕΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

© 2017 – All rights reserved



Εθνικό Μετσόβιο Πολυτεχνείο

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Εργαστήριο Επεξεργασίας Εικόνας, Βίντεο και Πολυμέσων

Copyright ©–All rights reserved Αλέξανδρος Χ. Φερλές, 2017.

Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Περίληψη

Η αναγνώριση σύνθετων γεγονότων (Complex Event Recognition) αποτελεί αντικείμενο έρευνας στο πεδίο της μηχανικής μάθησης (Machine Learning). Στόχος ενός συστήματος αναγνώρισης σύνθετων γεγονότων είναι να χρησιμοποιεί διατυπώσεις γνωστών γεγονότων χαμηλού επιπέδου (low-level events) συνδυαστικά με ορισμούς σύνθετων γεγονότων ώστε να αναγνωρίζει τα γεγονότα υψηλού επιπέδου (high-level events) τα οποία πραγματοποιούνται δεδομένες χρονικές στιγμές. Το σύστημα Μάθησης Ορισμών Γεγονότων σε πραγματικό χρόνο (Online Learning of Event Definitions; OLED) χρησιμοποιεί τεχνικές επαγωγικού λογικού προγραμματισμού ώστε να διατυπώσει τις προϋποθέσεις οι οποίες πρέπει να πληρούνται προκειμένου να λαμβάνει χώρα ένα σύνθετο γεγονός, με τη μηχανική μάθηση να πραγματοποιείται σε πραγματικό χρόνο, υπό την έννοια ότι κάθε διαθέσιμο δεδομένο χρησιμοποιείται ακριβώς μία φορά χωρίς να διατηρείται στη μνήμη για περαιτέρω επεξεργασία. Σκοπός της διπλωματικής εργασίας είναι η ανάπτυξη ενός συστήματος μάθησης ορισμών γεγονότων σε πραγματικό χρόνο για καταναμημένα συστήματα επεξεργασίας. Η σχεδίαση του νέου αυτού συστήματος βασίστηκε στο προϋπάρχον σύστημα του OLED. Συγκεκριμένα, έγινε χρήση του μαθηματικού μοντέλου των δραστών (Actors), το οποίο χρησιμοποιείται την υλοποίηση ασύγχρονων συστημάτων καταναμημένης επεξεργασίας, και στηρίζεται στην ανταλλαγή και αξιοποίηση μηνυμάτων μεταξύ των υπολογιστικών νημάτων. Στην εν λόγω διπλωματική εργασία, παρουσιάζουμε το σύστημα Καταναμημένης Μάθησης Ορισμών Γεγονότων σε πραγματικό χρόνο (Distributed Online Learning of Event Definitions; DOLED) το οποίο επιτυγχάνει καταναμημένη μάθηση κανόνων με χρήση ενός συνόλου δεδομένων το οποίο έχει διασπαστεί σε τμήματα τα οποία με τη σειρά τους αναθέτονται σε πλήθος υπολογιστικών νημάτων. Το σύστημα αυτό παρουσιάζεται σε δύο διακριτές και ανεξάρτητες μεταξύ τους υλοποιήσεις, το Μοντέλο Ολικού Κλειδώματος και το Μοντέλο Ροής Πληροφορίας.

Λέξεις Κλειδιά

Λογισμός Πράξης, Αναγνώριση Σύνθετων Γεγονότων, Κατανεμημένα Συστήματα Επεξεργασίας, Επαγωγικός Λογικός Προγραμματισμός, Δράστες

Abstract

Complex Event Recognition is a scientific research area in Machine Learning. The purpose of a complex event recognition system is to combine known low-level events with complex-event definitions in order to recognize high-level events that occur in certain time points. Online Learning of Event Definitions (OLED) is an Incremental Logic Programming (ILP) system for learning Event Calculus theories in an Online manner. The scope of this thesis is the development of a distributed online learning system. The design of this system is based on the OLED system. In order to achieve a distributed algorithm, we deploy the Actor-model, which is an incoherent model of distributed processing that mainly operates by exchanging messages between active computing threads. We present the Distributed Online Learning of Event Definitions system, which achieves distributed learning of rules by using disjoint subsets of a dataset which are allocated to a number of computing threads. The DOLED system is presented in two distinct models, the Global-Lock Model and the Streaming Model.

Keywords

Event Calculus, Complex-Event Recognition, Distributed systems, Inductive Logic Programming, Actors

Αυτή η σελίδα έχει σκοπίμως αφαιρεθεί κενή.

Ευχαριστίες

Η παρούσα διπλωματική εργασία αποτελεί προϊόν συνεργασίας μεταξύ του εργαστηρίου Ψηφιακής Επεξεργασίας Εικόνας, Βίντεο και Πολυμέσων, της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, του Ε.Μ.Π, και του εργαστηρίου Τεχνολογίας Γνώσεων και Λογισμικού, του Ινστιτούτου Πληροφορικής και Τηλεπικοινωνιών, του Ε.Κ.Ε.Φ.Ε Δημόκριτος.

Θα ήθελα να ευχαριστήσω τον καθηγητή Γεώργιο Στάμου που δέχτηκε να επιβλέψει την διπλωματική αυτή εργασία. Επίσης θα ήθελα να ευχαριστήσω τους καθηγητές Αλέξανδρο Αρτίκη και Γεώργιο Παλιούρα για τη συνεχή καθοδήγηση τους και την υπομονή τους κατά την εκπόνηση της εργασίας. Ευχαριστώ ιδιαίτερα τον Δρ. Νίκο Κατζούρη για την πολύτιμη συνεργασία του σε όλο αυτό το διάστημα. Η διάθεση του να συζητήσει κάθε ιδέα που είχα και οι συμβουλές του για το πως να την μετουσιώσω σε κάτι λειτουργικό με βοήθησαν να ολοκληρώσω την εργασία αυτή.

Είναι πολλοί οι φίλοι τους οποίους ευχαριστώ από καρδιάς για τα όμορφα χρόνια των σπουδών μου. Θα ήθελα να αναφερθώ ιδιαίτερα στον Μιχάλη, τον Νίκο, τον Βασίλη, τον Σπύρο, τον Πέτρο, τον Τζίμη και τον Οδυσσέα για τις ατέλειωτες ώρες που περάσαμε μαζί είτε διαβάζοντας για τις εξεταστικές είτε για να προλάβουμε να παραδώσουμε την κάθε ομαδική εργασία πριν εκπνεύσει η τελευταία προθεσμία. Χωρίς εσάς δεν θα έφτανα στο σημείο να ξεκινήσω να γράφω το κείμενο αυτό.

Περισσότερο από όλους, θα ήθελα να ευχαριστήσω την οικογένεια μου για την αγάπη, την αδιάκοπη υποστήριξη και τη συμπαράσταση που μου προσέφερε απλόχερα όλα αυτά τα χρόνια.

Στον Χρήστο, τη Γεωργία, τη Λητώ...

Αυτή η σελίδα έχει σκοπίμως αφαιρεθεί κενή.

Περιεχόμενα

Περίληψη	1
Abstract	3
Ευχαριστίες	5
Περιεχόμενα	10
Κατάλογος Σχημάτων	13
Κατάλογος Πινάκων	15
1 Εισαγωγή	17
1.1 Λογισμός Γεγονότων	17
1.2 Σκοπός της διπλωματικής	18
1.3 Διάρθρωση εργασίας	18
2 Θεωρητικό υπόβαθρο	21
2.1 Αναγνώριση Σύνθετων Γεγονότων	21
2.1.1 Λογισμός Γεγονότων	21
2.1.2 Επαγωγικός Λογικός Προγραμματισμός	22

2.1.3	Συστήματα Αναγνώρισης Σύνθετων Γεγονότων	24
2.2	Μάθηση Ορισμών Γεγονότων σε πραγματικό χρόνο	25
2.2.1	Το όριο του Hoeffding	25
2.2.2	Το σύστημα OLED	26
2.2.3	Αλγοριθμική περιγραφή του OLED	28
2.2.4	Αξιολόγηση θεωρίας στο σύστημα OLED	32
2.3	Δράστες	32
2.3.1	Το μοντέλο των Δραστών	32
2.3.2	Διάρθρωση του OLED με τη χρήση Δραστών	34
2.4	Σχετικές εργασίες	35
3	Πρώτες επεκτάσεις στο σύστημα OLED	37
3.1	Κριτήρια επικοινωνίας	37
3.1.1	Νέοι κανόνες	38
3.1.2	Clause υποψήφιο για επέκταση	38
3.1.3	Εξειδίκευση ενός clause	39
3.1.4	Αποκοπή (pruning) ενός κανόνα	39
3.2	Μοντέλο Αναμονής Απαντήσεων	40
3.2.1	Η δομή Future	40
3.2.2	Το Μοντέλο Αναμονής Απαντήσεων	40
3.2.3	Ελαττώματα του μοντέλου	41
3.3	Μοντέλα Χρήσης Κοινών Δεδομένων	42

3.3.1	Διαφοροποίηση στον αλγόριθμο του OLED για χρήση σε καταναμημένα συστήματα	42
3.3.2	Περιορισμοί του μοντέλου των Δραστών	45
3.3.3	Ο Δράστης τύπου Modifier	45
3.3.4	Μοντέλο Ένα-προς-Ένα	45
3.3.5	Μοντέλο Ένα-προς-N	47
4	Καταναμημένη Μάθηση Ορισμών Γεγονότων σε πραγματικό χρόνο	49
4.1	Διαχωρισμός των παραδειγμάτων σε παρτίδες (batches) δεδομένων	49
4.2	Μοντέλο Ολικού Κλειδώματος	50
4.2.1	Προσέγγιση Ολικού Κλειδώματος	50
4.2.2	Ο Δράστης τύπου Συντονιστή	51
4.2.3	Αρχιτεκτονική διάρθρωση Δραστών στο Μοντέλο Ολικού Κλειδώματος . . .	51
4.2.4	Τύποι μηνυμάτων	52
4.2.4.1	New Rules	52
4.2.4.2	StatsRequested - Clauses Wanted	53
4.2.4.3	Clauses Wanted -Response	54
4.2.4.4	Drop Clause	55
4.2.4.5	Awaiting - Proceed	56
4.2.4.6	Συγκεντρωτικός πίνακας μηνυμάτων	57
4.2.5	Περιγραφή της λειτουργίας του Μοντέλου Ολικού Κλειδώματος	58
4.2.6	Αλγοριθμική παρουσίαση του Μοντέλου Ολικού Κλειδώματος	58
4.3	Μοντέλο Ροής Πληροφορίας	61

4.3.1	Χρονικοί περιορισμοί στο MOK	61
4.3.2	Ελαττωμένη αρχιτεκτονική Δραστών	61
4.3.3	Τύποι μηνυμάτων	62
4.3.3.1	New Rules - Clauses Wanted - Response	62
4.3.3.2	Refined Rule	63
4.3.3.3	Proceed	63
4.3.4	Προσέγγιση Ροής Πληροφορίας	63
4.3.5	Αλγοριθμική περιγραφή του Μοντέλου Ροής Πληροφορίας	64
5	Πειραματικά Αποτελέσματα και Παρατηρήσεις	67
5.1	Οργάνωση των πειραμάτων	67
5.2	Μοντέλο Ολικού Κλειδώματος	68
5.2.1	Meeting Event	69
5.2.2	Moving Event	71
5.3	Μοντέλο Ροής Πληροφορίας	73
5.3.1	Meeting Event	73
5.3.2	Moving Event	76
5.4	Συγκεντρωτικός πίνακας αποτελεσμάτων	78
6	Επίλογος	79
6.1	Σύνοψη και Συμπεράσματα	79
6.2	Μελλοντική Εργασία	80
	Βιβλιογραφία	82

Κατάλογος Σχημάτων

2.1	Λογικός Μηχανισμός ενός συστήματος Λογισμού Γεγονότων	22
2.2	Βήματα μάθησης κανόνων στο σύστημα OLED	30
2.3	Διάγραμμα επικοινωνίας Δραστών	34
2.4	Ιεραρχία Δραστών στο σύστημα OLED	35
3.1	Επικοινωνία μεταξύ Δραστών τύπου Learner και Scorer	41
3.2	Αρχιτεκτονική διάρθρωση με Modifiers στο μοντέλο Ένα-προς-Ένα	46
3.3	Αρχιτεκτονική διάρθρωση με έναν Modifier στο μοντέλο Ένα-προς-N	48
4.1	Διάρθρωση της ιεραρχίας των Δραστών στο Μοντέλο Ολικού Κλειδώματος	52
4.2	Δημιουργία και χειρισμός μηνύματος NewRules στο Μοντέλο Ολικού Κλειδώματος	53
4.3	Διαδικασία εξειδίκευσης ενός κανόνα στο Μοντέλο Ολικού Κλειδώματος	55
4.4	Επικοινωνία μεταξύ Learners στο Μοντέλο Ροής Πληροφορίας	62
4.5	Διάγραμμα ροής 2 Δραστών στο Μοντέλο Ροής Πληροφορίας	64
5.1	Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ολικού Κλειδώματος για το γεγονός meeting και απλή επεξεργασία του Caviar	69
5.2	Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ολικού Κλειδώματος για το γεγονός meeting και απλή επεξεργασία του Caviar	69

5.3	Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ολικού Κλειδώματος για το γεγονός meeting και εκτεταμένη επεξεργασία του Caviar	70
5.4	Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ολικού Κλειδώματος για το γεγονός meeting και εκτεταμένη επεξεργασία του Caviar	70
5.5	Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ολικού Κλειδώματος για το γεγονός moving και απλή επεξεργασία του Caviar	71
5.6	Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ολικού Κλειδώματος για το γεγονός moving και απλή επεξεργασία του Caviar	71
5.7	Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ολικού Κλειδώματος για το γεγονός moving και εκτεταμένη επεξεργασία του Caviar	72
5.8	Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ολικού Κλειδώματος για το γεγονός moving και εκτεταμένη επεξεργασία του Caviar	72
5.9	Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ροής Πληροφορίας για το γεγονός meeting και απλή επεξεργασία του Caviar	73
5.10	Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ροής Πληροφορίας για το γεγονός meeting και απλή επεξεργασία του Caviar	74
5.11	Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ροής Πληροφορίας για το γεγονός meeting και εκτεταμένη επεξεργασία του Caviar . . .	74
5.12	Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ροής Πληροφορίας για το γεγονός meeting και εκτεταμένη επεξεργασία του Caviar	75
5.13	Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ροής Πληροφορίας για το γεγονός moving και απλή επεξεργασία του Caviar	76
5.14	Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ροής Πληροφορίας για το γεγονός moving και απλή επεξεργασία του Caviar	76
5.15	Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ροής Πληροφορίας για το γεγονός moving και εκτεταμένη επεξεργασία του Caviar	77

5.16 Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ροής Πληροφορίας για το γεγονός moving και εκτεταμένη επεξεργασία του Caviar	77
--	----

Αυτή η σελίδα έχει σκοπίμως αφαιρεθεί κενή.

Κατάλογος Πινάκων

2.1	Απλοποιημένη μορφή του Λογισμού Γεγονότων	22
2.2	Θετικά και αρνητικά παραδείγματα ως προς ένα σύνθετο γεγονός	24
2.3	Ανανέωση δεικτών στο σύστημα OLED	29
2.4	Αλγόριθμος του OLED	31
3.1	Τροποποιημένος αλγόριθμος της μάθησης κανόνων με χρήση κοινών δεδομένων . .	44
4.1	Τύπος μηνυμάτων για το σύστημα Κατανεμημένης Ακολουθιακής Αναγνώρισης Σύνθετων Γεγονότων σε συνδυασμό με τον τρόπο χειρισμού τους από τους δράστες τύπου Learner και Coordinator	57
4.2	Συναρτήσεις receive των Learner και Coordinator	60
4.3	Συνάρτηση receive των Learner στο μοντέλο Ροής Πληροφορίας	66
5.1	Συγκεντρωτικός πίνακας αποτελεσμάτων	78

Αυτή η σελίδα έχει σκοπίμως αφαιρεθεί κενή.

Κεφάλαιο 1

Εισαγωγή

1.1 Λογισμός Γεγονότων

Ο Λογισμός Γεγονότων (Event Calculus,[26]) εισήχθη το 1986 από τους R. Kowalski και M.Sergot ως μία προσπάθεια επέκτασης της Τυπικής Λογικής (Formal Logic), με σκοπό να επεκτείνει τις διατυπώσεις λογικών γεγονότων ώστε να σημειώνεται ο χρόνος στον οποίο πραγματοποιούνται αυτά τα γεγονότα. Μέσω αυτής της επέκτασης δίνεται η δυνατότητα να διατυπωθούν πλήρως γεγονότα σε αυστηρά υπολογιστική γλώσσα, και να υποστούν επεξεργασία από υπολογιστικές μονάδες.

Η Αναγνώριση Σύνθετων Γεγονότων (Complex Event Recognition, συναντάται συχνά στη βιβλιογραφία ως Complex Event Processing) αποτελεί πεδίο της Μηχανικής Μάθησης (Machine Learning), και αφορά την αναγνώριση σύνθετων γεγονότων[28] τα οποία πραγματοποιούνται σε δεδομένες χρονικές στιγμές χρησιμοποιώντας τη γνώση για τα γεγονότα μικρής κλίμακας (low-level-events) τα οποία πραγματοποιούνται σε συνδυασμό με τους ορισμούς των ιδίων των σύνθετων γεγονότων τα οποία αναγνωρίζουν. Οι έν λόγω ορισμοί μπορούν να εξαχθούν από συστήματα Μάθησης Ορισμών Γεγονότων, τα οποία χρησιμοποιούν διατυπώσεις γεγονότων χαμηλής και υψηλής κλίμακας για να διατυπώσουν θεωρίες για την πραγματοποίηση των γεγονότων υψηλής κλίμακας. Μία διαδικασία Μάθησης Ορισμών Γεγονότων είναι χρονικά και υπολογιστικά δαπανηρή, καθώς αφορά επεξεργασία μεγάλου όγκου αρχικών δεδομένων, τα οποία σε πολλές προσεγγίσεις επαναχρησιμοποιούνται για την εξαγωγή της διατύπωσης ενός σύνθετου γεγονότος. Ακόμα και προσεγγίσεις οι οποίες πραγματοποιούνται σε πραγματικό χρόνο (δηλαδή προσεγγίσεις οι οποίες χρησιμοποιούν ακριβώς μία φορά κάθε διαθέσιμο δεδομένο) απαιτούν μεγάλο χρονικό διάστημα για την περίπτωση

τους, λόγω των ολοένα αυξανόμενων όγκων δεδομένων.

Ως αποτέλεσμα, έχουν κάνει την εμφάνιση τους συστήματα κατανεμημένης επεξεργασίας των δεδομένων, όπου υπολογιστές οι οποίοι είναι ανεξάρτητοι μεταξύ τους αναλαμβάνουν τον χειρισμό ενός μέρους των δεδομένων και συνδυάζουν τα αποτελέσματα τους με σκοπό να εξάγουν πληροφορία παρόμοιας ποιότητας σε σχέση με τα μονολιθικά συστήματα, σε μικρότερο χρονικό διάστημα. Καθώς μία διαδικασία μάθησης Ορισμών Γεγονότων οι οποίοι θα είναι σε θέση να περιγράψουν τις συνθήκες οι οποίες θα πρέπει να πληρούνται ώστε να πραγματοποιούνται συγκεκριμένα σύνθετα γεγονότα δεν είναι καθολικά ανεξάρτητη, υπό την έννοια πως κάθε υπολογιστικός κόμβος πρέπει να χρησιμοποιεί πληροφορίες οι οποίες δεν είναι πάντα δυνατό να εξαχθούν από τα διαθέσιμα σε εκείνον δεδομένα, υπάρχει η ανάγκη επικοινωνίας μεταξύ των κόμβων υπό προϋποθέσεις οι οποίες τίθενται από το σχεδιαστή του κατανεμημένου συστήματος. Γίνεται χρήση διαφόρων μοντέλων παράλληλης και κατανεμημένης επεξεργασίας, και τα κριτήρια επικοινωνίας μεταξύ των κόμβων ποικίλλουν.

1.2 Σκοπός της διπλωματικής

Σκοπός της διπλωματικής αυτής εργασίας είναι ο σχεδιασμός ενός συστήματος Κατανεμημένης Μάθησης Ορισμών Γεγονότων σε πραγματικό χρόνο, επεκτείνοντας το προϋπάρχον σύστημα Μάθησης Ορισμών Γεγονότων σε πραγματικό χρόνο (Online Learning of Event Definitions; OLED). Χρησιμοποιώντας ως βάση το μοντέλο των Δραστών, πραγματοποιήθηκε ο σχεδιασμός διαφορετικών αρχιτεκτονικών και ειδών Δραστών, όπως επίσης οι τρόποι με τους οποίους πραγματοποιούταν η επικοινωνία μέσω μηνυμάτων στις αρχιτεκτονικές αυτές. Το αποτέλεσμα της εργασίας συνοψίζεται κατά κύριο λόγο σε δύο διακριτά μοντέλα: το Μοντέλο Ολικού Κλειδώματος και το Μοντέλο Ροής Πληροφορίας.

1.3 Διάρθρωση εργασίας

Η εργασία οργανώνεται σε κεφάλαια με την ακόλουθη δομή:

- **Κεφάλαιο 2:** Σε αυτό το κεφάλαιο γίνεται αναφορά στην υπάρχουσα βιβλιογραφία σχετικά με το Λογισμό Πράξης, τις τεχνικές Μηχανικής Μάθησης που χρησιμοποιούνται στην Αναγνώριση Σύνθετων Γεγονότων όπως επίσης και στο μοντέλο των Δραστών. Επιπλέον,

περιέχει μια αναλυτική περιγραφή του συστήματος Σύγχρονης Αναγνώρισης Σύνθετων Γεγονότων το οποίο χρησιμοποιήθηκε ως πρότυπο για τη δημιουργία ενός αντίστοιχου κατανεμημένου συστήματος, καθώς και η μοντελοποίηση του με τη χρήση Δραστών. Εν κατακλείδι, γίνεται αναφορά στις σχετικές εργασίες που έχουν γίνει για τη δημιουργία συστημάτων κατανεμημένης μάθησης.

- **Κεφάλαιο 3:** Πραγματοποιείται μία σύντομη αναφορά στις πρώτες επεκτάσεις του συστήματος, οι οποίες στη συνέχεια εγκαταλείφθηκαν για λόγους οι οποίοι εξηγούνται στο κεφάλαιο αυτό.
- **Κεφάλαιο 4:** Γίνεται θεωρητική περιγραφή των δύο βασικών μοντέλων Κατανεμημένης Μάθησης Ορισμών Γεγονότων σε πραγματικό χρόνο. Εξηγούνται οι τρόποι επικοινωνίας μεταξύ των Δραστών οι οποίοι λειτουργούν στα συστήματα και παρουσιάζεται η αρχιτεκτονική διάρθρωση και περιγράφονται οι αλγόριθμοι των συστημάτων αυτών.
- **Κεφάλαιο 5:** Παρουσιάζονται οι πειραματικές μετρήσεις που έγιναν στις υλοποιήσεις των δύο μοντέλων τα οποία παρουσιάζονται στο κεφάλαιο 4, καθώς και τα αποτελέσματά τους. Γίνεται επίσης ποιοτικός σχολιασμός, όπως και παρατηρήσεις πάνω σε αυτά.
- **Κεφάλαιο 6:** Συνοψίζονται τα συμπεράσματα και η συνεισφορά της εργασίας μας και προτείνονται διορθώσεις και επεκτάσεις πάνω σε αυτές.

Αυτή η σελίδα έχει σκοπίμως αφεθεί κενή.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

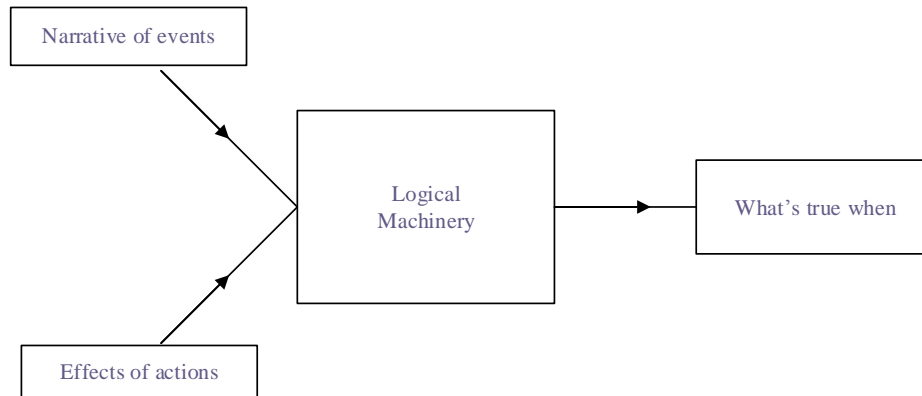
2.1 Αναγνώριση Σύνθετων Γεγονότων

2.1.1 Λογισμός Γεγονότων

Ο Λογισμός Γεγονότων[34] (Event Calculus) αποτελεί έναν τυπικό φορμαλισμό ο οποίος έχει χρησιμοποιηθεί σε αρκετές εφαρμογές Αναγνώρισης Γεγονότων. Χρησιμοποιείται για τη διατύπωση τους, κατά τέτοιο τρόπο ώστε να γίνεται αντιληπτή η επιρροή τους σε άτομα του προς μελέτη σύμπαντος. Μπορούμε να διακρίνουμε μια οντολογία Λογισμού Γεγονότων σε 3 μέρη:

1. **Χρονικές στιγμές**(time points): Ένας πραγματικός αριθμός ο οποίος διατυπώνει μία δεδομένη χρονική στιγμή κατά την οποία πραγματοποιείται ένα γεγονός.
2. **Μεταβλητές**(fluents): Άτομα τα οποία σε μία δεδομένη χρονική στιγμή έχουν σταθερή (λογική) τιμή.
3. **Γεγονότα**(events): Γεγονότα τα οποία με την εμφάνιση τους μπορούν να τροποποιήσουν την τιμή μίας ή περισσότερων μεταβλητών.

Ένα σύστημα Λογισμού Γεγονότων λειτουργεί ως ένας λογικός μηχανισμός[34] στον οποίο παρέχονται πληροφορίες για μία σειρά από χρονικές στιγμές. Κάθε πληροφορία μπορεί να συνίσταται από την αναφορά γεγονότων τα οποία πραγματοποιούνται (narrative of events) σε συνδυασμό με τη γνώση της επιρροής που μπορούν να έχουν συγκεκριμένες πράξεις σε άτομα του σύμπαντος (effects of actions), ώστε να διατυπώσει ποια γεγονότα είναι αληθή σε δεδομένες χρονικές στιγμές.



Σχήμα 2.1 – Λογικός Μηχανισμός ενός συστήματος Λογισμού Γεγονότων: Οι διατυπώσεις απλών γεγονότων που πραγματοποιούνται σε δεδομένες χρονικές στιγμές συνδυάζονται με επιδράσεις των πράξεων οι οποίες είναι γνωστές στο μηχανισμό για τη διατύπωση ευρύτερων γεγονότων που επίσης πραγματοποιούνται στο ίδιο χρονικό διάστημα.

Ο μηχανισμός ενός συστήματος Λογισμού Γεγονότων, συνδυάζει τις διατυπώσεις αυτές και συνθέτει δηλώσεις για να περιγράψει τα γεγονότα μεγάλης κλίμακας.

Τα αξιώματα του Λογισμού Γεγονότων ακολουθούν την αρχή της αδράνειας: όλες οι σταθερές διατηρούν την τιμή τους αναλλοίωτη με την πάροδο του χρόνου, εκτός εάν πραγματοποιηθεί κάποιο γεγονός το οποίο θα την μεταβάλει. Για τις ανάγκες της εργασίας χρησιμοποιείται μια απλοποιημένη μορφή του Λογισμού Γεγονότων, η οποία παρουσιάζεται στον ακόλουθο πίνακα:

Κατηγορία	Ποιοτική σημασία	Αξίωμα
$\text{happensAt}(E, T)$	Το γεγονός E πραγματοποιείται τη χρονική στιγμή T	
$\text{initiatedAt}(F, T)$	Τη χρονική στιγμή T εκκινείται ένα χρονικό διάστημα κατά το οποίο η σταθερά F είναι αληθής	$\text{holdsAt}(F, T + 1) \leftarrow \text{initiatedAt}(F, T)$
$\text{terminatedAt}(F, T)$	Τη χρονική στιγμή T ολοκληρώνεται ένα χρονικό διάστημα κατά το οποίο η σταθερά F ήταν αληθής	$\text{holdsAt}(F, T + 1) \leftarrow \text{holdsAt}(F, T),$
$\text{holdsAt}(E, T)$	Η σταθερά E είναι αληθής τη χρονική στιγμή T	$\neg \text{terminatedAt}(F, T)$

Πίνακας 2.1 – Απλοποιημένη μορφή του Λογισμού Γεγονότων.

2.1.2 Επαγωγικός Λογικός Προγραμματισμός

Ο Επαγωγικός Λογικός Προγραμματισμός[8] (Inductive Logic Programming, ILP) είναι μία τεχνική μηχανικής μάθησης για την εξαγωγή κανόνων με τη χρήση ενός συνόλου παραδειγ-

μάτων¹ E και με αναφορά σε ήδη υπάρχουσα γνώση, η οποία αποκαλείται γνώση βάσης (Background Knowledge, B). Επιπρόσθετα γίνεται χρήση του συνόλου M το οποίο αποτελεί υποσύνολο των συνολικών πιθανών υποθέσεων και αποκαλείται language bias.

Για τις ανάγκες του Επαγωγικού Λογικού Προγραμματισμού, το σύνολο E χωρίζεται στα υποσύνολα:

- E^+ : Ένα σύνολο από θετικά παραδείγματα, δηλαδή παραδείγματα τα οποία έχουν αληθή τιμή ως προς τη γενική γνώση που θέλει να εξάγει ο επαγωγικός λογικός προγραμματισμός.
- E^- : Ένα σύνολο από αρνητικά παραδείγματα, στα οποία κατά αντιστοιχία η τιμή της γενικής γνώσης είναι ψευδής².

Ένα σύστημα Επαγωγικού Λογικού Προγραμματισμού προσπαθεί επομένως να εξάγει ένα σύνολο H με κανόνες μη κενούς ως προς το περιεχόμενο (non-ground clauses), διατυπωμένους στη γλώσσα που ορίζει το σύνολο M και σε συνδυασμό με το σύνολο αναφοράς B να καλύπτει τα παραδείγματα του συνόλου E κατά τέτοιο τρόπο ώστε:

$$B \cup H \models E^+, B \cup H \not\models E^-$$

¹Με τον όρο παράδειγμα (example) αναφερόμαστε σε οποιαδήποτε πληροφορία αναφέρεται αληθώς για μία δεδομένη χρονική στιγμή

²Κατά κανόνα ισχύει η υπόθεση του κλειστού κόσμου (closed-world assumption) η οποία δηλώνει πως ένα παράδειγμα το οποίο δεν αναφέρεται ρητά ως θετικό, μπορεί να ληφθεί ως αρνητικό, και παρέχει τη δυνατότητα της άρνησης ως αποτυχίας[7] (negation as failure).

Narrative	Annotation
holdsAt (orientation(id4,0,5680))	holdsAt (moving(id4,id5)5720)
holdsAt (visible(id4),5680)	holdsAt (moving(id5,id4)5720)
holdsAt (orientation(id5,169,5680))	
holdsAt (visible(id5),5680)	
happensAt (walking(id4),5680)	
holdsAt (coords(id4,46,184,5680))	
happensAt (walking(id5),5680)	
holdsAt (orientation(id4,0,5720))	holdsAt (moving(id4,id5),5720)
holdsAt (visible(id4),5720)	holdsAt (moving(id5,id4),5720)
holdsAt (orientation(id5,169,5720))	
holdsAt (visible(id5),5720)	
happensAt (walking(id4),5720)	
holdsAt (coords(id4,46,184,5720))	
happensAt (walking(id5),5720)	
holdsAt (coords(id5,45,167,5720))	
holdsAt (orientation(id4,0,5760))	holdsAt (meeting(id4,id5),5760)
holdsAt (visible(id4),5760)	holdsAt (meeting(id5,id4),5760)
holdsAt (orientation(id5,169,5760))	
holdsAt (visible(id5),5760)	
happensAt (walking(id4),5760)	
holdsAt (coords(id4,46,184,5760))	
happensAt (walking(id5),5760)	
holdsAt (coords(id5,45,167,5760))	

Πίνακας 2.2 – Θετικά και αρνητικά παραδείγματα ως προς ένα σύνθετο γεγονός: Το γεγονός meeting παίρνει αληθή τιμή τη χρονική στιγμή 5760 και αρνητική τιμή στις 2 προηγούμενες χρονικές στιγμές (5680 και 5720), αφού δεν υπάρχει αναφορά σε αυτό.

Μία διαδικασία μάθησης κανόνων με χρήση Επαγωγικού Λογικού Προγραμματισμού συνήθως ακολουθεί την λογική του διαίρει-και-βασίλευε, με την έννοια ότι αρχικά κατασκευάζονται κανόνες οι οποίοι προσπαθούν να καλύψουν υποσύνολα των δεδομένων και η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου να μπορέσουν να καλύψουν όλα τα παραδείγματα του συνόλου E . Οι κανόνες αρχικά είναι κενοί ως προς το περιεχόμενο και εξειδικεύονται με την προσθήκη ατόμων (literals) σε αυτό. Η επιλογή των ατόμων τα οποία θα στελεχώσουν κάθε κανόνα γίνεται βάσει μίας ευριστικής συνάρτησης G : τα άτομα με το υψηλότερο G -score προστίθενται στο σώμα των κανόνων και η διαδικασία αυτή επαναλαμβάνεται αναδρομικά μέχρι το σύνολο E να καλυφθεί.

2.1.3 Συστήματα Αναγνώρισης Σύνθετων Γεγονότων

Τα συστήματα Αναγνώρισης Σύνθετων Γεγονότων[11] επεξεργάζονται δηλώσεις απλών γεγονότων (simple events) τα οποία πραγματοποιούνται σε δεδομένες χρονικές στιγμές σε συνδυασμό με τις προϋποθέσεις οι οποίες πρέπει να πληρούνται προκειμένου να λαμβάνου χώρα κάποια

σύνθετα γεγονότα ώστε να διατυπώσουν ώστε να αναγνωρίσουν ποια σύνθετα γεγονότα είναι αληθή σε ποιές χρονικές στιγμές. Η χρήση του Λογισμού Γεγονότων σε συνδυασμό με Επαγωγικό Λογικό Προγραμματισμό συναντάται σε συστήματα Αναγνώρισης Σύνθετων Γεγονότων[3, 4]. Οι διατυπώσεις των παραπάνω προϋποθέσεων προκύπτουν συχνά από συστήματα Μάθησης Ορισμών Γεγονότων, τα οποία χρησιμοποιούν γνωστά γεγονότα χαμηλής και υψηλής κλίμακας για να διατυπώσουν θεωρίες για την πλήρωση των γεγονότων υψηλής κλίμακας μέσω συνδυασμών των γεγονότων χαμηλής κλίμακας. Η εμφάνιση θορύβου είναι συχνή στις ροές δεδομένων που επεξεργάζονται τέτοιου είδους συστήματα. Για την αντιμετώπιση του φαινομένου αυτού, ακολουθείται μία τακτική χειρισμού των παραδειγμάτων σε πραγματικό χρόνο[13], με την έννοια ότι κάθε διαθέσιμο δεδομένο τίθεται υπό επεξεργασία ακριβώς μία φορά.

Στην επόμενη παράγραφο παρουσιάζεται ένα σύγχρονο σύστημα αναγνώρισης γεγονότων, το οποίο μαθαίνει κανόνες μέσω ερμηνειών[6] (κάθε παράδειγμα αντιμετωπίζεται ως ένα σύνολο από αληθή άτομα βάσης), το οποίο αποτελεί και τη βάση της διπλωματικής αυτής εργασίας.

2.2 Μάθηση Ορισμών Γεγονότων σε πραγματικό χρόνο

2.2.1 Το όριο του Hoeffding

Το όριο του Hoeffding[17] (Hoeffding bound) αποτελεί ένα στατιστικό εργαλείο με το οποίο μπορούμε να περιορίσουμε σε ένα αρκετά μικρό διάστημα πραγματικών αριθμών, τον πραγματικό μέσο \hat{X} μιας τυχαίας μεταβλητής X . Πιο συγκεκριμένα, για μια τυχαία μεταβλητή $X \in [0, 1]$ και τη μέση τιμή \bar{X} η οποία έχει προκύψει από N παρατηρήσεις, η πραγματική μέση τιμή της μεταβλητής X , \hat{X} βρίσκεται εντός του διαστήματος $[\bar{X} - \epsilon, \bar{X} + \epsilon]$ με πιθανότητα $1 - \delta$, όπου το ϵ αποκαλείται το όριο του Hoeffding και ποιοτικά αποτελεί το περιθώριο λάθους που φέρει η βεβαιότητα $1 - \delta$ με την οποία θέλουμε να προσεγγίσουμε την πραγματική τιμή. Το όριο του Hoeffding εξαρτάται και από το πλήθος N των παρατηρήσεων και η τιμή του δίδεται από τον ακόλουθο τύπο:

$$\epsilon = \sqrt{\frac{\ln(1/\delta)}{2N}} \quad (2.1)$$

2.2.2 Το σύστημα OLED

Το σύστημα της Μάθησης Ορισμών Γεγονότων[24, 22] σε πραγματικό χρόνο (Online Learning of Event Definitions; OLED) αποτελεί ένα ILP σύστημα για τη μάθηση κανόνων μέσω παραδειγμάτων τα οποία αντλούνται από ένα σετ δεδομένων (datasets). Τα παραδείγματα λογίζονται ως ανεξάρτητες παρατηρήσεις (independent observations) σύμφωνα και με την παρατήρηση των Blockeet et al.[6] για τη μάθηση μέσω ερμηνειών.

Σύμφωνα και με την παρατήρηση αυτή, ικανοποιείται η προϋπόθεση για τη χρήση του ορίου του Hoeffding, όπως αυτό ορίστηκε στη σχέση (2.1). Υιοθετώντας μία μέθοδο των Domingos και Hulten[10], στο σύστημα της Σύγχρονης Αναγνώρισης Σύνθετων Γεγονότων επιλέγονται οι δύο κατά σειρά καλύτερες εξειδικεύσεις ενός κανόνα, r_1 και r_2 , βάσει της τιμής των scores τους τα οποία εκτιμώνται μέσω μίας γνωστής ευριστικής συνάρτησης G , ώστε:

$$\Delta \bar{G} = \bar{G}(r_1) - \bar{G}(r_2) > 0 \quad (2.2)$$

Σύμφωνα επομένως και με τον ορισμό του ορίου του Hoeffding, το παραπάνω συνεπάγεται για την πραγματική μέση τιμή των G-scores πώς:

$$\Delta \hat{G} > \Delta \bar{G} - \epsilon \quad (2.3)$$

με πιθανότητα $1 - \delta$, όπου το δ αποτελεί ένα προκαθορισμένο περιθώριο λάθους που είναι διατεθειμένος να κάνει ο σχεδιαστής ενός συστήματος προκειμένου να γίνει χρήση λιγότερων παραδειγμάτων για την εξειδίκευση ενός κανόνα. Επομένως μπορούμε να υποστηρίξουμε ότι:

$$\Delta \bar{G} > \epsilon \Rightarrow \Delta \hat{G} > 0 \quad (2.4)$$

Εφόσον επομένως ισχύει η πρώτη εκ των δύο ανισώσεων, η r_1 είναι η καλύτερη δυνατή εξειδίκευση ενός κανόνα. Μέσω αυτής της προσέγγισης ένας αλγόριθμος μπορεί να χρησιμοποιήσει κάθε διαθέσιμο παράδειγμα ακριβώς μία φορά ώστε να αξιολογήσει τους κανόνες και τις εξειδικεύσεις τους βάσει της ευριστικής συνάρτησης G .

Σκοπός του συστήματος OLED είναι να μάθει κανόνες στη μορφή `initiatedAt` και `terminatedAt` από ένα σετ παραδειγμάτων ώστε να μπορέσει να διατυπώσει θεωρίες που να μπορούν

να περιγράψουν κατάλληλα τα παραδείγματα αυτά. Απαιτείται επομένως κατάλληλος ορισμός της ευριστικής συνάρτησης G ώστε να μπορεί να γίνει χρήση της κατά την εξειδίκευση κανόνων.

Το σύστημα OLED χρησιμοποιεί ως βάση γνώσης αξιώματα του απλοποιημένου Λογισμού γεγονότων όπως αυτά ορίστηκαν στον πίνακα 2.3. Επιπλέον, θεωρούμε έναν κανόνα r και μία ερμηνεία I , η οποία με τη σειρά της χωρίζεται στα μέρη $narrative(I)$ και $annotation(I)$, όπως επίσης το σύνολο (πιθανών) απαντήσεων M των $BUnarrative(I)Ur$. Θέλουμε επομένως να αξιολογήσουμε έναν κανόνα r πάνω στα διαθέσιμα δεδομένα.

Κάθε άτομο α το οποίο προκύπτει με χρήση του κανόνα r μπορεί να διακριθεί στις ακόλουθες 3 κατηγορίες:

- **Αληθώς Αληθές** άτομο (True Positive atom): Ένα άτομο α θεωρείται αληθώς αληθές όταν ανήκει τόσο στο $annotation$ της ερμηνείας I όσο και στο σετ απαντήσεων M ($\alpha \in annotation(I) \cup M$).
- **Εσφαλμένα Αληθές** άτομο (False Positive atom): Ένα άτομο α θεωρείται εσφαλμένα αληθές όταν ανήκει στο σετ απαντήσεων M χωρίς όμως να ανήκει στο $annotation$ της ερμηνείας I ($\alpha \in M, \alpha \notin annotation(I)$).
- **Εσφαλμένα Ψευδές** άτομο (False Negative atom): Ένα άτομο α θεωρείται εσφαλμένα ψευδές όταν ενώ ανήκει στο $annotation$ της ερμηνείας I , δεν έχει προσαρτηθεί στο σετ απαντήσεων M ($\alpha \in annotation(I), \alpha \notin M$).

Βασικό στόχο του συστήματος αποτελεί η εύρεση μίας θεωρίας (δηλαδή ενός συνόλου κανόνων τύπου $initiatedAt$ και $terminatedAt$) η οποία να οδηγεί σε ταυτόχρονα μεγάλο αριθμό TP ατόμων και μικρό αριθμό FP, FN ατόμων. Οι λόγοι στους οποίους οφείλεται η εμφάνιση FP και FN ατόμων, συνοψίζονται ως εξής:

- Εσφαλμένα Ψευδές άτομο:
 1. Η πρώτη (πιθανή) αιτία για το γεγονός πως ένα άτομο που θα έπρεπε να βρίσκεται στο σετ απαντήσεων απουσιάζει από αυτό, είναι πως δεν έχει δημιουργηθεί ο κανόνας της μορφής $initiatedAt$ ο οποίος θα το εντάξει στο σετ απαντήσεων. Απαιτείται δημιουργία αυτού του κανόνα και ένταξη του στο σετ απαντήσεων ώστε να καλυφθεί το άτομο αυτό.
 2. Στην αντίθετη περίπτωση, υπάρχει ένας κανόνας της μορφής $terminatedAt$ στο σετ απαντήσεων (είναι πιθανό να υπάρχουν και περισσότεροι του ενός κανόνες με παρόμοια

συμπεριφορά) ο οποίος δηλώνει πως δεν πραγματοποιείται το γεγονός στο οποίο υπάγεται το άτομο αυτό. Ο κανόνας πρέπει να εξειδικευτεί σε μία λιγότερο γενική δήλωση, ώστε να μην απορρίπτεται το άτομο από το σετ απαντήσεων.

- Εσφαλμένα Αληθές άτομο:

1. Ένα άτομο μπορεί να βρίσκεται στο σετ απαντήσεων χωρίς να αποτελεί μέρος της ερμηνείας η οποία εξετάζεται. Η πρώτη (πιθανή) αιτία είναι πως δεν υπάρχει κανόνας της μορφής `terminatedAt` ο οποίος να παύει ένα σύνθετο γεγονός. Σύμφωνα και με την αρχή της αδράνειας, το γεγονός εξακολουθεί να θεωρείται αληθές. Απαιτείται η δημιουργία ενός νέου `terminatedAt` κανόνα ώστε να πάψει την ισχύ του γεγονότος.
2. Αντίστοιχα ένας ή περισσότεροι κανόνες `initiatedAt` έχουν μία αρκετά γενική δήλωση ώστε να συμπεριλάβουν εσφαλμένα το άτομο αυτό στο σετ απαντήσεων. Για την αποφυγή του συμβάντος αυτού, απαιτείται η εξειδίκευση των κανόνων σε μία λιγότερο συγκεκριμένη δήλωση η οποία δεν θα περικλείει το άτομο αυτό.

Με βάση τα παραπάνω, ορίζεται η ακόλουθη ευριστική συνάρτηση για την αξιολόγηση ενός κανόνα r στο σύστημα OLED:

$$G(r) = \begin{cases} \frac{TP}{TP+FP} & \text{if } r \text{ is an initiatedAt rule,} \\ \frac{TP}{TP+FN} & \text{if } r \text{ is an terminatedAt rule} \end{cases} \quad (2.5)$$

2.2.3 Αλγοριθμική περιγραφή του OLED

Το σύστημα OLED χειρίζεται ένα σετ δεδομένων, όπου κάθε δεδομένο λογίζεται ως ανεξάρτητο παράδειγμα. Σκοπός του είναι η σύγχρονη μάθηση κανόνων, με την έννοια πως κάθε παράδειγμα θα τεθεί υπό επεξεργασία ακριβώς μία φορά, πριν την εξαγωγή της θεωρίας H με τη μορφή ενός συνόλου κανόνων οι οποίοι θα καλύπτουν τα παραδείγματα του σετ δεδομένων. Κάθε κανόνας διατηρεί, μεταξύ άλλων, δείκτες σχετικά με τα άτομα TP, FP και FN οι οποίοι ανανεώνονται σύμφωνα με τον ακόλουθο πίνακα:

Μία διαδικασία μάθησης κανόνων στο σύστημα OLED, συνίσταται στα εξής βήματα:

1. **Δημιουργία νέων κανόνων** Κατά την επέκταση της θεωρίας, με σκοπό να περιγράψει παραδείγματα τα οποία δεν καλύπτονται από τους ήδη υπάρχοντες κανόνες, απαιτείται η δη-

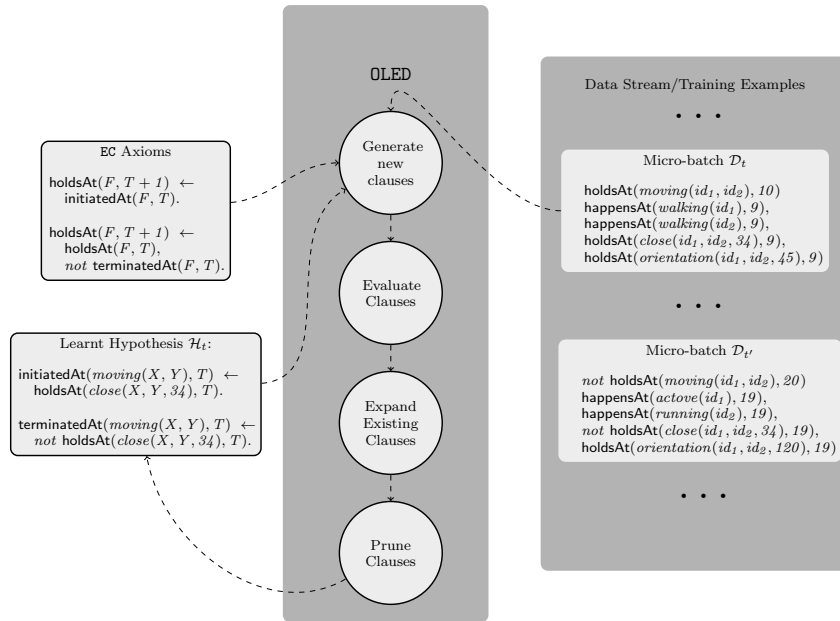
Είδος κανόνα	initiatedAt	terminatedAt
TP	Αυξάνεται κάθε φορά που ενεργοποιεί σωστά ένα σύνθετο γεγονός	Αυξάνεται κάθε φορά που δεν τερματίζει ένα σύνθετο γεγονός το οποίο εξακολουθεί να πραγματοποιείται
FP	Αυξάνεται κάθε φορά που ενεργοποιεί εσφαλμένα ένα σύνθετο γεγονός	
FN		Αυξάνεται κάθε φορά που τερματίζει εσφαλμένα ένα σύνθετο γεγονός το οποίο εξακολουθεί να πραγματοποιείται

Πίνακας 2.3 – Ανανέωση δεικτών στο σύστημα OLED

μιουργία νέων κανόνων. Ένας νέος κανόνας είναι κενός αρχικά ως προς το περιεχόμενο και φέρει μόνο την επικεφαλίδα της κλάσης του (`initiatedAt` ή `terminatedAt`). Επιπλέον, φέρει μία λίστα από δυνατές επεκτάσεις οι οποίες στην πορεία αξιολογούνται μαζί με τον ίδιο τον κανόνα στα παραδείγματα. Η διαδικασία της δημιουργίας νέου κανόνα υπάγεται στην Μη-παρατηρήσιμη Μάθηση Κατηγορημάτων[30], καθώς οι διατυπώσεις των γεγονότων στα παραδείγματα δεν είναι στη μορφή `initiatedAt`, `terminatedAt` αλλά στη μορφή `holdsAt`, `happensAt`. Η δημιουργία κανόνων στην επιθυμητή μορφή γίνεται με χρήση απαγωγής[9] η οποία συναντάται και σε άλλα συστήματα εξαγωγής κανόνων όπως το ILED[23].

2. **Εξειδίκευση κανόνων** Η εξειδίκευση ενός κανόνα πραγματοποιείται με την ενσωμάτωση επεκτάσεων στο σώμα του, επιλέγοντας κανόνες από τη λίστα των επεκτάσεων του. Ορίζεται μία παράμετρος d η οποία φράζει τον αριθμό των επεκτάσεων που μπορούν να προστεθούν σε ένα κανόνα ταυτόχρονα. Η επέκταση που διαλέγεται ικανοποιεί το κριτήριο του ορίου του Hoeffding όπως αυτό ορίστηκε στην προηγούμενη παράγραφο. Ως αποτέλεσμα, επιλέγεται η επέκταση r' , με το υψηλότερο G-score σύμφωνα με τη σχέση (2.5) όταν η απόκλιση του G-score από το αντίστοιχο G-score της δεύτερης κατά σειρά επέκτασης του κανόνα υπερβαίνει το τρέχον Hoeffding bound, το οποίο με τη σειρά του εξαρτάται από το πλήθος N των παραδειγμάτων στα οποία έχει αξιολογηθεί ο κανόνας και την παράμετρο δ , όπου $1 - \delta$ η βεβαιότητα με την οποία προσεγγίζεται το ορθό αποτέλεσμα.

Στην περίπτωση όπου οι 2 καλύτερες εξειδικεύσεις ενός κανόνα σημειώνουν παρόμοια scores (το οποίο συνήθως οφείλεται σε παρόμοιες διατυπώσεις του ίδιου κανόνα), και μόνο αφού ο κανόνας έχει αξιολογηθεί σε επαρκές πλήθος παραδειγμάτων, υιοθετείται ένα μικρότερο όριο απόκλισης το οποίο ισούται με τη μέση τιμή των Hoeffding bounds[38] ώστε ο κανόνας να μπορέσει να επεκταθεί χωρίς να μείνει στάσιμος λόγω αυτού του φαινομένου.



Σχήμα 2.2 – Βήματα μάθησης κανόνων στο σύστημα OLED

3. **Αποκοπή κανόνων** Δοθέντος ενός επαρκούς πλήθους παραδειγμάτων N στο οποίο ένας κανόνας έχει αξιολογηθεί, ο κανόνας αυτός είναι πιθανό να αποκοπεί από την τρέχουσα θεωρία εφόσον το G-score διατηρείται κάτω από ένα προκαθορισμένο όριο S_{min} .

Συνοψίζοντας όλα τα παραπάνω, ακολουθεί ο αλγόριθμος του συστήματος OLED:

Algorithm 1 Online Learning($\mathcal{I}, B, G, \delta, d, N_{min}, S_{min}$)

Input: \mathcal{I} : A stream of training interpretations; B : Background knowledge; G : Clause evaluation function; δ : Confidence for the Hoeffding test; d : Specialization depth; S_{min} : Clause G -score quality threshold.

```

1:  $H := \emptyset$ 
2: for each  $I \in \mathcal{I}$  do
3:   Update  $TP_r, FP_r, FN_r$  and  $N_r$  counts from  $I$ , for each  $r \in H$  and each  $r' \in \rho_d(r)$ ,
   where  $N_r$  denotes the number of examples on which  $r$  has been evaluated so far.
4:   if ExpandTheory( $B, H, I$ ) then
5:      $H \leftarrow H \cup \text{StartNewClause}(B, I)$ 
6:   else
7:     for each clause  $r \in H$  do
8:        $r \leftarrow \text{ExpandClause}(r, G, \delta)$ 
9:    $H \leftarrow \text{Prune}(H, S_{min})$ 
10: return  $H$ 
11: function StartNewClause( $B, I$ ):
12:   Generate a bottom clause  $\perp$  from  $I$  and  $B$ 
13:    $r := \text{head}(\perp) \leftarrow$ 
14:    $\perp_r := \perp$ 
15:    $N_r = FP_r = TP_r = FN_r := 0$ 
16:   return  $r$ 
17: function ExpandClause( $r, G, \delta$ ):
18:   Compute  $\epsilon = \sqrt{\frac{\ln(1/\delta)}{2N_r}}$  and let  $\bar{G}$  denote the mean value of a clause's  $G$ -score
19:   Let  $r_1$  be the best specialization of  $r$ ,  $r_2$  the second best and  $\Delta\bar{G} = \bar{G}(r_1) - \bar{G}(r_2)$ 
20:   Let  $\tau$  equal the mean value of  $\epsilon$  observed so far
21:   if  $\bar{G}(r_1) > \bar{G}(r)$  and [ $\Delta\bar{G} > \epsilon$  or  $\tau < \epsilon$ ]:
22:      $\perp_{r_1} := \perp_r$ 
23:     return  $r_1$ 
24:   else return  $r$ 
25: function prune( $H, S_{min}$ ):
26:   Remove from  $H$  each clause  $r$  for which  $S_{min} - \bar{G}(r) > \epsilon$ , where  $\epsilon$  is the current Hoeffding bound
27:   return  $H$ 

```

Πίνακας 2.4 – Αλγόριθμος του OLED

2.2.4 Αξιολόγηση θεωρίας στο σύστημα OLED

Στα συστήματα Μηχανικής Μάθησης, τα πιο γνωστά μέτρα αξιολόγησης αποτελούν οι μονάδες της ακρίβειας (precision), ανάκλησης (recall) και f-score, το οποίο προκύπτει μέσω της ανάκλησης και της ακρίβειας ως:

$$f - score = \frac{2 * precision * recall}{precision + recall} \quad (2.6)$$

Ο όρος της ακρίβειας αναφέρεται στο πλήθος των γεγονότων τα οποία κατετάγησαν σωστά σε μία κατηγορία, ως προς το συνολικό πλήθος των γεγονότων τα οποία κατετάγησαν σε αυτή την κατηγορία. Δεδομένου πως αντικείμενο εξέτασης στο σύστημα του OLED αποτελεί ένα σύνθετο γεγονός (complex event), για το οποίο εμφανίζονται άτομα τα οποία με τη σειρά του χαρακτηρίζονται ως αληθώς αληθή ή εσφαλμένα αληθή, για μία θεωρία η οποία αποτελείται από ένα σύνολο κανόνων μπορούμε να εκτιμήσουμε το μέτρο της ακρίβειας της ως εξής:

$$precision = \frac{\sum_{r \in H} TP_r}{\sum_{r \in H} (TP_r + FP_r)} \quad (2.7)$$

Αντιθέτως, ο όρος της ανάκλησης αφορά τη σύγκριση μεταξύ ατόμων τα οποία έχουν καταγεί σωστά σε μία κατηγορία ως προς το πραγματικό πλήθος των ατόμων που ανήκουν στην κατηγορία αυτή. Αντίστοιχα, μέσω του συνόλου των κανόνων οι οποίοι υπάγονται σε μία θεωρία, μπορούμε να ορίσουμε το μέτρο της ανάκλησης της ως εξής:

$$recall = \frac{\sum_{r \in H} TP_r}{\sum_{r \in H} (TP_r + FN_r)} \quad (2.8)$$

2.3 Δράστες

2.3.1 Το μοντέλο των Δραστών

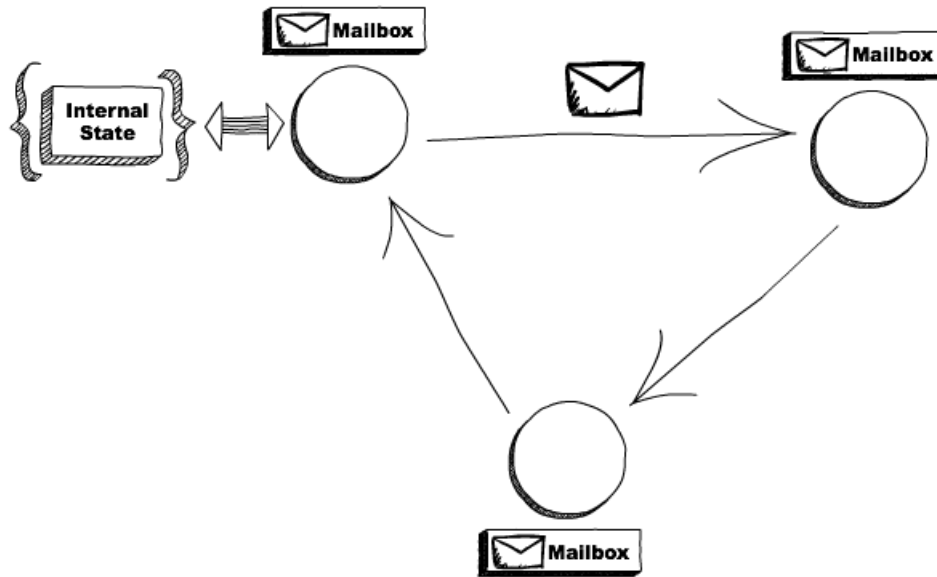
Οι Δράστες[16, 1] (Actors) αποτελούν ένα μαθηματικό, υπολογιστικό μοντέλο το οποίο χρησιμοποιείται σε δομές παράλληλου προγραμματισμού, και παρουσιάστηκαν πρώτη φορά στη βι-

βλιογραφία από τον CarlHewitt[15]. Κάθε Δράστης λειτουργεί ανεξάρτητα από τους υπόλοιπους και η ενεργοποίηση του σημαίνει αυτόματα την εκκίνηση ενός καινούριου υπολογιστικού νήματος (thread) από την υπολογιστική μονάδα η οποία τους χειρίζεται. Οι Δράστες αποτελούν μια θεμελιώδη μονάδα υπολογισμού η οποία οφείλει να περιέχει τα 3 ακόλουθα βασικά μέρη:

- Δυνατότητα επεξεργασίας δεδομένων, ώστε να μπορεί να χειρίζεται πληροφορίες και να οδηγείται σε αποτελέσματα.
- Αποθηκευτικό χώρο, ώστε να διατηρεί στη μνήμη της πληροφορίες που να μπορεί να χρησιμοποιεί όποτε αυτό είναι επιθυμητό.
- Δυνατότητα επικοινωνίας με τις υπόλοιπες μονάδες υπολογισμού, ώστε να μεταφέρει και να ενημερώνεται με πληροφορίες που αφορούν τα επιμέρους υπολογιστικά τμήματα ενός συστήματος.

Σε ένα σύστημα μπορούν να οριστούν αρκετοί διαφορετικοί τύποι Δραστών, με σκοπό να εκτελούν επιμέρους εργασίες του και να συνδυάζουν τα αποτελέσματά τους. Κάθε Δράστης ο οποίος χρησιμοποιείται σε ένα σύστημα οφείλει να έχει διακριτό όνομα σε σχέση με τους υπόλοιπους, το οποίο είναι και το μοναδικό του αναγνωριστικό στο σύστημα αυτό.

Το βασικό χαρακτηριστικό του μοντέλου των Δραστών είναι πως οι Δράστες ενός συστήματος μπορούν να επικοινωνούν μεταξύ τους, και να ανταλλάσσουν δεδομένα μέσω κατάλληλων μηνυμάτων[37]. Για το λόγο αυτό, κάθε Δράστης διατηρεί μία συγκεκριμένη διεύθυνση μέσω της οποίας οποιοδήποτε άλλο μέλος του συστήματος μπορεί να απευθυνθεί σε αυτόν (εφόσον είναι γνωστή σε αυτό), όπως επίσης μία δομή η οποία αποκαλείται γραμματοκιβώτιο μηνυμάτων (mailbox). Στη δομή αυτή διατηρούνται όλα τα μηνύματα που παραλαμβάνει ένας Δράστης. Ένας Δράστης αντιλαμβάνεται τα μηνύματα τα οποία υπάρχουν στο γραμματοκιβώτιο του ως ανεξάρτητα, αποκλειστικά πακέτα εντολών, με την έννοια πως χειρίζεται κάθε μήνυμα μεμονωμένα. Επιπλέον, εάν λάβει κάποιο μήνυμα μέσα στο χρονικό διάστημα στο οποίο επεξεργάζεται ένα προηγούμενο μήνυμα, απλώς θα το αποθηκεύσει στο γραμματοκιβώτιο του και θα το χειριστεί μελλοντικά και μόνο αφού έχει ολοκληρώσει τις εντολές του τρέχοντος μηνύματος. Η σειρά με την οποία επεξεργάζονται οι Δράστες τα μηνύματα είναι FCFS (First Come First Served), επομένως κάθε φορά επιλέγεται από το γραμματοκιβώτιο για επεξεργασία το μήνυμα που βρίσκεται σε αυτό μεγαλύτερο χρονικό διάστημα από τα υπόλοιπα.



Σχήμα 2.3 – Διάγραμμα επικοινωνίας Δραστών: Η επικοινωνία μεταξύ των Δραστών είναι ασύγχρονη, και συμπεριλαμβάνει αποστολή μηνυμάτων. Τα μηνύματα αποθηκεύονται στο γραμματοκιβώτιο ενός Δράστη μέχρι την επεξεργασία τους.

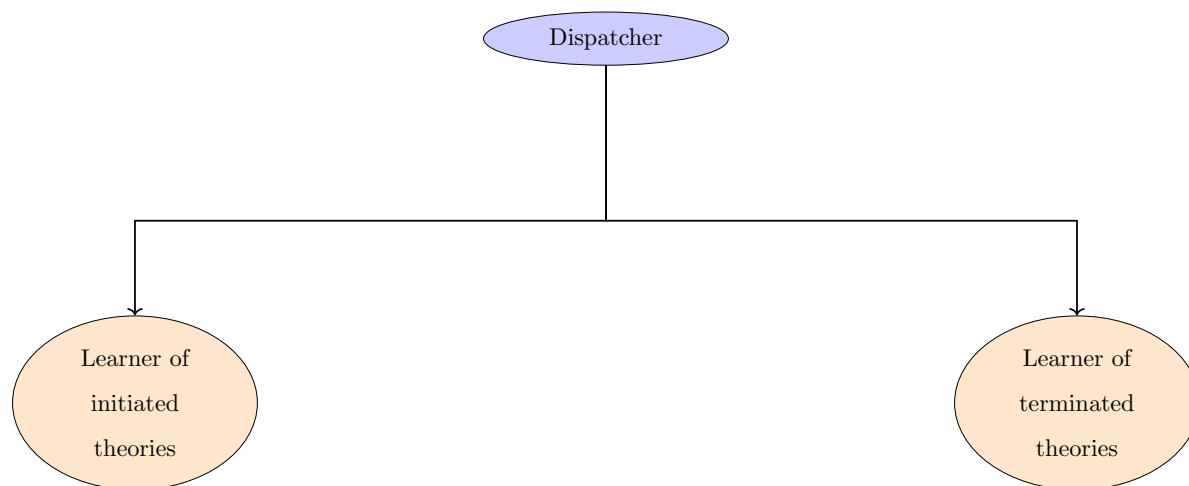
Σε ένα σύστημα Δραστών, μπορούν να υπάρξουν πολλοί διαφορετικοί τύποι μηνυμάτων, οι οποίοι με τη σειρά τους μπορούν να περικλείουν διαφορετικά δεδομένα τόσο ως προς το πλήθος τους όσο και ως προς την ποιότητα τους. Κάθε Δράστης με τη σειρά του, μέσω κατάλληλου σχεδιασμού είναι σε θέση να επεξεργαστεί ποικίλους τύπους μηνυμάτων, εφαρμόζοντας διαφορετική αντιμετώπιση στον κάθε ένα από αυτούς. Βασικό επομένως κομμάτι στο σχεδιασμό ενός Δράστη αποτελεί ο κατάλληλος ορισμός της συνάρτησης παραλαβής μηνυμάτων του (receive function). Στη συνάρτηση παραλαβής μηνυμάτων ορίζεται ακριβώς η συμπεριφορά του Δράστη όσον αφορά συγκεκριμένους τύπους μηνυμάτων, και οφείλει να είναι σαφώς διατυπωμένη πριν την ενεργοποίησή του. Ένας Δράστης είναι σε θέση να δεχθεί και να αποθηκεύσει στο γραμματοκιβώτιο του μηνύματα για τα οποία δεν υπάρχει συνταγή αντιμετώπισης τους στη συνάρτηση παραλαβής, όμως όταν έρθει η σειρά τους για επεξεργασία απλώς θα τα αμελήσει και θα προχωρήσει σε επόμενα μηνύματα.

2.3.2 Διάρθρωση του OLED με τη χρήση Δραστών

Το σύστημα της Μάθησης Ορισμών Γεγονότων σε πραγματικό χρόνο θεωρείται ένα μονολιθικό σύστημα, καθώς ένας υπολογιστικός κόμβος αναλαμβάνει την επεξεργασία ενός σετ δεδομένων προκειμένου να μάθει κανόνες οι οποίοι να περιγράψουν επαρκώς τα παραδείγματα του. Όμως ακόμα και στο μονολιθικό αυτό σύστημα διακρίνεται μια μορφή παραλληλοποίησης, καθώς με την ενεργοποίησή του θα εκκινήσουν δύο διακριτές διαδικασίες: μία για τη μάθηση κανόνων

initiatedAt και αντίστοιχα μία για τη μάθηση κανόνων terminatedAt. Μπορούμε επομένως να αναθέσουμε την κάθε μία από αυτές τις διεργασίες σε έναν Δράστη, αποκλειστικά υπεύθυνο για την εκτέλεση του αλγορίθμου της μάθησης κανόνων, όπως αυτός περιγράφηκε στην προηγούμενη ενότητα. Ο Δράστης αποκαλείται Μαθητής (Learner) και στο μονολιθικό σύστημα εκτελεί τον αλγόριθμο του OLED για τα δεδομένα ολόκληρου του σετ δεδομένων.

Επιπρόσθετα, μπορούμε να ορίσουμε ένα Δράστη-γονέα (parent-Actor) του Learner, ο οποίος στο μονολιθικό σύστημα είναι υπεύθυνος για τη διανομή των δεδομένων στους δύο Learners οι οποίοι μαθαίνουν ξεχωριστά τους initiated και terminated κανόνες αντίστοιχα. Ο Δράστης αυτός αποκαλείται Αποστολέας (Dispatcher) και εκκινείται και λειτουργεί σε υψηλότερο ιεραρχικό επίπεδο από τους δύο Learners.



Σχήμα 2.4 – Ιεραρχία Δραστήων στο σύστημα OLED

Μέσω της παραπάνω διάρθρωσης, θα μπορέσουμε να επεκτείνουμε μελλοντικά το σύστημα OLED, διατηρώντας τον Dispatcher ως έχει και θα αυξήσουμε τον αριθμό των Learners, οι οποίοι θα αναλαμβάνουν τη μάθηση κανόνων σε ένα τμήμα του αρχικού σετ δεδομένων.

2.4 Σχετικές εργασίες

Στη βιβλιογραφία συναντώνται αρκετές περιπτώσεις καταναμημένων συστημάτων τα οποία χρησιμοποιούν κριτήρια επικοινωνίας μεταξύ των παράλληλων κόμβων τους, όπως επίσης καταναμημένα συστήματα μάθησης.

Οι Mosche et al.[12], υιοθετούν γεωμετρική επίβλεψη[14, 25] των ροών δεδομένων ενός καταναμημένου συστήματος για να αποφασίσουν πότε πραγματοποιείται επικοινωνία μεταξύ των

κόμβων. Σε προκαθορισμένα σημεία, πραγματοποιείται συγχρονισμός των παράλληλων κόμβων ώστε να τεθεί ένα μέτρο αναφοράς β_{global} με το οποίο κάθε κόμβος συγκρίνει τα τοπικά του αποτελέσματα β_{local} . Εφόσον τα δύο αυτά μοντέλα αποκλίνουν κατά ένα όριο ϵ ενεργοποιείται η διαδικασία επαναφοράς του συστήματος εντός των προκαθορισμένων παραμέτρων του, είτε μέσω επικοινωνίας των κόμβων είτε μέσω επαναπροσδιορισμού του μοντέλου β_{global} .

Οι Kamp et al.[20] προτείνουν δυναμικά μοντέλα συγχρονισμού, με βασικά κριτήρια i) την αθροιστική απώλεια ενός κατανεμημένου συστήματος (δηλαδή το κατά πόσο αποκλίνουν οι προβλέψεις των τοπικών μοντέλων από την πραγματικότητα) και ii) την επιβάρυνση του δικτύου από την ανταλλαγή πληροφοριών μεταξύ των επιμέρους learners. Σε κάθε παραβίαση ενός προκαθορισμένου ορίου από το τοπικό μοντέλο ενός κόμβου, επικοινωνούν με τον ελάχιστο δυνατό αριθμό κόμβων (ξεκινώντας δηλαδή από έναν κόμβο του συστήματος και αυξάνοντας κατά ένα κόμβο τη φορά) χρησιμοποιώντας πληροφορίες από τα μοντέλα τους για να εξομαλύνουν το αρνητικό μοτέλο σε ένα διορθωμένο μοντέλο το οποίο επανατοποθετείται σε όλους τους κόμβους οι οποίοι συμμετείχαν στη διαδικασία. Τα όρια προς εξέταση του δυναμικού συστήματος επαναπροσδιορίζονται[19], ενώ μια βελτιωμένη εκδοχή του πρωτοκόλλου αυτού[21] κάνει χρήση μοντέλων πυρήνων[33] που εμπίπτουν στους Αναπαράξιμο Χώρο Hilbert[2, 31] (Reproducing Kernel Hilbert Space).

Προσεγγίσεις σχετικά με μάθηση σε κατανεμημένα περιβάλλοντα συναντώνται και σε άλλες επιστημονικές περιοχές, όπως η Θεωρία Παιγνίων. Με χρήση πολυπρακτορικών συστημάτων[35], οι Marden et al.[29] απεικονίζουν την κατάσταση κάθε πράκτορα σε ένα γύρο i ως ένα ζεύγος $(\bar{a}_i, \bar{u}_i, m_i)$ όπου τα 2 πρώτα μεγέθη αντιστοιχούν στην επιλογή του πράκτορα και την απόδοση της στο γύρο αυτό αντίστοιχα, ενώ η διάθεση m_i (με δυαδική τιμή content ή discontent) καθορίζει την πιθανότητα με την οποία κάθε πράκτορας μεταβαίνει σε μία νέα κίνηση a_i με $a_i \neq \bar{a}_i$. Σε επόμενη εργασία[32] οι πιθανές τιμές της μεταβλητής m_i αυξάνονται με αποτέλεσμα μεγαλύτερο αριθμό επιλογών και συνδυασμών των πιθανών κινήσεων κάθε πράκτορα.

Κεφάλαιο 3

Πρώτες επεκτάσεις στο σύστημα OLED

Στο κεφάλαιο αυτό παρουσιάζονται οι πρώιμες επεκτάσεις του συστήματος OLED. Αρχικά γίνεται αναφορά στα κριτήρια που ορίστηκαν για την επικοινωνία μεταξύ των κόμβων υπολογισμού(3.1). Εν συνεχεία, παρουσιάζεται η υλοποίηση του μοντέλου αναμονής απαντήσεων(3.2) το οποίο εκμεταλλεύεται την ικανότητα των Δραστών να αιτούνται και να λαμβάνουν δεδομένα σε συγκεκριμένα σημεία της υπολογιστικής τους επεξεργασίας, καθώς και οι λόγοι για τους οποίους εγκαταλείφθηκε αυτή η προσέγγιση. Εν κατακλείδι, περιγράφονται δύο διακριτές υλοποιήσεις οι οποίες βασίζονται σε χρήση κοινών δεδομένων μεταξύ δραστών, σε περιβάλλοντα κατανεμημένης(3.3) και παράλληλης επεξεργασίας(3.4). Οι εν λόγω υλοποιήσεις προέκυψαν από την ανάγκη να αντιμετωπιστεί ένας βασικός περιορισμός του μοντέλου των Δραστών.

3.1 Κριτήρια επικοινωνίας

Ως προς την σχεδίαση ενός κατανεμημένου συστήματος, είναι σημαντικό να αντιληφθούμε πως κάθε υπολογιστικό νήμα οφείλει να λειτουργεί ανεξάρτητα από τα υπόλοιπα, και ταυτόχρονα να έχει διαθέσιμη όλη την πληροφορία που απαιτείται για τη σωστή λειτουργία του, όπως ακριβώς θα συνέβαινε σε έναν μονολιθικό σύστημα. Ως αποτέλεσμα, πριν τη σχεδίαση κατανεμημένων αρχιτεκτονικών και αλγορίθμων, διακρίνουμε περιπτώσεις όπου θα πρέπει να εφαρμοστεί επικοινωνία μεταξύ των υπολογιστικών νημάτων τα οποία επεξεργάζονται ανεξάρτητα μεταξύ τους τμήματα του αρχικού σετ δεδομένων.

3.1.1 Νέοι κανόνες

Ο OLED χειρίζεται ένα σετ από παραδείγματα, για τα οποία στοχεύει να δημιουργήσει και να επεκτείνει κανόνες (clauses) τα οποία να μπορούν να τα καλύψουν. Έχοντας αρχικοποιηθεί με μία θεωρία κενή από clauses, είναι δεδομένο πως όταν θα επεξεργαστεί το πρώτο θετικό παράδειγμα ως προς το high-level event που εξετάζει, θα χρειαστεί να δημιουργήσει ένα κενό κανόνα (bottom rule) για να το καλύψει. Η περίπτωση αυτή συναντάται ακόμα και όταν έχουν δημιουργηθεί μη κενές θεωρίες οι οποίες αξιολογούνται σε παραδείγματα των οποίων οι περιγραφές δεν καλύπτονται από τους διαθέσιμους σε εκείνο το σημείο κανόνες. Ένας bottom rule αποτελείται μόνο από τη δήλωση της στοχευμένης κλάσης την οποία μαθαίνει ένας Learner Actor (initiatedAt:- / terminatedAt:-) και φέρει μαζί του υποψήφιας επεκτάσεις (refinements). Οι επεκτάσεις αποτελούν εξειδικεύσεις του ίδιου του κανόνα κατά ένα βήμα εξειδίκευσης και παρότι δεν αποτελούν μέρος της θεωρίας, αξιολογούνται μαζί με τον κανόνα στον οποίο ανήκουν σε κάθε παράδειγμα.

Περνώντας από το μονολιθικό σύστημα στο καταναμημένο, οφείλουμε να εξασφαλίσουμε πως τα bottom rules που θα δημιουργηθούν κατά τη μελέτη ενός υποσυνόλου των δεδομένων από έναν Learner μαζί με τα refinements που φέρουν θα μεταφερθούν με τη μορφή αντιγράφων στους υπόλοιπους Learners οι οποίοι θα τα ενσωματώσουν στην τρέχουσα θεωρία τους.

3.1.2 Clause υποψήφιο για επέκταση

Σε κάθε παράδειγμα ένα clause αξιολογείται μαζί με τα refinements του, τα οποία ταξινομούνται σε φθίνουσα σειρά με κριτήριο το score τους. Επιπρόσθετα, εκτελείται έλεγχος μεταξύ των δύο καλύτερων refinements, ώστε να διαπιστωθεί εάν η διαφορά της μέσης τιμής των $G - scores$ τους ξεπερνάει το Hoeffding bound όπως αυτό ορίστηκε στο κεφάλαιο 2.

Στην περίπτωση του μονολιθικού συστήματος, θετική απόκριση σε αυτό τον έλεγχο σημαίνει πως το clause επεκτείνεται στο καλύτερο refinement του. Στην περίπτωση του καταναμημένου συστήματος, η ποιοτική σημασία ενός θετικού ελέγχου είναι πως το clause το οποίο εξετάζεται σε ένα υποσύνολο δεδομένων είναι αρχούντως εύστοχο για να περιγράψει τα παραδείγματα του, χωρίς όμως να είναι διαθέσιμη η γνώση για τις επιδόσεις των αντιγράφων τους στα υπόλοιπα υποσύνολα δεδομένων.

Για να αξιολογηθούν σωστά οι υποψήφιας επεκτάσεις ενός κανόνα, απαιτείται η γνώση των επιδόσεων τους σε όλα τα παραδείγματα στα οποία έχουν εξεταστεί μέχρι εκείνο το σημείο.

Επομένως, όταν ένας κανόνας ικανοποιεί τοπικά το κριτήριο του Hoeffding bound, πρέπει να αντληθούν οι επιδόσεις των αντιγράφων τους από τους υπόλοιπους Learners που τα έχουν ενσωματωμένα στη δική τους θεωρία. Πιο συγκεκριμένα, χρειάζονται οι δείκτες TP, FP ή FN (ανάλογα με το αν ο κανόνας ανήκει στην κλάση initiatedAt ή terminatedAt) του κάθε αντιγράφου και των επεκτάσεων του, καθώς και το πλήθος των παραδειγμάτων στα οποία έχει αξιολογηθεί, καθώς η τιμή του Hoeffding bound εξαρτάται από αυτό. Ενδεικτικά για έναν κανόνα r_i (και κάθε επέκταση του ref_i) ο οποίος ανήκει στη θεωρία του i -ου Learner του κατανεμημένου συστήματος, η ανανέωση των δεικτών TP βάσει των αντιγράφων τους $\{ r'_j, j \neq i \}$ και $\{ ref'_j, j \neq i \}$ γίνεται ως εξής:

$$TP_{r_i} = TP_{r_i} + \sum_{j, j \neq i}^N TP_{r'_j}, \quad TP_{ref_i} = TP_{ref_i} + \sum_{j, j \neq i}^N TP_{ref'_j} \quad (3.1)$$

3.1.3 Εξειδίκευση ενός clause

Το σύστημα του OLED μαθαίνει και σκοράρει κανόνες οι οποίοι εμπεριέχονται στη θεωρία του. Μόλις ικανοποιηθεί το κριτήριο του Hoeffding bound, ένα clause αντικαθίσταται από τη βέλτιστη εξειδίκευση του. Επομένως, σε έναν κατανεμημένο αλγόριθμο θα πρέπει να πληρούνται οι δύο ακόλουθες προϋποθέσεις:

- Η προσθήκη ενός εξειδικευμένου κανόνα στη θεωρία όλων των Learners.
- Η αφαίρεση της προηγούμενης (και πιο γενικής) μορφής του εξειδικευμένου αυτού κανόνα από τη θεωρία όλων των Learners.

Κρίνεται επομένως αναγκαίο να υπάρχει ενημέρωση από τον Learner που προχώρησε στην επέκταση του clause προς τους υπόλοιπους, παρέχοντας την εξειδικευμένη μορφή του και ενημερώνοντας τους να αφαιρέσουν το αντίγραφο της προηγούμενης μορφής του από τη θεωρία τους.

3.1.4 Αποκοπή (pruning) ενός κανόνα

Εφόσον ένας κανόνας αποκοπεί από τη θεωρία ενός Learner, πρέπει να υπάρξει ενημέρωση των υπολοίπων Learners οι οποίοι δρουν στο κατανεμημένο σύστημα, ώστε να αφαιρεθεί ο κανόνας αυτός και από τη δική τους θεωρία.

3.2 Μοντέλο Αναμονής Απαντήσεων

3.2.1 Η δομή Future

Το μοντέλο των Δραστών σχεδιάστηκε και δομήθηκε ώστε κάθε υπολογιστικό τμήμα να εκτελείται ανεξάρτητα των υπολοίπων. Όπως αναφέραμε και στο κεφάλαιο 2, κάθε Δράστης αντιλαμβάνεται τα μηνύματα που βλέπει στο γραμματοκιβώτιο του ως ένα ξεχωριστό πακέτο εντολών. Ως αποτέλεσμα, δεν μπορεί να γίνει καμμία παρέμβαση στα δεδομένα που διαθέτει κατά τη διάρκεια χειρισμού του, μέσω επιπλέον μηνυμάτων στον ίδιο Δράστη. Στην περίπτωση αλγορίθμων οι οποίοι χρειάζονται ανανέωση των δεδομένων που χειρίζονται (όπως ακριβώς συμβαίνει στην περίπτωση του OLED, όπου μεταξύ άλλων είναι αναγκαίο να γνωρίζουμε τις εξειδικεύσεις που πραγματοποιούνται στα clauses μια θεωρίας) προτείνεται η χρήση περαιτέρω δομών[36].

Η χρήση της δομής Future επιτρέπει σε ένα Δράστη:

- Να παύσει τη λειτουργία του για ένα προκαθορισμένο χρονικό διάστημα.
- Να ζητήσει πληροφορίες από άλλους δράστες, μέσω κατάλληλων μηνυμάτων, τα οποία μπορούν να περιλαμβάνουν δεδομένα. Με τον τρόπο αυτό, μπορεί να δώσει τα απαραίτητα στοιχεία για να επεξεργαστούν από έναν άλλο κόμβο για την εξαγωγή των απαιτούμενων πληροφοριών.
- Να λάβει απάντηση και να συνεχίσει την επεξεργασία (διαθέτοντας πλέον την πρόσθετη πληροφορία) από το σημείο του αλγορίθμου στο οποίο έθεσε τον εαυτό του σε αναμονή.

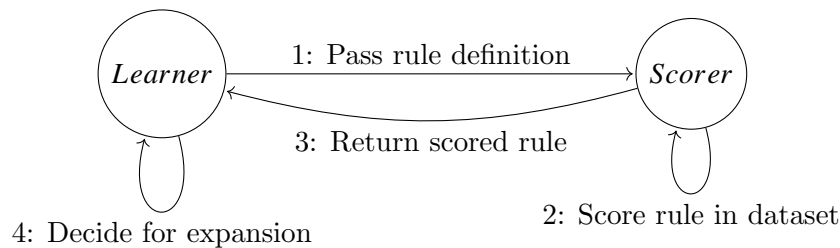
Να παρατηρηθεί πως για την κατάλληλη αξιοποίηση της δομής Future απαιτείται προσεκτικός σχεδιασμός και πρόβλεψη για τις περιπτώσεις όπου τα μηνύματα δε θα παραλειφθούν εντός του (υποχρεωτικά) προκαθορισμένου χρονικού πλαισίου.

3.2.2 Το Μοντέλο Αναμονής Απαντήσεων

Με βάση τα κριτήρια επικοινωνίας όπως αυτά ορίστηκαν στην παράγραφο 3.1 και γνωρίζοντας τις δυνατότητες αλλά και τους περιορισμούς που εσωκλείει η δομή Future προχωρήσαμε στη σχεδίαση της πρώτης επέκτασης του συστήματος OLED. Δεδομένου ότι η χρήση της δομής αυτής είναι εφικτή όταν ένας Δράστης θέτει σημεία αναμονής στον εαυτό του, γίνεται εύκολα αντιληπτό πως επικοινωνία μπορεί να υπάρξει μόνο στην περίπτωση όπου ένα clause πληρεί το κριτήριο του

Hoeffding bound(3.1.2) και είναι απαραίτητο για το Δράστη να μάθει τα στατιστικά των αντιγράφων του στα υπόλοιπα υποσύνολα του αρχικού σετ δεδομένων. Καθώς η μάθηση δε μπορεί να ολοκληρωθεί πριν την εξάντληση των παραδειγμάτων, δεν μπορούμε να ενημερώσουμε τους υπόλοιπους Learners για νέους(3.1.1) ή εξειδικευμένους(3.1.3) κανόνες που δημιουργήθηκαν σε διαφορετικό dataset από αυτό που τους ανήκει.

Το Μοντέλο Αναμονής Απαντήσεων (Pause-and-Wait Model) διατηρεί τη δομή του OLED και προσθέτει ένα δράστη εν ονόματι Scorer ο οποίος ιεραρχικά ανήκει στο ίδιο επίπεδο με τον Learner. Σκοπός ενός Scorer είναι να λάβει ένα clause για το οποίο κάποιος Learner διαπίστωσε ότι η απόκλιση των scores μεταξύ των 2 καλύτερων εξειδικεύσεων του ξεπερνάει την τρέχουσα τιμή του Hoeffding bound και να αξιολογήσει το clause αυτό στα δεδομένα ενός διαφορετικού subset από αυτό που ανήκει στον Learner, επιστρέφοντας τα scores του.



Σχήμα 3.1 – Επικοινωνία μεταξύ Δραστών τύπου Learner και Scorer

Εφόσον ο Learner λάβει απαντήσεις από όλα τα sets, θα είναι σε θέση να κρίνει κατά πόσο θα επεκτείνει το clause ή θα το διατηρήσει στην τρέχουσα μορφή του.

3.2.3 Ελαττώματα του μοντέλου

Η χρήση δομών όπως η Future σε συνδυασμό με το μοντέλο των Δραστών συναντάται στη βιβλιογραφία ως μία μέθοδος να διατηρούνται ενεργά τα νήματα υπολογισμού παρά την ανάγκη να προμηθευθούν με νέα δεδομένα. Παρότι μπορεί να είναι αρκετά χρήσιμες σε αρχιτεκτονικές στις οποίες ως επί το πλείστον δε συναντώνται τακτικές ανταλλαγές μηνυμάτων, σε περιπτώσεις όπως το σύστημα του OLED οδηγούν σε αποτέλεσμα αντίστροφο του αναμενόμενου, για τους εξής λόγους:

1. Ο χρόνος απόκρισης δεν μπορεί να προσεγγιστεί επακριβώς. Σαν αποτέλεσμα, κατά τη διάρκεια πειραμάτων αντιμετωπίσαμε το φαινόμενο κάποια μηνύματα να αργήσουν να παραδοθούν, με αποτέλεσμα να εκπνεύσει το χρονικό διάστημα αναμονής και η ποιοτική και χρονική απόδοση του συστήματος να ελαττωθεί.
2. Μεγαλύτερος αριθμός μηνυμάτων συνεπάγεται αύξηση της καθυστέρησης στο σύστημα, λόγω

τως περισσότερων χρονικών περιόδων κατά τις οποίες η μάθηση κανόνων παραμένει ανενεργή. Επομένως, μοιράζοντας το σετ δεδομένων σε μεγάλο αριθμό πυρήνων το σύστημα οδηγήθηκε σε μεγαλύτερο χρόνο εκτέλεσης από ότι στη μονολιθική του μορφή, αποτέλεσμα αντίστροφο του επιθυμητού.

3. Αναφέραμε νωρίτερα πως δεν καλύπτονται όλες οι περιπτώσεις επικοινωνίας , όπως αυτές διατυπώθηκαν στην παράγραφο 3.1.

Ως αποτέλεσμα απαιτούνται σχεδιασμοί οι οποίοι να αντιμετωπίζουν τα παραπάνω ελαττώματα.

3.3 Μοντέλα Χρήσης Κοινών Δεδομένων

3.3.1 Διαφοροποίηση στον αλγόριθμο του OLED για χρήση σε καταναμημένα συστήματα

Ο αλγόριθμος του OLED όπως περιγράφηκε στην ενότητα 2.2, συνίσταται από συνεχή ανάγνωση των δοθέντων παραδειγμάτων (έως την εξάντληση τους) και διενέργεια συγκεκριμένων διαδικασιών σε κάθε παράδειγμα. Συνοπτικά, μπορούμε να περιγράψουμε τις διαδικασίες αυτές ως εξής:

1. Έλεγχος σχετικά με το κατά πόσο ένα παράδειγμα καλύπτεται από την υπάρχουσα θεωρία του Learner, και δημιουργία καινούριου bottom rule στην περίπτωση αρνητικού ελέγχου.
2. Αξιολόγηση των κανόνων και ανανέωση των δεικτών tps, fps (για initiatedAt κανόνες) ή fns(για terminatedAt κανόνες) οι οποίοι επηρεάζουν το attribute score του κάθε κανόνα.
3. Μεμονωμένος έλεγχος του ορίου του Hoeffding για κάθε κανόνα. Στον έλεγχο αυτό συγχρίνεται η αριθμητική διαφορά των scores των δύο καλύτερων εξειδικεύσεων (με κριτήριο την τιμή του score τους) του κανόνα για το εάν είναι μεγαλύτερη από την τιμή του ϵ όπως αυτή ορίζεται στην ενότητα (2.2.1).
4. Ανανέωση κάθε κανόνα που επαληθεύει τον παραπάνω έλεγχο στην καλύτερη εξειδίκευση του.

Τα βήματα 1 και 3 σχετίζονται με τα κριτήρια επικοινωνίας που έχουν οριστεί. Προχωρώντας στο σχεδιασμό ενός νέου, καταναμημένου συστήματος, στο βήμα 1 αρκεί η ενημέρωση όλων

των Δραστών με αντίγραφα του καινούριου κανόνα. Αντιθέτως, το βήμα 3 πρέπει να επεκταθεί σε μια διαδικασία όπου:

- Ο Δράστης που εντόπισε έναν κανόνα που ικανοποιεί το κριτήριο του Hoeffding bound θα ενημερώνει τους υπόλοιπους Δράστες, ζητώντας τα στατιστικά των αντιγράφων τους.
- Κάθε Δράστης μεμονωμένα να ανταποκρίνεται μέσω μηνυμάτων με τα στατιστικά του αντιγράφου που ανήκει στη θεωρία τους.
- Ο αρχικός Δράστης, εφόσον συλλέξει στατιστικά από όλα τα αντίγραφα θα επαναλαμβάνει τον έλεγχο και αναλόγως του αποτελέσματος του θα προχωράει στην επέκταση ή μη του κανόνα.
- Σε περίπτωση επέκτασης να υπάρξει ενημέρωση όλων των Δραστών του συστήματος, ώστε κάθε αντίγραφο της αρχική μορφής να αντικατασταθεί με ένα αντίγραφο της εξειδικευμένης μορφής.

Προκύπτει επομένως η ανάγκη διατύπωσης επιπλέον μηνυμάτων για τη συνάρτηση **receive** κάθε Δράστη ώστε να υλοποιηθούν τα παραπάνω βήματα. Ως αποτέλεσμα ,προκύπτει σε πρώτο βαθμό ο παρακάτω αλγόριθμος:

<p>(a) <code>OnlineLearning($\mathcal{I}, B, G, \delta, d, N_{min}, S_{min}$)</code></p> <p>Input: \mathcal{I}: A stream of training interpretations; B: Background knowledge; G: Clause evaluation function; δ: Confidence for the Hoeffding test; d: 1: Specialization depth; S_{min}: Clause G-score quality threshold.</p> <pre> 1: $H := \emptyset$ 2: for each $I \in \mathcal{I}$ do 3: Update TP_r, FP_r, FN_r and N_r counts from I, for each $r \in H$ and each $r' \in \rho_d(r)$, where N_r denotes the number of examples on which r has been evaluated so far. 4: if <code>ExpandTheory(B, H, I)</code> then 5: $H \leftarrow H \cup \text{StartNewClause}(B, I)$ 6: else 7: for each clause $r \in H$ do 8: $r \leftarrow \text{ExpandClause}(r, G, \delta)$ 9: $H \leftarrow \text{Prune}(H, S_{min})$ 10: return H 11: function <code>StartNewClause(B, I)</code>: 12: Generate a bottom clause \perp from I and B 13: $r := \text{head}(\perp) \leftarrow$ 14: $\perp_r := \perp$ 15: $N_r = FP_r = TP_r = FN_r := 0$ 16: Send copies of r to the other learners with <code>NewRule(r)</code> message 17: return r 18: function <code>ExpandClause(r, G, δ)</code>: 19: Compute $\epsilon = \sqrt{\frac{\ln(1/\delta)}{2N_r}}$ and let \overline{G} denote the mean value of a clause's G-score 20: Let r_1 be the best specialization of r, r_2 the second best and $\Delta\overline{G} = \overline{G}(r_1) - \overline{G}(r_2)$ 21: Let τ equal the mean value of ϵ observed so far 22: if $\overline{G}(r_1) > \overline{G}(r)$ and $[\Delta\overline{G} > \epsilon \text{ or } \tau < \epsilon]$: 23: Request r copies stats from other Learners using <code>Request(r)</code> message 24: $wait := true$ 25: While $wait = true$ 26: Wait until all responses have arrived 27: if $\overline{G}(r_1) > \overline{G}(r)$ and $[\Delta\overline{G} > \epsilon \text{ or } \tau < \epsilon]$: 28: $\perp_{r_1} := \perp_r$ 29: return r_1 30: else return r 31: function <code>prune(H, S_{min})</code>: 32: Remove from H each clause r for which $S_{min} - \overline{G}(r) > \epsilon$, where ϵ is the current Hoeffding bound 33: return H </pre>	<p>(b) <code>Receive($message$)</code></p> <p>Input: $message$: A message received from an other learner</p> <pre> function <code>Receive($message$)</code>: 2: Case <code>NewRule(r)</code>: 3: $H \leftarrow H \cup r$ 4: Case <code>Request(r)</code>: 5: Search H for copy r' of r 6: Respond with r' 7: Case <code>Response(r')</code>: 8: Search in H for rule r that responds to r 9: Update TP_r, FP_r, FN_r and N_r counts of r and of each $r' \in \rho_d(r)$ 10: if all learners have responded: 11: $wait := false$ </pre>
--	---

Alg. 3.1 – Τροποποιημένος αλγόριθμος της μάθησης κανόνων με χρήση κοινών δεδομένων

3.3.2 Περιορισμοί του μοντέλου των Δραστών

Το μοντέλο των Δραστών έχει υλοποιηθεί κατά τέτοιο τρόπο ώστε κάθε υπολογιστικό νήμα (άρα και κάθε Δράστης) να λειτουργεί ανεξάρτητα από τα υπόλοιπα, χωρίς να υπάρχει η δυνατότητα απευθείας επέμβασης στον αλγόριθμο του. Σαν αποτέλεσμα, ένας Δράστης αντιμετωπίζει το mailbox του ως ξεχωριστά πακέτα εντολών, τα οποία εκτελεί μεμονωμένα, και δεν είναι σε θέση να χρησιμοποιήσει την πληροφορία την οποία μπορεί να μεταφέρει ένα μήνυμα το οποίο είναι μεταγενέστερο αυτού που επεξεργάζεται. Αντιθέτως, ολοκληρώνει τις εντολές που έχουν οριστεί στη συνάρτηση receive για τον τύπο αυτού του μηνύματος, και μόνο εφόσον ολοκληρωθεί η διαχείριση του εξετάζει το μήνυμα που βρίσκεται στη κορυφή του mailbox του. Ένας Learner όπως αυτός ορίστηκε στην προηγούμενη παράγραφο δεν μπορεί να λειτουργήσει, καθώς τη στιγμή που θα μπει σε στάση αναμονής (waiting state) δεν θα μπορέσει να λάβει κάποιο επόμενο μήνυμα από τη στιγμή που δεν έχει ολοκληρώσει την επεξεργασία ενός τρέχοντος μηνύματος, και δεν θα καταφέρει να τερματίσει την εκτέλεση του.

3.3.3 Ο Δράστης τύπου Modifier

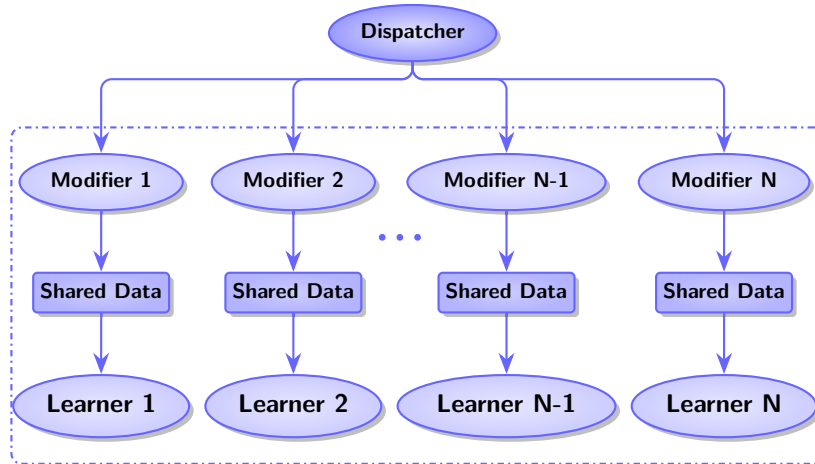
Για την αντιμετώπιση αυτού του φαινομένου, και διατηρώντας τα βήματα της μάθησης ως έχουν, ορίζεται ένα νέο είδος Δραστών (Επεξεργαστές ; Modifiers), οι οποίοι έχουν πρόσβαση από κοινού με τους Learners σε ορισμένα δεδομένα, και οι αλλαγές που πραγματοποιούνται σε αυτά μπορούν να γίνουν ορατές χωρίς αποστολή μηνύματος στα mailbox των Learners.

Με την προσθήκη Modifiers προχωρούμε στη διατύπωση δύο μοντέλων Χρήσης Κοινών Δεδομένων μεταξύ Δραστών (σε παρόμοια λογική με τη δουλειά των Lesani και Lain[27]), το Μοντέλο Ένα-προς-Ένα (3.3.4) το οποίο συσχετίζει έναν Modifier ανά Learner, και το Μοντέλο Ένα-προς-N το οποίο συσχετίζει έναν κοινό Modifier για όλους τους Learners οι οποίοι συμμετέχουν στη διαδικασία μάθησης κανόνων.

3.3.4 Μοντέλο Ένα-προς-Ένα

Στο μοντέλο αυτό εισάγεται ένα επιπλέον στρώμα Δραστών τύπου Modifier, ανάμεσα στον Dispatcher και στους Learners. Ο Modifier λειτουργεί ως γονέας (parent Actor) του Learner, και μπορεί να τροποποιεί τα κοινά δεδομένα τους. Πιο συγκεκριμένα, έχει πρόσβαση στη θεωρία όπου εξετάζει ο Learner και όπου προσθέτει ή αφαιρεί clauses από αυτή. Επιπλέον, μπορεί να

τροποποιεί τα στατιστικά των clauses τα οποία είναι ήδη ενσωματωμένα στη θεωρία ώστε ο Learner να ενημερώνεται πλαγίως για τα scores των αντιγράφων τους.



Σχήμα 3.2 – Αρχιτεκτονική διάφθρωση με Modifiers στο μοντέλο Ένα-προς-Ένα

Οι Learners σε αυτή την εκδοχή, αποστέλλουν μηνύματα μονάχα στον Modifier με σκοπό να τον ενημερώσουν για τις όποιες αλλαγές προέκυψαν στα δεδομένα τους και για αιτήματα στατιστικών τα οποία θα προκύψουν κατά την επεξεργασία των παραδειγμάτων τους. Οι αλλαγές οι οποίες προκύπτουν από τη μάθηση αντιγράφων των κανόνων στα υπόλοιπα σετ δεδομένων, μπορούν να γίνουν ορατές μόνο μέσω των κοινών δεδομένων. Οι Modifiers είναι σε θέση να επικοινωνούν μεταξύ τους ώστε να ανταλλάσσουν πληροφορίες. Οι περιπτώσεις επικοινωνίας τροποποιούνται ως εξής:

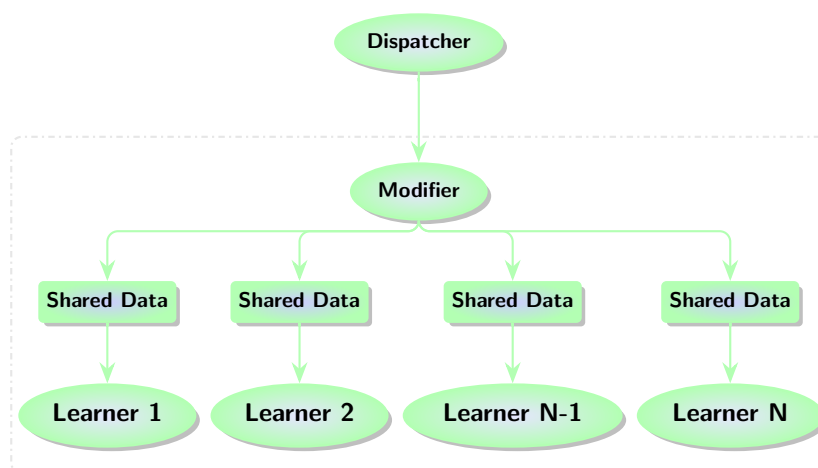
1. **Νέοι κανόνες:** Ο Learner που δημιούργησε έναν καινούριο κανόνα, τον προσθέτει στη θεωρία του και ενημερώνει τον Modifier με τον οποίο χειρίζονται κοινά δεδομένα. Αυτός με τη σειρά του αποστέλλει μηνύματα στους υπόλοιπους Modifiers ενημερώνοντας τους για το νέο κανόνα. Κάθε Modifier γράφει στην κοινή θεωρία που έχει με τον Learner ο οποίος σχετίζεται με αυτόν. Η αναθεωρημένη θεωρία χρησιμοποιείται από τον Learner για την επεξεργασία των υπόλοιπων παραδειγμάτων του.
2. **Clause υποψήφιο για επέκταση:** Μόλις ένας Learner εντοπίσει ένα κανόνα ο οποίος βάσει του κριτηρίου του Hoeffding bound είναι κατάλληλος για εξειδίκευση:
 - i. Ενημερώνει τον Modifier που αντιστοιχεί σε αυτόν πως θέλει τα στατιστικά των αντιγράφων του, και θέτει τον εαυτό του σε ανενεργή κατάσταση (idle state) μέχρις ότου ληφθούν οι απαντήσεις.

- ii. Ακολουθεί η αποστολή μηνυμάτων στους υπόλοιπους Modifiers οι οποίοι εξάγουν τα στατιστικά του αντίστοιχου clause το οποίο ανήκει στη θεωρία που επεξεργάζονται απο κοινού με τους αντίστοιχους Learners και απαντούν στον Modifier.
 - iii. Τελικά ο Modifier μόλις συλλέξει όλες τις απαντήσεις, ανανεώνει τα στατιστικά του clause και ταυτόχρονα ειδοποιεί τον Learner πως μπορεί να προχωρήσει στον τελικό έλεγχο.
3. **Εξειδίκευση ενός clause:** Εφόσον ο παραπάνω έλεγχος είναι θετικός, ο Learner επεκτείνει το clause στην καλύτερη εξειδίκευση του και ενημερώνει τον Modifier. Αυτός με τη σειρά του στέλνει από ένα αντίγραφο του εξειδικευμένου κανόνα στους υπόλοιπους Modifiers ώστε να το τοποθετήσουν στη θεωρία τους αντί της παλαιάς του μορφής.

3.3.5 Μοντέλο Ένα-προς-N

Το μοντέλο Ένα-προς-N αποτέλεσε μια προσπάθεια βελτιστοποίησης του μοντέλου Ένα-προς-Ένα, στο οποίο παρατηρείται μεγάλος αριθμός υπολογιστικών νημάτων, καθώς κάθε μάθηση κανόνων απασχολεί δύο νήματα/σετ δεδομένων αντί του ενός νήματος το οποίο είναι το επιθυμητό. Δεδομένου ότι το πλήθος των νημάτων είναι πεπερασμένο, ακόμα και σε υπολογιστικές μονάδες με αρκετά υψηλό διαθέσιμο αριθμό νημάτων μειώνεται σημαντικά το πλήθος των διεργασιών μάθησης κανόνων.

Στο μοντέλο Ένα-προς-N, εγκαταλείπεται η τακτική του ενός Modifier ανά Learner, τοποθετείται ένα ολικός Modifier ο οποίος διαθέτει κοινά δεδομένα (σε ξεχωριστά όμως πλαίσια για την αποφυγή συγκρούσεων) με όλους τους Learners. Η φιλοσοφία παραμένει η ίδια σε σχέση με το προηγούμενο μοντέλο, όμως αντί για N επιπλέον νήματα (όπου ο αριθμός N αντιστοιχεί στο πλήθος των διαθέσιμων σετ δεδομένων) απαιτείται μόλις ένα επιπλέον νήμα. Επιπλέον, μειώνεται ο αριθμός των μηνυμάτων τα οποία ανταλλάσσονται καθώς αντί της επικοινωνίας μεταξύ των Modifiers, ο μοναδικός πλέον Modifier έχει διαθέσιμη όλη την απαιτούμενη πληροφορία την οποία εγγράφει στο κατάλληλο ή κατάλληλα πλαίσια κάθε φορά, ανάλογα με τους Learners στα οποία προορίζονται.



Σχήμα 3.3 – Αρχιτεκτονική διάρθρωση με έναν Modifier στο μοντέλο Ένα-προς-N

Κεφάλαιο 4

Κατανεμημένη Μάθηση Ορισμών Γεγονότων σε πραγματικό χρόνο

Στο κεφάλαιο αυτό γίνεται παρουσίαση του συστήματος Κατανεμημένης Μάθησης Ορισμών Γεγονότων σε πραγματικό χρόνο. Αρχικά παρουσιάζεται η τακτική διαχωρισμού των δεδομένων (4.1) η οποία επιτρέπει τη χρήση παρόμοιας αρχιτεκτονικής με το μοντέλο Ένα-προς-N, χωρίς την παρέμβαση περισσότερων του ενός Δραστών στα ίδια δεδομένα. Ακολουθεί η ποιοτική περιγραφή του Μοντέλου Ολικού Κλειδώματος (4.2), της πρώτης εκ των δύο υλοποιήσεων που πραγματοποιήθηκαν κατά τη συγγραφή της εργασίας, όπου γίνεται ανάλυση της προσέγγισης η οποία ακολουθήθηκε μαζί με τους τύπους Δραστών οι οποίοι την εξυπηρετούν και προβάλλεται η αρχιτεκτονική διάρθρωση του συστήματος. Η ίδια διαδικασία ακολουθείται για το μοντέλο Ροής Πληροφορίας (4.3), το οποίο αποτελεί τη δεύτερη υλοποίηση του συστήματος Κατανεμημένης Μάθησης Ορισμών Γεγονότων σε πραγματικό χρόνο.

4.1 Διαχωρισμός των παραδειγμάτων σε παρτίδες (batches) δεδομένων

Τα Μοντέλα Χρήσης Κοινών Πληροφορίας(3.3) αποτέλεσαν την πρώτη ολοκληρωμένη υλοποίηση ενός κατανεμημένου(3.3.4) και παράλληλου(3.3.5) συστήματος σύμφωνα με τις προδιαγραφές που ορίστηκαν στην παράγραφο 3.1. Η χρήση τους όμως αποδείχτηκε μη πρακτική, καθώς το γεγονός πώς περισσότερα του ενός νήματα είχαν δικαίωμα εγγραφής στα ίδια δεδομένα περιέπλεξε σε μεγάλο βαθμό το σχεδιασμό τους. Επιπλέον, το μοντέλο Ένα-προς-N (ως μοντέλο παράλληλης

επεξεργασίας) περιοριζόταν από το γεγονός πως μπορούσε να εκτελεστεί σε έναν μόνο υπολογιστή, καθώς σε αντίθετη περίπτωση τα κοινά δεδομένα δε θα μπορούσαν να γίνουν ορατά στην πράξη από απομακρυσμένους υπολογιστές.

Πληροφορίας των παραπάνω, κρίθηκε σκόπιμο να προχωρήσουμε σε μια πιο ολοκληρωμένη αρχιτεκτονική, η οποία να διατηρεί την ελευθερία που παρέχει το μοντέλο αυτό (μικρός αριθμός νημάτων σε σύγκριση με το πλήθος των σετ δεδομένων, έλεγχος ροής από μικρό αριθμό Δραστών) σε ένα βελτιωμένο μοντέλο το οποίο δε θα συνοδεύεται από τους περιορισμούς που θέτει ο αλγόριθμος του Learner ως έχει.

Παρουσιάζουμε το σύστημα Κατανεμημένης Ακολουθιακής Αναγνώρισης Σύνθετων Γεγονότων (Distributed Online Learning of Event Definitions; DOLED), το οποίο προσαρμόζει την αρχιτεκτονική του Ένα-προς-N (ένας Δράστης ο οποίος βρίσκεται στο ενδιάμεσο επίπεδο ανάμεσα Dispatcher και τους Learners) χωρίς η διάρθρωση αυτή να συνεπάγεται την επεξεργασία κοινών δεδομένων από περισσότερους τους ενός Δραστών. Για κάθε διαθέσιμο σετ δεδομένων το οποίο αναμένεται να τεθεί σε επεξεργασία από έναν Learner, ορίζονται υπό-παρτίδες δεδομένων (batches) οι οποίες περιλαμβάνουν ένα προκαθορισμένο πλήθος παραδειγμάτων τα οποία θα επεξεργαστεί ο Learner προτού παύσει προσωρινά τη λειτουργία του. Με αυτό τον τρόπο, παρέχεται η δυνατότητα ενημέρωσης των Learners για τις πρόσθετες πληροφορίες που προκύπτουν μέσω αποστολής μηνυμάτων σε αυτούς τα οποία παρεμβάλλονται ανάμεσα στο χειρισμό των batches.

4.2 Μοντέλο Ολικού Κλειδώματος

4.2.1 Προσέγγιση Ολικού Κλειδώματος

Προσέγγιση Ολικού Κλειδώματος

Η Προσέγγιση Ολικού Κλειδώματος (Global-Lock Approach) θέτει ως προτεραιότητα για όλους τους Learners την έγκαιρη πληροφόρησή τους για την επέκταση ή μη ενός κανόνα.

Σύμφωνα με την παραπάνω δήλωση, κρίνεται απαραίτητη η παύση του συστήματος κάθε φορά που εξετάζεται μία πιθανή επέκταση κάποιου κανόνα, ώστε να μην πραγματοποιείται επεξεργασία παραδειγμάτων μέχρι τη λήψη απόφασης σχετικά με την επέκτασή του. Πρέπει συνεπώς να υπάρχει ο αντίστοιχος αλγοριθμικός σχεδιασμός, ώστε να το σύστημα να μην εξετάζει παραδείγματα σε κανένα υπολογιστικό νήμα κατά το χρονικό διάστημα ανάμεσα στον εντοπισμό των κανόνων αυτών έως τη λήψη της απόφασης.

4.2.2 Ο Δράστης τύπου Συντονιστή

Η αρχιτεκτονική του συστήματος DOLED υιοθετεί ένα νέο είδος Δραστών, το οποίο ονομάζεται Συντονιστής (Coordinator) και σε πλήρη αναλογία με τον Modifier τοποθετείται ανάμεσα στον Dispatcher και τους Learners. Ο Coordinator διατηρεί δομές οι οποίες περιέχουν χρήσιμη πληροφορία για τον κάθε Learner ξεχωριστά. Οι δομές αυτές λειτουργούν ως λεξικά dictionaries¹ με κλειδί τον Learner στον οποίο αναφέρονται οι δομές αυτές και τιμές οι οποίες να ανταποκρίνονται στην εκάστοτε πληροφορία.

Συγκεκριμένα, οι δομές αυτές είναι:

- **clausesToAdd(learner)**: Κανόνες (είτε στη μορφή bottom clauses είτε στη μορφή εξειδικευμένων κανόνων) οι οποίοι οφείλουν να προστεθούν στη θεωρία του Learner.
- **clausesToRemove(learner)**: Αναγνωριστικά (id's) των κανόνων οι οποίοι πρέπει να αφαιρεθούν από τη θεωρία του Learner.

Η ειδοποιός διαφορά σε σχέση με τον Modifier είναι πως οι δομές αυτές δεν μπορούν να γίνουν ορατές από τους ίδιους τους Learners, καθώς δεν έχουν πρόσβαση σε αυτές. Αντιθέτως, μέσω κατάλληλων μηνυμάτων τα οποία θα αναλυθούν σε επόμενη παράγραφο, ενημερώνουν τους Learners ανάμεσα στον χειρισμό των batches ώστε να μπορούν να χρησιμοποιήσουν αυτές τις πληροφορίες.

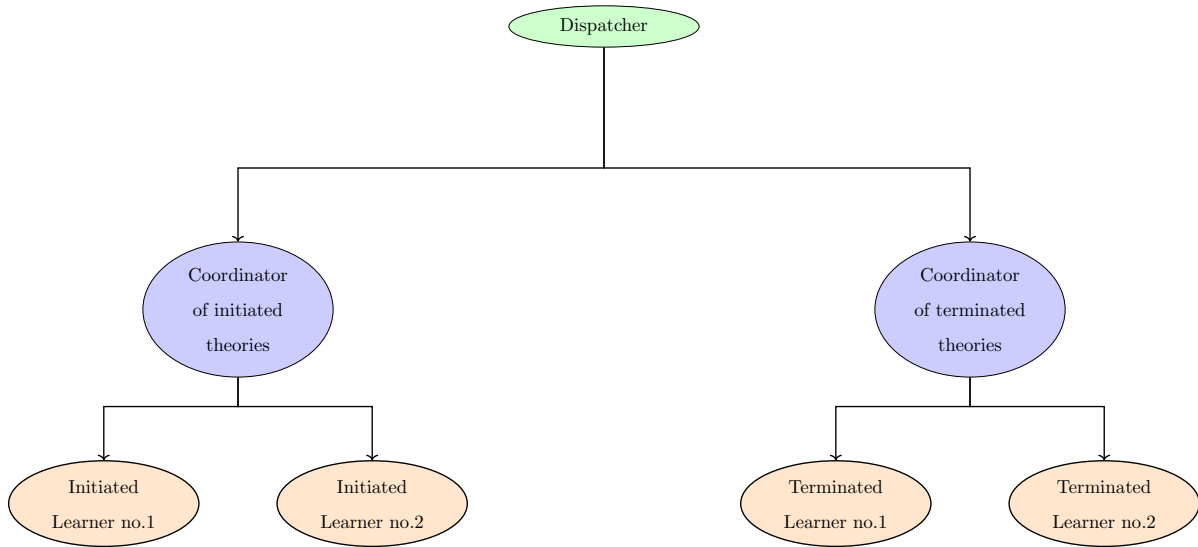
4.2.3 Αρχιτεκτονική διάρθρωση Δραστών στο Μοντέλο Ολικού Κλειδώματος

Για κάθε σύστημα στο μοντέλο ενεργοποιούνται ακριβώς δύο Coordinators, ένας για μάθηση initiated κανόνων και αντιστοίχως ένας για μάθηση terminated κανόνων, οι οποίοι τοποθετούνται κατά ένα επίπεδο υψηλότερα από τους Learners οι οποίοι μαθαίνουν κανόνες της αντίστοιχης κλάσης. Σε πλήρη αντιστοιχία με το μονολιθικό σύστημα του OLED, οι διεργασίες αυτές είναι ανεξάρτητες, και κάθε Coordinator επικοινωνεί αποκλειστικά με τους Learners και όχι με τον έταίρο Coordinator του συστήματος. Αντίστοιχα, οι Learners μπορούν να στείλουν μηνύματα μόνο στον Coordinator ο οποίος είναι συσχετισμένος μαζί τους.

Στο ακόλουθο σχήμα βλέπουμε τη διάρθρωση ενός συστήματος DOLED σχεδιασμένο

¹<https://xlinux.nist.gov/dads/HTML/dictionary.html>

στο Μοντέλο Ολικού Κλειδώματος το οποίο χειρίζεται δύο ανεξάρτητα σετ δεδομένων για να μάθει κανόνες.



Σχήμα 4.1 – Διάρθρωση της ιεραρχίας των Δραστών των Δραστών Μοντέλο Ολικού Κλειδώματος. Οι Initiated Learner no.1 και Terminated Learner no.1 (και ομοίως κάθε ζεύγος Initiated Learner no.i, Terminated Learner no.j όπου $i = j$) χειρίζονται ακριβώς τα ίδια δεδομένα διατυπωμένα με την ίδια σειρά, μαθαίνοντας όμως διαφορετική κατηγορία κανόνων (initiated και terminated κανόνες αντίστοιχα)

4.2.4 Τύποι μηνυμάτων

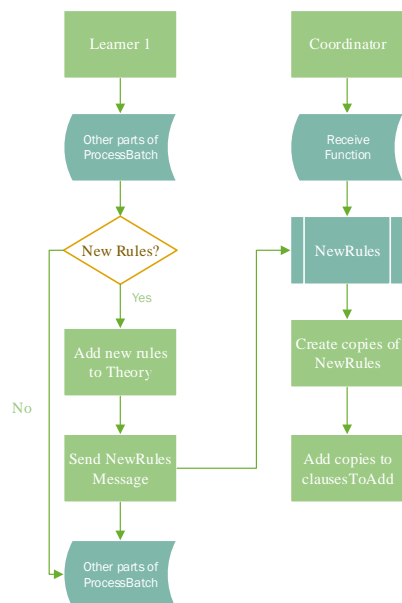
Είναι σημαντικό να παρατηρήσουμε πως στο σύστημα DOLED, τόσο οι δύο Coordinators όσο και οι Learners οι οποίοι δρουν στο σύστημα αποστέλλουν και λαμβάνουν μηνύματα, για τα οποία έχει προβλεφθεί η συμπεριφορά τους σύμφωνα με τον τύπο των εκάστοτε μηνυμάτων. Επομένως, απαιτείται κατάλληλος ορισμός των συναρτήσεων Receive και για τους 2 τύπους Δραστών που χρησιμοποιούνται κατά κόρον στη σχεδίαση αυτή.

Παρακάτω διακρίνουμε τα μηνύματα σε κατηγορίες, εξηγούμε ποιοτικά τη σημασία τους και παρουσιάζουμε τον τρόπο χειρισμού τους από τις δύο κατηγορίες (Coordinator/Learner).

4.2.4.1 New Rules

Η ποιοτική σημασία του μηνύματος **NewRules** είναι ταυτόσημη με αντίστοιχα μηνύματα που παρουσιάσαμε στις προηγούμενες υλοποιήσεις. Αφορούν κανόνες κενούς ως προς το σώμα (bottom rules) οι οποίοι συνοδεύονται από εξειδικεύσεις (refinements) προς αξιολόγηση.

Ένα μήνυμα NewRules περιέχει μια λίστα με αντίγραφα ενός ή περισσότερων bottom rules που δημιουργήθηκαν από έναν Learner κατά την επεξεργασία παραδειγμάτων του σετ δεδομένων του, και αποστέλλεται στον Coordinator. Με την παραλαβή ενός μηνύματος NewRules ο Coordinator προσθέτει τη λίστα αυτή στη δομές clausesToAdd όλων των Learners, εκτός από τη δομή η οποία έχει ως κλειδί τον αποστολέα.



Σχήμα 4.2 – Δημιουργία και χειρισμός μηνύματος NewRules στο Μοντέλο Ολικού Κλειδώματος

4.2.4.2 StatsRequested - Clauses Wanted

Στην περίπτωση που το κριτήριο του Hoeffding bound ικανοποιείται τοπικά για έναν κανόνα (στο σετ δεδομένων ενός από τους Learners), πρέπει να κινητοποιηθεί η διαδικασία ενημέρωσης με όλη τη διαθέσιμη πληροφορία ώστε να ληφθεί απόφαση όσον αφορά την εξειδίκευση του κανόνα αυτού.

Στην περίπτωση αυτή ενημερώνεται ο Coordinator με τη δήλωση του κανόνα, ο οποίος με τη σειρά του αποστέλλει μήνυμα ClausesWanted (το οποίο περιέχει το id του κανόνα αυτού) στους υπόλοιπους Learners ζητώντας τα στατιστικά του κάθε αντιγράφου που ανήκει στη θεωρία τους και αξιολογείται στα σετ δεδομένων τους.

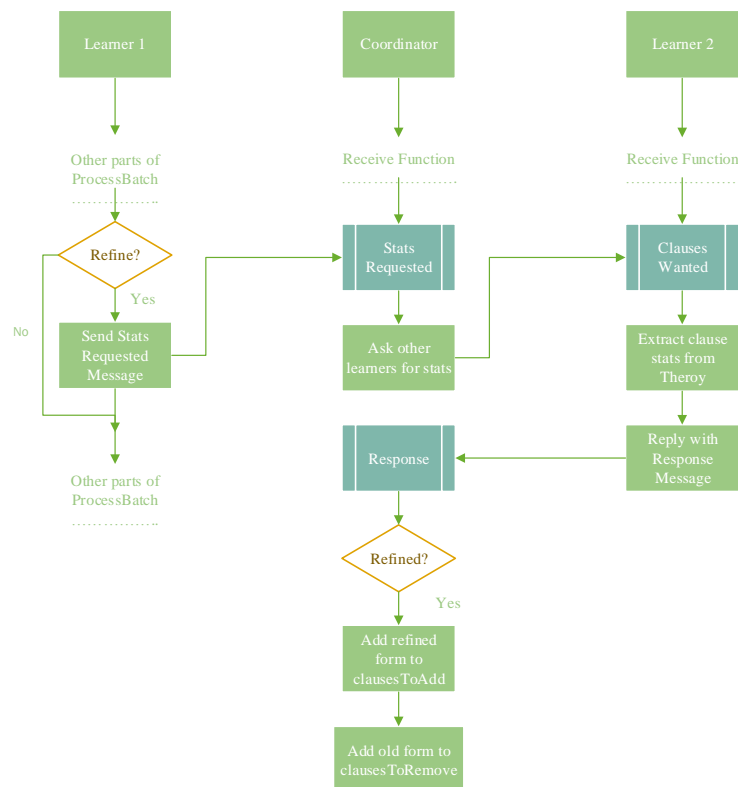
4.2.4.3 Clauses Wanted -Response

Η διαδικασία απόφασης συνίσταται σε:

- Ανανέωση των δεδομένων του κανόνα με ενσωμάτωση των scores των αντιγράφων από όλα τα επιμέρους σετ δεδομένων.
- Επαναπροσδιορισμό της τιμής Hoeffding bound (καθώς εξαρτάται από το πλήθος των παραδειγμάτων στα οποία έχει αξιολογηθεί ο κανόνας).
- Ανανέωση των scores τόσο του κανόνα όσο και των εξειδικεύσεων του με βάση τα στατιστικά που έγιναν διαθέσιμα από την αξιολόγηση τους στα υπόλοιπα σετ δεδομένων.
- Επανάληψη του ελέγχου για κατά πόσο ικανοποιείται το κριτήριο και με τα ανανεωμένα δεδομένα.
- Εξειδίκευση του κανόνα σε περίπτωση θετικού ελέγχου.

Οι Learners λαμβάνουν μήνυμα τύπου ClausesWanted με το οποίο ζητούνται τα στατιστικά του clause που αντιστοιχεί στο id που περιέχεται στο μήνυμα αυτό. Μόλις το εντοπίσουν ανταποκρίνονται με μήνυμα Response το οποίο περιέχει τα εν λόγω στατιστικά. Εφόσον ληφθούν όλες οι απαντήσεις, λαμβάνεται η απόφαση για την εξειδίκευση του clause. Στην περίπτωση αυτή:

- Ανανεώνονται οι δομές clausesToAdd όλων των Learners με τη δήλωση του κανόνα που αποτελεί την καλύτερη εξειδίκευση.
- Προστίθεται σε όλες τις δομές clausesToRemove το id της προηγούμενης μορφής για να αφαιρεθεί μελλοντικά από τις θεωρίες όλων των Learners



Σχήμα 4.3 – Διαδικασία εξειδίκευσης ενός κανόνα στο Μοντέλο Ολικού Κλειδώματος

4.2.4.4 Drop Clause

Επιθυμώντας να διατηρήσουμε την ακολουθιακή φιλοσοφία (single pass) του μονολιθικού συστήματος, προβλέπεται η αποκοπή κανόνων από τη θεωρία των οποίων το score παραμένει χαμηλό μετά από την εξέταση ενός σημαντικά μεγάλου αριθμού παραδειγμάτων. Πρέπει επομένως να εξασφαλίσουμε πως ένας κανόνας, εφόσον αποκοπεί από μία θεωρία, θα αποκοπεί και από τις υπόλοιπες υπάρχουσες θεωρίες του συστήματος.

Αποστέλλεται επομένως μήνυμα τύπου DropClause από τον Learner ο οποίος προχώρησε στην αποκοπή ενός κανόνα προς τον Coordinator. Το μήνυμα αυτό περιέχει το id του αποκομμένου κανόνα, και αφού πραγματοποιηθούν κάποιο έλεγχοι(αν ο κανόνας βρίσκεται υπό κρίση για την επέκταση του σύμφωνα με πληροφορίες οι οποίες έχουν παραληφθεί από κάποιον άλλο Learner), το id προστίθεται στις δομές clausesToRemove των υπολοίπων Learners.

4.2.4.5 Awaiting - Proceed

Το ζεύγος μηνυμάτων Awaiting - Proceed είναι απαραίτητο για τη μετάβαση σε ένα μοντέλο όπως αυτό του DOLED το οποίο χειρίζεται batches από παραδείγματα στη μάθηση των κανόνων. Επιτρέπουν την ομαλή μεταφορά πληροφοριών στους Learners μετά από την επεξεργασία κάθε batch και εξασφαλίζουν την αλγοριθμική συνέπεια του κατανεμημένου μοντέλου ως προς το μονολιθικό.

- Το μήνυμα **Awaiting** είναι κενό ως προς το περιεχόμενο του, και είναι ο τρόπος με τον οποίο ενημερώνει ένας Learner τον Coordinator πως ολοκλήρωσε το batch το οποίο επεξεργαζόταν και πως αναμένει άδεια ώστε να προχωρήσει στο επόμενο batch. Οποιαδήποτε πληροφορία έχει προκύψει κατά την εν λόγω επεξεργασία έχει ήδη μεταβιβαστεί στον Coordinator μέσω του αντίστοιχου τύπου μηνυμάτων.
- Το μήνυμα **Proceed** αποστέλλεται από τον Coordinator σε κάθε Learner ως απάντηση σε ένα μήνυμα Awaiting, υπό την προϋπόθεση ότι δεν έχει εκκινήσει η διαδικασία για την επέκταση ή μη κάποιου clause. Σε αντίθετη περίπτωση, πρώτα ολοκληρώνεται η διαδικασία εξειδίκευσης και εν συνεχεία αποστέλλονται μαζικά μηνύματα τύπου Proceed προς όλους τους Learners. Ένα μήνυμα Proceed μπορεί να περιέχει μία λίστα με κανόνες (clausesToAdd) τους οποίους ο Learner θα ενσωματώσει στη δική του θεωρία όπως επίσης μία λίστα με αναγνωριστικά id's των κανόνων (clausesToRemove) που θα αφαιρεθούν από τη θεωρία αυτή. Οι λίστες αυτές αντιστοιχούν στις δομές clausesToAdd και clausesToRemove τις οποίες διατηρεί ο Coordinator με κλειδί τον Learner.

4.2.4.6 Συγκεντρωτικός πίνακας μηνυμάτων

Ο ακόλουθος πίνακας περιέχει μία σύνοψη των των μηνυμάτων και των τρόπων χειρισμού τους.

Τύπος Μηνύματος	Learner		Coordinator	
	Αποστολή	Παραλαβή	Αποστολή	Παραλαβή
NewRules	Αποστέλλεται στον Coordinator όταν κατά τη μάθηση κανόνων δημιουργούνται νέοι κανόνες			Προσθέτει αντίγραφα των κανόνων στη δομή clausesToAdd κάθε Learner πλην του αποστολέα
StatsRequested	Αποστέλλεται στον Coordinator όταν ένας κανόνας ικανοποιεί το κριτήριο του Hoeffding bound ζητώνταςτα στατιστικά των αντιγράφων του			Στέλνει μήνυμα ClausesWanted σε όλους τους Learners πλην του αποστολέα
ClausesWanted		Ανταποκρίνεται μέσω μηνύματος Response το οποίο περιέχει τα στατιστικά του κανόνα ο οποίος ζητήθηκε	Αποστέλλεται στους Learners μετά τη λήψη μηνύματος StatsRequested	
Response	Αποστολή μηνύματος Response στον Coordinator			Γίνεται χρήση των περιλαμβανόμενων στατιστικών για την απόφαση ανανέωσης ενός κανόνα. Σε περίπτωση ανανέωσης προστίθεται ο εξειδικευμένος κανόνας στις δομές clausesToAdd και προστίθεται η προηγούμενη δήλωση του στις δομές clausesToRemove όλων των Learners
DropClause	Αποστέλλεται στον Coordinator όταν ένας κανόνας διατηρεί χαμηλό score χωρίς να επεκτείνεται για μεγάλο διάστημα οπότε και αφαιρείται από τη θεωρία			Πρόσθεση του κανόνα στις δομές clausesToRemove όλων των Learners
Awaiting	Αποστέλλεται στον Coordinator μετά την επεξεργασία κάθε batch μετά την επεξεργασία κάθε batch			Αποστολή μηνύματος Proceed εφόσον δεν υπάρχει κανόνας υποψήφιος προς επέκταση
Proceed		Προσθήκη και αφαίρεση κανόνων από τη θεωρία κατά τις οδηγίες και χειρισμός επόμενου batch	Αποστολή μετά από λήψη μηνύματος Awaiting ή μετά την απόφαση σχετικά με την επέκταση κανόνων	

Πίνακας 4.1 – Τύπος μηνυμάτων για το σύστημα Κατανεμημένης Ακολουθιακής Αναγνώρισης Σύνθετων Γεγονότων σε συνδυασμό με τον τρόπο χειρισμού τους από τους δράστες τύπου Learner και Coordinator

4.2.5 Περιγραφή της λειτουργίας του Μοντέλου Ολικού Κλειδώματος

Η διαδικασία με την οποία το σύστημα DOLED μαθαίνει κανόνες στηρίζεται στον έλεγχο της ροής των δεδομένων από τον Coordinator. Η μάθηση κανόνων από τον Learner εκτελείται με τρόπο παρόμοιο του μονολιθικού συστήματος, με τη διαφορά ότι τα παραδείγματα δίνονται σε batches αντί να παρέχονται όλα μαζί, ώστε να υπάρχει η δυνατότητα ανταπόκρισης στα μηνύματα που λαμβάνονται στο νεκρό χρόνο μεταξύ των παραδειγμάτων. Ο Learner ενημερώνει τον Coordinator με μηνύματα κατά τη διάρκεια της επεξεργασίας των batches (λογική η οποία ακολουθήθηκε και στα μοντέλα χρήσης κοινών δεδομένων) χωρίς όμως να θέτει τον εαυτό του σε αναμονή όταν περιμένει πληροφορίες. Η απαραίτητη παύση του συστήματος επιτυγχάνεται με έναν απλό έλεγχο από τη μεριά του Coordinator. Εφόσον υπάρχουν ένας ή περισσότεροι κανόνες για τους οποίους αναμένονται πληροφορίες, ο Coordinator δεν στέλνει μήνυμα τύπου Proceed μετά την παραλαβή μηνύματος Awaiting από τους Learners. Με αυτό τον τρόπο οι Learners παραμένουν ανενεργοί (idle state) για ένα διάστημα, καθώς δεν επεξεργάζονται κάποιο batch, και ανταποκρίνονται μόνο στα μηνύματα ClausesWanted που θα λάβουν ώστε να ενημερώσουν τον Coordinator με τα στατιστικά των κανόνων οι οποίοι υπόκεινται σε αξιολόγηση. Μόλις ληφθεί η απόφαση για κάθε κανόνα μεμονωμένα και ενημερωθούν κατάλληλα όλες οι δομές clausesToAdd και clausesToRemove, ο Coordinator στέλνει μαζικά μηνύματα Proceed και οι Learners ανανεώνουν τις θεωρίες τους σύμφωνα με τις οδηγίες και προχωρούν στο επόμενο batch.

4.2.6 Αλγοριθμική παρουσίαση του Μοντέλου Ολικού Κλειδώματος

Συνοψίζοντας τις πληροφορίες που παρουσιάσαμε στο κεφάλαιο, παρουσιάζουμε τους αλγορίθμους για τους Δράστες τύπου Learner και Coordinator. Ο πρώτος αλγόριθμος(;;) αφορά το τμήμα της επεξεργασίας παραδειγμάτων στο σύστημα DOLED, ενώ ο επόμενος αλγόριθμος(;;) παρουσιάζει μαζί τις συναρτήσεις receive των δύο τύπων Δραστών, ώστε να είναι πιο εύληπτη η διαφορά στη συμπεριφορά τους για κοινούς τύπους μηνυμάτων.

Algorithm 2 Learner's Process Batch function in the DOLED

system($batchI, H, B, G, \delta, d, N_{min}, S_{min}$)

Input: $batchI$: A batch of the Learner's stream of training interpretations; H : An empty or incomplete theory; B : Background knowledge; G : Clause evaluation function; δ : Confidence for the Hoeffding test; d : Specialization depth; S_{min} : Clause G -score quality threshold.

```

1: function ProcessBatch( $batchI, H, B, G, \delta, d, N_{min}, S_{min}$ ):
2:   for each  $I \in \mathcal{I}$  do
3:     Update  $TP_r, FP_r, FN_r$  and  $N_r$  counts from  $I$ , for each  $r \in H$  and each  $r' \in \rho_d(r)$ ,
       where  $N_r$  denotes the number of examples on which  $r$  has been evaluated so far.
4:     if ExpandTheory( $B, H, I$ ) then
5:        $H \leftarrow H \cup \text{StartNewClause}(B, I)$ 
6:     else
7:       for each clause  $r \in H$  do
8:          $r \leftarrow \text{ExpandClause}(r, G, \delta)$ 
9:        $H \leftarrow \text{Prune}(H, S_{min})$ 
10:  return  $H$ 
11: function StartNewClause( $B, I$ ):
12:  Generate a bottom clause  $\perp$  from  $I$  and  $B$ 
13:   $r := \text{head}(\perp) \leftarrow$ 
14:   $\perp_r := \perp$ 
15:   $N_r = FP_r = TP_r = FN_r := 0$ 
16:  Send message  $\text{NewRules}(r)$  to Coordinator
17:  return  $r$ 
18: function ExpandClause( $r, G, \delta$ ):
19:  Compute  $\epsilon = \sqrt{\frac{\ln(1/\delta)}{2N_r}}$  and let  $\bar{G}$  denote the mean value of a clause's  $G$ -score
20:  Let  $r_1$  be the best specialization of  $r$ ,  $r_2$  the second best and  $\Delta\bar{G} = \bar{G}(r_1) - \bar{G}(r_2)$ 
21:  Let  $\tau$  equal the mean value of  $\epsilon$  observed so far
22:  if  $\bar{G}(r_1) > \bar{G}(r)$  and  $[\Delta\bar{G} > \epsilon \text{ or } \tau < \epsilon]$ :
23:    Send message  $\text{StatsRequested}(r.id)$  to Coordinator
24: function prune( $H, S_{min}$ ):
25:  Remove from  $H$  each clause  $r$  for which  $S_{min} - \bar{G}(r) > \epsilon$ , where  $\epsilon$  is the current Hoeffding bound
26:  Send message  $\text{DropClause}(r.id)$  to Coordinator
27:  return  $H$ 

```

(a) Learner's receive function in DOLED	(b) Coordinator's receive function in DOLED
1: function Receive(<i>MessageType</i>):	1: function Receive(<i>MessageType</i>):
2: Case <i>p</i> : <i>Proceed</i> :	2: Case <i>a</i> : <i>Awaiting</i> :
3: $H \leftarrow H \cup p.clausesToAdd$	3: If (! <i>lock</i>):
4: $H \leftarrow H - p.clausesToRemove$	4: $add \leftarrow clausesToAdd(sender)$
5: Get <i>newBatch</i> from <i>I</i>	5: $remove \leftarrow clausesToRemove(sender)$
6: ProcessBatch(<i>batchI</i> , <i>H</i> , <i>B</i> , <i>G</i> , δ , <i>d</i> , N_{min} , S_{min})	6: Send message <i>Proceed</i> (<i>add</i> , <i>remove</i>) to sender
7: Case <i>c</i> : <i>ClausesWanted</i> :	7: Case <i>s</i> : <i>StatsRequested</i> :
8: For each $r \in H$:	8: $lock := true$ //controls the flow
9: If ($r.id == c.id$):	9: //There can be more than one
10: Respond with stats <i>tps</i> , <i>fps</i> , <i>fns</i>	10: //up for expansion
11: and , <i>examplesNumber</i> of <i>r</i>	11: $pendingClauses := pendingClauses + 1$
12: and of all $r' \in \rho_d(r)$	12: For each <i>learner</i> !sender:
13: via a <i>Response</i> (<i>stats</i>) message	13: Send message <i>ClausesWanted</i> (<i>s.id</i>) to learner
	14: Case <i>r</i> : <i>Response</i> :
	15: If all learners have responded for <i>r.id</i> :
	16: Compute $\epsilon = \sqrt{\frac{\ln(1/\delta)}{2N_r}}$ and let \bar{G}
	17: denote the mean value of a clause's <i>G</i> -score
	18: Let r_1 be the best specialization of <i>r</i> ,
	19: r_2 the second best and $\Delta\bar{G} = \bar{G}(r_1) - \bar{G}(r_2)$
	20: Let τ equal the mean value of ϵ observed so far
	21: if $\bar{G}(r_1) > \bar{G}(r)$ and $[\Delta\bar{G} > \epsilon$ or $\tau < \epsilon]$:
	22: $\perp_{r_1} := \perp_r$
	23: For all <i>learners</i> :
	24: Add r_1 to <i>clausesToAdd</i> (<i>learner</i>)
	25: Add <i>r.id</i> to <i>clausesToRemove</i> (<i>learner</i>)
	26: $pendingClauses := pendingClauses - 1$
	27: If ($pendingClauses = 0$):
	28: $lock := false$ //controls the flow
	29: For each <i>learner</i> :
	30: $add \leftarrow clausesToAdd(sender)$
	31: $remove \leftarrow clausesToRemove(sender)$
	32: Send message <i>Proceed</i> (<i>add</i> , <i>remove</i>) to sender
	33: Case <i>d</i> : <i>DropClause</i> :
	34: Add <i>d.id</i> to <i>clausesToRemove</i> (<i>sender</i>)

Πίνακας 4.2 – Συναρτήσεις receive των Learner και Coordinator

4.3 Μοντέλο Ροής Πληροφορίας

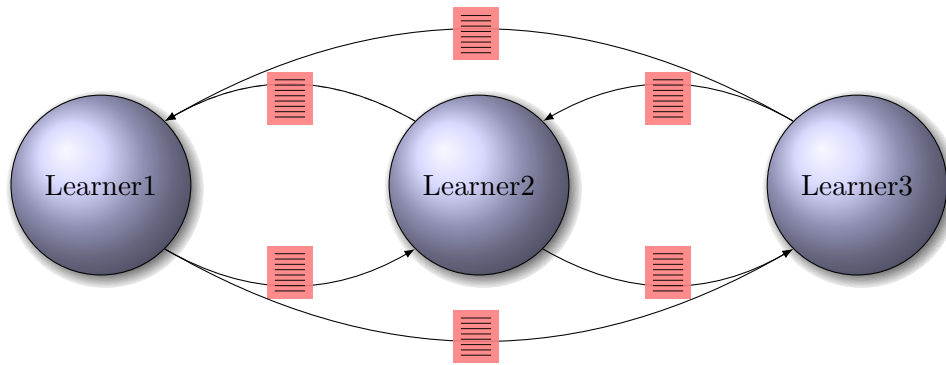
4.3.1 Χρονικοί περιορισμοί στο ΜΟΚ

Στόχο της σχεδίασης του Μοντέλου Ροής Πληροφορίας αποτέλεσε η σύνθεση ενός συστήματος παρόμοιου του ΜΟΚ το οποίο θα μπορεί να μαθαίνει κανόνες παρόμοιας ποιότητας σε μικρότερο χρονικό διάστημα εκτέλεσης. Ο βασικός παράγοντας ο οποίος καθορίζει τη χρονική διάρκεια του ΜΟΚ (πέραν σαφώς από το πλήθος των παραδειγμάτων που εμπεριέχει το κάθε διακριτό σετ δεδομένων) είναι τα νεκρά διαστήματα που δημιουργούνται κατά τη διαδικασία εξειδίκευσης των clauses. Όπως περιγράψαμε και στην παράγραφο 4.2.5, και σύμφωνα με την Προσέγγιση Ολικού Κλειδώματος, όταν εμφανίζεται ένας κανόνας ο οποίος να ικανοποιεί το κριτήριο του Hoeffding bound, το σύστημα παύει για ένα χρονικό διάστημα την επεξεργασία παραδειγμάτων από τους Learners ώστε να αποφασίσει για την επέκταση του κανόνα, και μόνο αφού ολοκληρωθεί η διαδικασία το αποτέλεσμα γίνει διαθέσιμο σε όλους τους Learners μπορεί να συνεχιστεί η αξιολόγηση των υπόλοιπων κανόνων στα παραδείγματα των σετ δεδομένων.

4.3.2 Ελαττωμένη αρχιτεκτονική Δραστών

Για το λόγο αυτό, προσαρμόζουμε την αλγοριθμική υλοποίηση με τέτοιο τρόπο (όπως θα φανεί αργότερα και στην παραγράφο και 4.3), ώστε να μην απαιτείται η προσθήκη ενός επιπέδου Δραστών ανάμεσα στον Dispatcher και στους Learners. Πλέον κάθε Learner έχει τη δυνατότητα να επικοινωνήσει με τους υπόλοιπους και να ανταλλάσσει δεδομένα μαζί τους, όπως συνέβαινε στα μοντέλα Χρήσης Κοινών Πληροφορίας(3). Με τον χωρισμό των παραδειγμάτων σε batches η διαδικασία αυτή είναι εφικτή και δεν συνοδεύεται από τους περιορισμούς οι οποίοι είχαν προκύψει κατά τη σχεδίαση και υλοποίηση των μοντέλων αυτών.

Το ακόλουθο σχήμα δείχνει την επικοινωνία μεταξύ των Learners οι οποίοι λειτουργούν σε ένα σύστημα το οποίο χειρίζεται τρία διακριτά σετ δεδομένων.



Σχήμα 4.4 – Επικοινωνία μεταξύ Learners Μοντέλο Ροής Πληροφορίας. Κάθε Learner έχει τη δυνατότητα να επικοινωνεί άμεσα με τους υπόλοιπους Learners του συστήματος και να αποστέλλει μηνύματα σε αυτούς.

4.3.3 Τύποι μηνυμάτων

4.3.3.1 New Rules - Clauses Wanted - Response

Σε απόλυτη συνάφεια με το σύστημα MOK, τα μηνύματα τύπου NewRules, Clauses Wanted και Response περιλαμβάνουν ακριβώς τις ίδιες πληροφορίες σε σύγκριση με τους αντίστοιχους τύπους μηνυμάτων στο σύστημα αυτό. Η διαφορά σε σχέση με τον MOK είναι πως τα μηνύματα αυτά αποστέλλονται απευθείας στους Learners δεν παρεμβάλλεται ο Coordinator για το χειρισμό τους. Κάθε πληροφορία ενσωματώνεται ή αποκόπτεται από τη θεωρία των Learners αμέσως μετά την παραλαβή των μηνυμάτων.

Μία ειδική κατάσταση η οποία μπορεί να προκύψει σε ένα σύστημα Ροής Πληροφορίας, είναι να ζητηθούν από έναν Learner τα στατιστικά ενός κανόνα ο οποίος δεν υπάγεται στη θεωρία του. Το σενάριο αυτό είναι εφικτό στο σύστημα, καθώς δεν υπάρχουν καθυστερήσεις οι οποίες θα εξασφαλίζουν έναν σχετικό συγχρονισμό των επιμέρους νημάτων, με αποτέλεσμα να μην μπορεί να ελεγχθεί κατά πόσο ζητείται ένας κανόνας ο οποίος πιθανόν να μην έχει παραδοθεί.

Ο τρόπος αντιμετώπισης αυτής της κατάστασης περιλαμβάνει την επαναποστολή του μηνύματος ClausesWanted από έναν Learner στον εαυτό του κάθε φορά που δεν εντοπίζει τον κανόνα στη θεωρία του. Το μήνυμα τύπου ClausesWanted τροποποιείται κατάλληλα ώστε να περικλείει και έναν τρόπο πρόσβασης² στον Δράστη ο οποίος αρχικά είχε ζητήσει τα στατιστικά του κανόνα (καθώς στο MOK η απόκριση γινόταν στον αποστολέα, κάτι το οποίο θα οδηγούσε σε ανακύκλωση μηνυμάτων που περιέχουν στατιστικά κανόνων στον ίδιο Δράστη) και να εκτελεσθούν σωστά τα βήματα στη διαδικασία εξειδίκευσης.

²Συγκεκριμένα μια μεταβλητή τύπου ActorRef, ενδεικτικά ο αναγνώστης μπορεί να εξετάσει τον ακόλουθο σύνδεσμο: <http://doc.akka.io/docs/akka/snapshot/scala/general/addressing.html>

4.3.3.2 Refined Rule

Με την αφαίρεση του επιπλέον επιπέδου Δραστών ανάμεσα στον Dispatcher και τους Learners, δεν υφίσταται η δυνατότητα κεντρικής ενημέρωσης για την επέκταση κάποιου κανόνα. Αντιθέτως, ο Learner αποφασίζει τοπικά, και με τη λήψη όλων των απαντήσεων για την εξειδίκευση ενός κανόνα, και ενημερώνει τους υπόλοιπους με ένα καινούριο μήνυμα τύπου RefinedRule, το οποίο περιλαμβάνει τη δήλωση του εξειδικευμένου κανόνα και το αναγνωριστικό (id) της προηγούμενης μορφής του. Κάθε ένας από τους υπόλοιπους Learners αντικαθιστά τον κανόνα που ταυτίζεται με το αναγνωριστικό αυτό με τον καινούριο κανόνα που έλαβε μέσω του μηνύματος αυτού.

4.3.3.3 Proceed

Χωρίς να υπάρχει το ενδιάμεσο επίπεδο δραστών τύπου Coordinator, η διαδικασία με την οποία οι Learners δέχονται την οδηγία για να συνεχίσουν οφείλει να διαφοροποιηθεί. Όπως θα δούμε και στη συνέχεια, οι καινούριες προσεγγίσεις είναι δομημένες με τέτοιο τρόπο ώστε να είναι εφικτός ο έλεγχος ροής του κάθε Learner από τον ίδιο, ώστε να αποφασίζει μόνος του κατά πόσο μπορεί να προχωρήσει στο επόμενο batch.

Ως αποτέλεσμα το μήνυμα Proceed είναι κενό ως προς το περιεχόμενο του και αποστέλλεται αποκλειστικά από κάθε Learner στον εαυτό του.

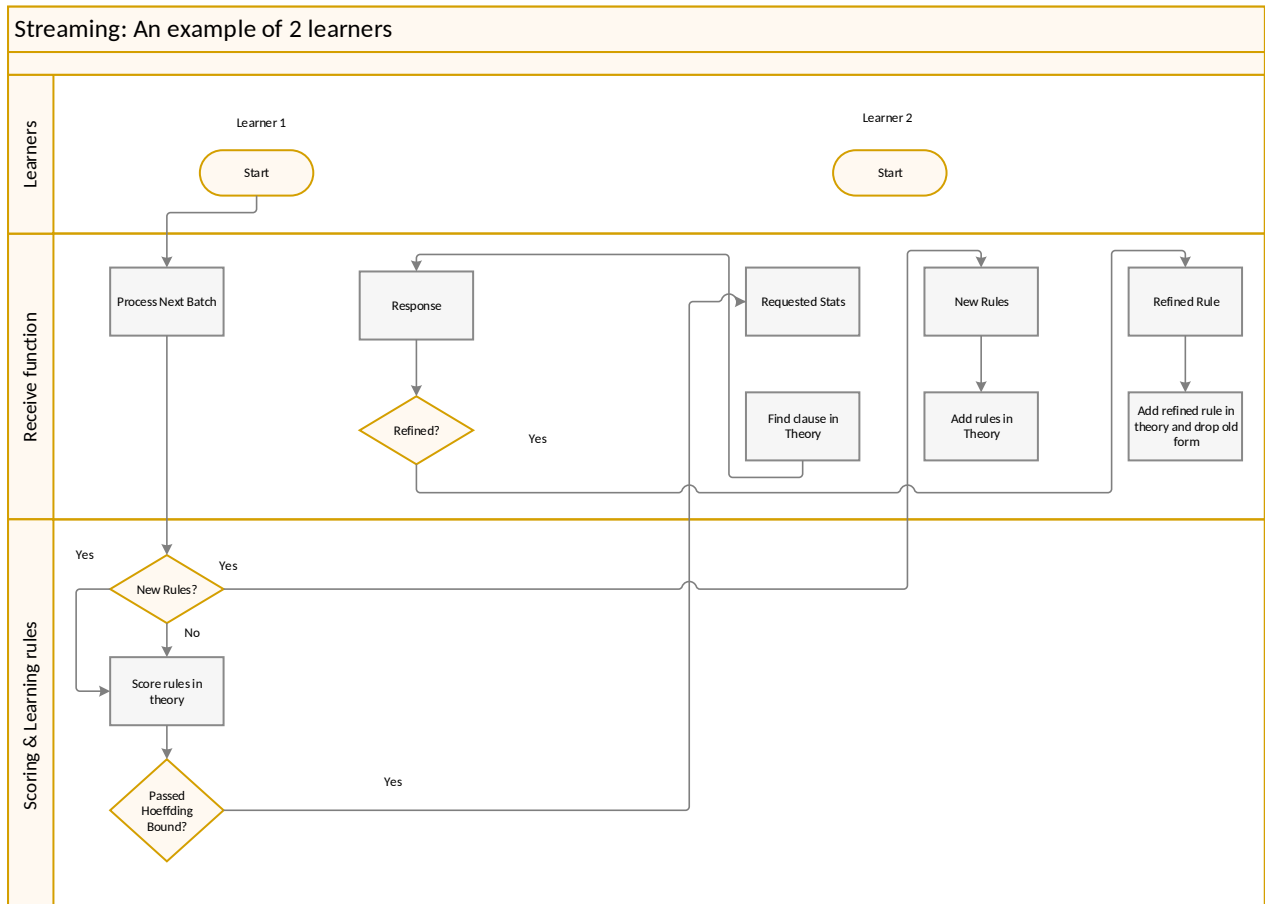
4.3.4 Προσέγγιση Ροής Πληροφορίας

Προσέγγιση Ροής Πληροφορίας

Η Προσέγγιση Ροής Πληροφορίας (Streaming Approach) θέτει ως προτεραιότητα την ολοκλήρωση της μάθησης κανόνων στο μικρότερο εφικτό χρονικό διάστημα.

Θεωρώντας πως σε μεγάλα συστήματα η ροή της πληροφορίας θα είναι συνεχής και αδιάκοπη, θεωρούμε ένα σύστημα στο οποίο δεν πραγματοποιείται κάποια παύση κατά την επεξεργασία των διαθέσιμων παραδειγμάτων. Αυτό πρακτικά σημαίνει πως οι Learners στέλνουν συνεχόμενα μηνύματα τύπου Proceed στον εαυτό τους αμέσως μετά την ολοκλήρωση του χειρισμού κάθε παρτίδας δεδομένων, χωρίς να πραγματοποιούν κάποιο τοπικό έλεγχο για κανόνες οι οποίοι βρίσκονται στη διαδικασία επέκτασής τους. Τα υπόλοιπα μηνύματα παρεμβάλλονται ανάμεσα στα μηνύματα Proceed και οι πληροφορίες που φέρουν εγγράφονται στη θεωρία του παραλήπτη όταν αυτό είναι επιθυμητό.

4.3.5 Αλγοριθμική περιγραφή του Μοντέλου Ροής Πληροφορίας



Σχήμα 4.5 – Δύο Learners επεξεργάζονται παραδείγματα και ανταλλάσσουν μηνύματα μεταξύ τους στο Μοντέλο Ροής Πληροφορίας.

Στο παραπάνω σχήμα μπορούμε να δούμε το διάγραμμα ροής για δύο Learners οι οποίοι δρουν στο μοντέλο Ροής Πληροφορίας. Ο Learner1 επεξεργάζεται τα διαθέσιμα σε αυτόν παραδείγματα. Εφόσον υπάρχει ανάγκη επικοινωνίας (NewRules , StatsRequested) αποστέλλει μηνύματα στον Learner 2, ο οποίος ανταποκρίνεται σε αυτά είτε εισάγοντας κανόνες στη θεωρία του (NewRules) είτε απαντώντας με στατιστικά των κανόνων του (Response). Ο Learner 1 χειρίζεται πιθανά μηνύματα Response ώστε να αξιολογήσει καθολικά έναν κανόνα, και στην περίπτωση εξειδίκευσης του ενημερώνει τον Learner 2 με μήνυμα τύπου RefinedRule. Με αυτό τον τρόπο, ο ανανεωμένος κανόνας αντικαθιστά την προηγούμενη έκφραση του στη θεωρία και των δύο Δραστών.

Algorithm 3 Learner's Process Batch function in the Streaming system($batchI, H, B, G, \delta, d, N_{min}, S_{min}$)

Input: $batchI$: A batch of the Learner's stream of training interpretations; H : An empty or incomplete theory; B : Background knowledge; G : Clause evaluation function; δ : Confidence for the Hoeffding test; d : Specialization depth; S_{min} : Clause G -score quality threshold; $mode$: Either streaming or single-lock.

```

1: function ProcessBatch( $batchI, H, B, G, \delta, d, N_{min}, mode, S_{min}$ ):
2:   for each  $I \in \mathcal{I}$  do
3:     Update  $TP_r, FP_r, FN_r$  and  $N_r$  counts from  $I$ , for each  $r \in H$  and each  $r' \in \rho_d(r)$ ,
       where  $N_r$  denotes the number of examples on which  $r$  has been evaluated so far.
4:     if ExpandTheory( $B, H, I$ ) then
5:        $H \leftarrow H \cup \text{StartNewClause}(B, I)$ 
6:     else
7:       for each clause  $r \in H$  do
8:          $r \leftarrow \text{ExpandClause}(r, G, \delta)$ 
9:        $H \leftarrow \text{Prune}(H, S_{min})$ 
10:  return  $H$ 
11: function StartNewClause( $B, I$ ):
12:  Generate a bottom clause  $\perp$  from  $I$  and  $B$ 
13:   $r := \text{head}(\perp) \leftarrow$ 
14:   $\perp_r := \perp$ 
15:   $N_r = FP_r = TP_r = FN_r := 0$ 
16:  For each other learner in system:
17:    Send message  $\text{NewRules}(r)$  to learner
18:  return  $r$ 
19: function ExpandClause( $r, G, \delta$ ):
20:  Compute  $\epsilon = \sqrt{\frac{\ln(1/\delta)}{2N_r}}$  and let  $\bar{G}$  denote the mean value of a clause's  $G$ -score
21:  Let  $r_1$  be the best specialization of  $r$ ,  $r_2$  the second best and  $\Delta\bar{G} = \bar{G}(r_1) - \bar{G}(r_2)$ 
22:  Let  $\tau$  equal the mean value of  $\epsilon$  observed so far
23:  if  $\bar{G}(r_1) > \bar{G}(r)$  and  $[\Delta\bar{G} > \epsilon$  or  $\tau < \epsilon]$ :
24:    For each other learner in system:
25:      Send message  $\text{StatsRequested}(r.id)$  to learner
26: function prune( $H, S_{min}$ ):
27:  Remove from  $H$  each clause  $r$  for which  $S_{min} - \bar{G}(r) > \epsilon$ , where  $\epsilon$  is the current Hoeffding bound
28:  For each other learner in system:
29:    Send message  $\text{DropClause}(r.id)$  to learner
30:  return  $H$ 

```

Algorithm 4 Learner's receive function in the Streaming Model

```

1: function Receive(MessageType):
2:   Case p : Proceed:
3:     If !finished :
4:       ProcessBatch(batchI, H, B, G,  $\delta$ , d, Nmin, Smin)
5:   Case s : StatsRequested:
6:     If rule that corresponds to s.id  $\in$  H:
7:       Send Response(stats) message containing rule's stats to s.requesterActor
8:     Else:
9:       If rule that corresponds to s.id  $\in$  oldClauses:
10:        Send Response(stats) message containing rule's stats to s.requesterActor
11:       Else:
12:        Resend s to self
13:   Case r : Response:
14:     If all learners have responded for r.id:
15:       Compute  $\epsilon = \sqrt{\frac{\ln(1/\delta)}{2N_r}}$  and let  $\bar{G}$ 
16:       denote the mean value of a clause's G-score
17:       Let r1 be the best specialization of r,
18:       r2 the second best and  $\Delta\bar{G} = \bar{G}(r_1) - \bar{G}(r_2)$ 
19:       Let  $\tau$  equal the mean value of  $\epsilon$  observed so far
20:       if  $\bar{G}(r_1) > \bar{G}(r)$  and [ $\Delta\bar{G} > \epsilon$  or  $\tau < \epsilon$ ]:
21:          $\perp_{r_1} := \perp_r$ 
22:         oldClauses := oldClauses  $\cup$  r
23:         For all learners:
24:           Send RefinedRule(r1, r.id)message
25:   Case d : DropClause:
26:     Remove clause r from H that matches d.id
27:     oldClauses := oldClauses  $\leftarrow$  clause

```

Πίνακας 4.3 – Συνάρτηση receive των Learner στο μοντέλο Ροής Πληροφορίας

Κεφάλαιο 5

Πειραματικά Αποτελέσματα και Παρατηρήσεις

5.1 Οργάνωση των πειραμάτων

Το σετ δεδομένων στο οποίο αξιολογήθηκε αρχικά ο αλγόριθμος του OLED και μετέπειτα εκτελέστηκαν πειραματικές μετρήσεις στα μοντέλα Ολικού Κλειδώματος είναι η βάση Caviar, στην οποία παρέχονται video frames κωδικοποιημένα σε διατυπώσεις μεμονωμένων γεγονότων χαμηλής κλίμακας (low-level events) που πραγματοποιούν διάφορα άτομα στο χώρο μαζί με διατυπώσεις συνδυαστικών γεγονότων υψηλής κλίμακας (high-level events) που πραγματοποιούνται μεταξύ των ατόμων αυτών¹.

Τα πειράματα εστιάζουν στη διατύπωση θεωριών για τις προϋποθέσεις οι οποίες πρέπει να πληρούνται προκειμένου να πραγματοποιούνται τα σύνθετα γεγονότα meeting και moving μεταξύ δύο ατόμων ids στο χώρο. Γίνεται σύγκριση του μονολιθικού συστήματος με τα μοντέλα Ολικού Κλειδώματος και Ροής Πληροφορίας. Η σύγκριση γίνεται για τα δύο γεγονότα υψηλής κλίμακας σε 2 βήματα:

- Χρησιμοποιώντας τη βάση Caviar στο μονολιθικό σύστημα και χωρίζοντας την σε 2,4 και 8 σετ δεδομένων στο κατανεμημένο σύστημα.
- Επαναλαμβάνοντας την παραπάνω μέτρηση ζητώντας από κάθε Learner να επεξεργαστεί 10

¹Πεισσότερες πληροφορίες μπορούν να αντληθούν από τον ιστότοπο <http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/>

φορές τα δεδομένα του μέχρι να εξάγει την τελική θεωρία.

Κάθε ένα από τα παραπάνω πειράματα χωρίζεται σε 10 επιμέρους υπο-πειράματα. Πραγματοποιείται 10-fold cross validation, καθώς το Caviar χωρίζεται σε κάθε επιμέρους πείραμα κατά αναλογία 90/10 σε training set και testing set. Το training set χρησιμοποιείται από κάθε σύστημα για να μάθει κανόνες και το testing set για να αξιολογηθούν οι κανόνες αυτοί. Στο τέλος συνδυάζονται τα αποτελέσματα και εξάγεται ο τελικός μέσος όρος κάθε πειράματος.

Σε κάθε πείραμα μας ενδιαφέρει να παρατηρήσουμε την αύξηση του speedup αναλογικά με την αύξηση των υπολογιστικών νημάτων, όπως επίσης και την ποιοτική σύγκριση των αποτελεσμάτων μεταξύ του μονολιθικού συστήματος και των κατανεμημένων σχεδιασμών σε 2,4 και 8 υπολογιστικά νήματα. Σε κάθε πείραμα παρουσιάζουμε δύο διαφορετικούς τύπους διαγραμμάτων:

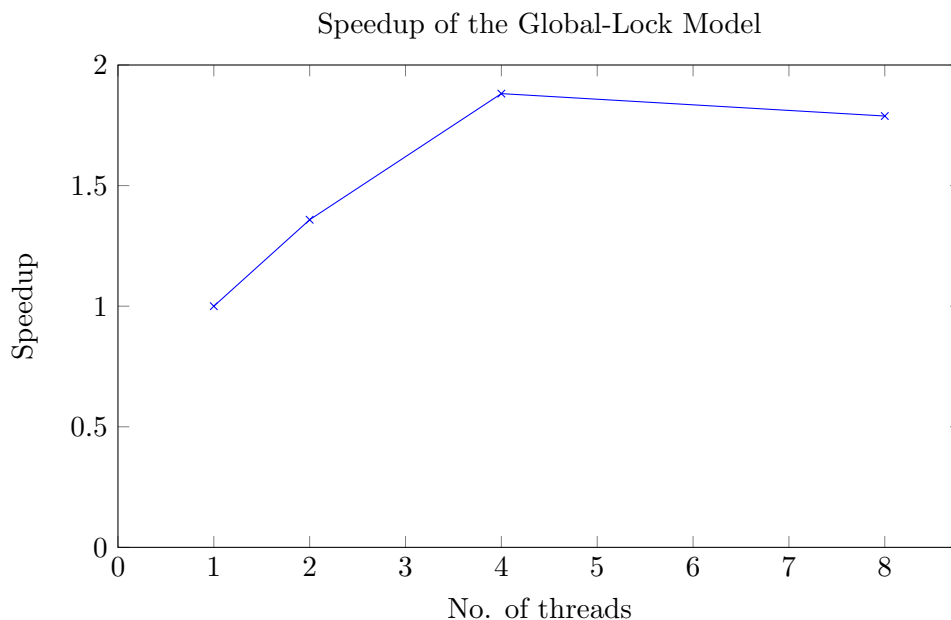
- Μεταβολή του speedup αναλογικά με την αύξηση των νημάτων.
- Συνδυαστικό διάγραμμα των παραμέτρων f-score της τελικής θεωρίας και του χρόνου περάτωσης της κάθε διαδικασίας.

5.2 Μοντέλο Ολικού Κλειδώματος

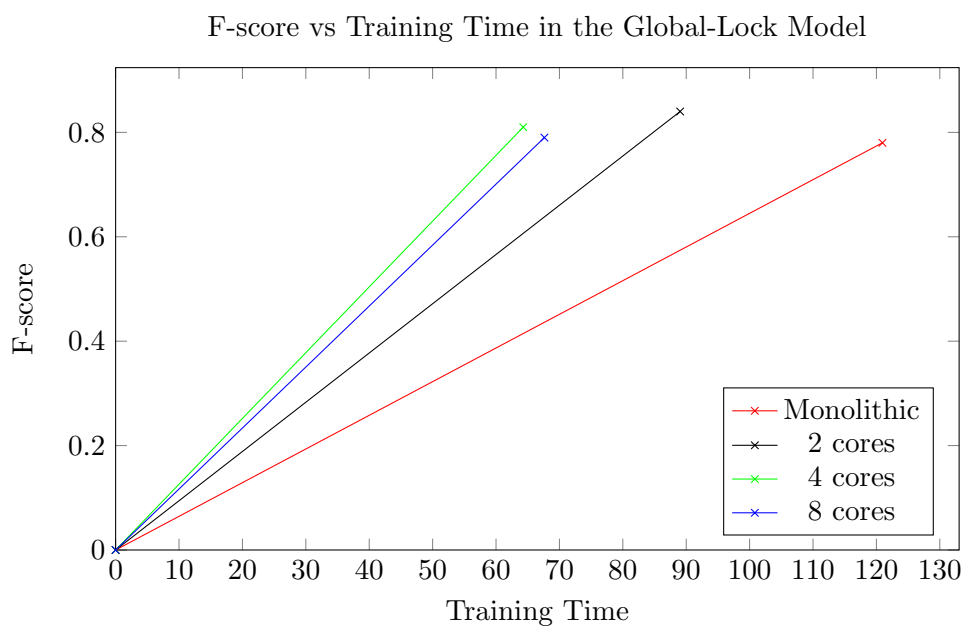
Στην παράγραφο αυτή παρουσιάζονται τα αποτελέσματα των πειραμάτων στο Μοντέλο Ολικού Κλειδώματος, για τα 2 γεγονότα υψηλής κλίμακας Meeting και Moving.

5.2.1 Meeting Event

Στα ακόλουθα 2 διαγράμματα τα συστήματα μαθαίνουν κανόνες για να περιγράψουν το γεγονός meeting σε απλή επεξεργασία του Caviar, δηλαδή κάθε διαθέσιμο παράδειγμα χρησιμοποιείται ακριβώς μία φορά:

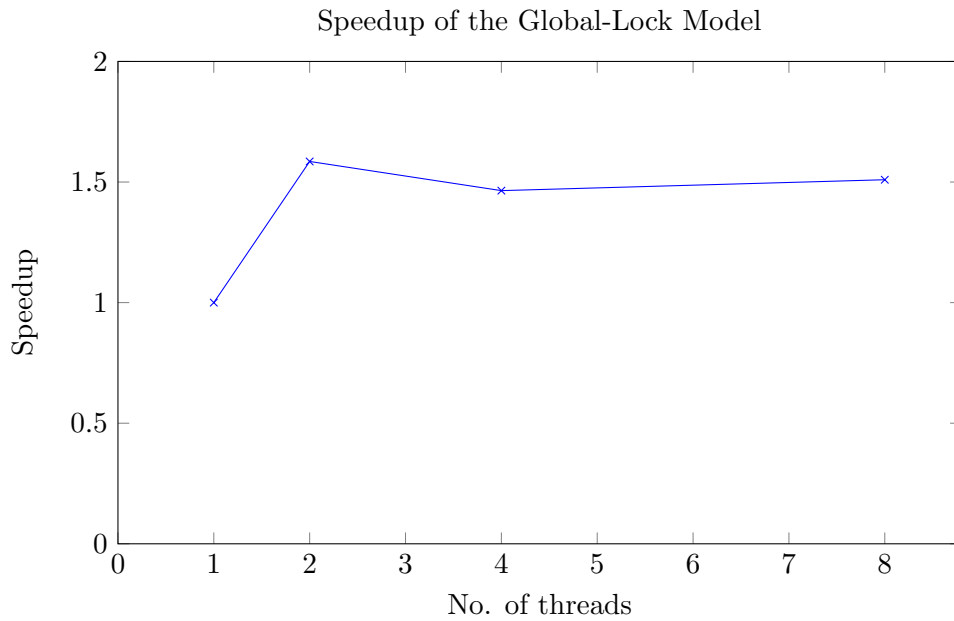


Σχήμα 5.1 – Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ολικού Κλειδώματος για το γεγονός meeting και απλή επεξεργασία του Caviar

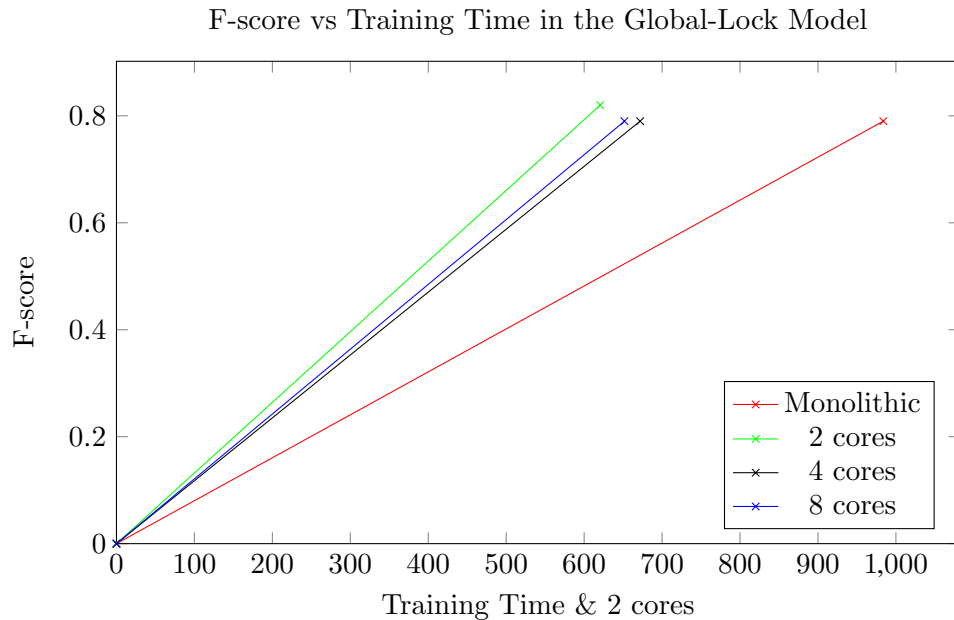


Σχήμα 5.2 – Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ολικού Κλειδώματος για το γεγονός meeting και απλή επεξεργασία του Caviar

Ακολουθούν τα αντίστοιχα διαγράμματα για το ίδιο γεγονός, στην εκτεταμένη εκδοχή του Caviar, όπου γίνεται χρήση των ίδιων παραδειγμάτων 10 φορές:



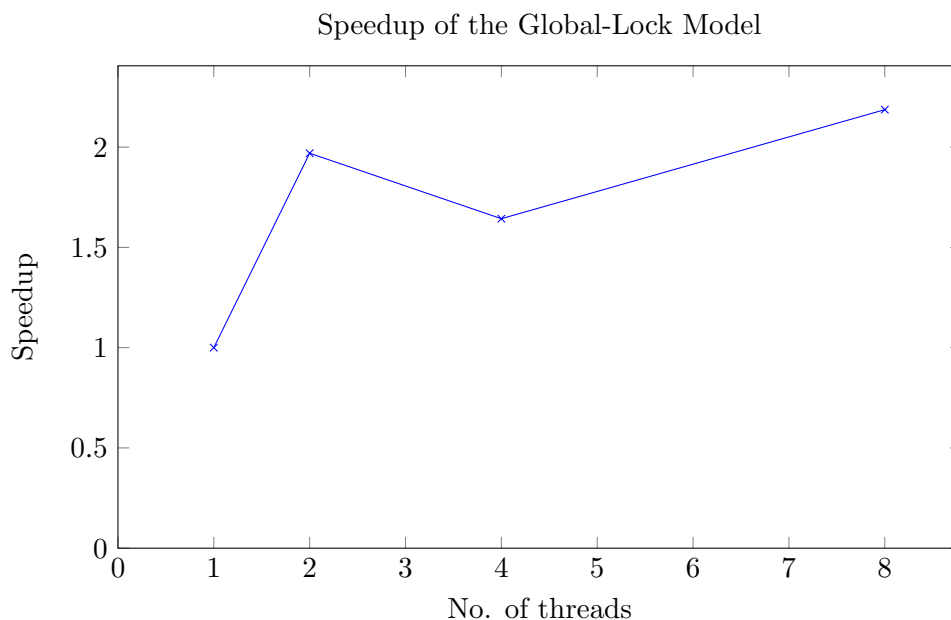
Σχήμα 5.3 – Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ολικού Κλειδώματος για το γεγονός meeting και εκτεταμένη επεξεργασία του Caviar



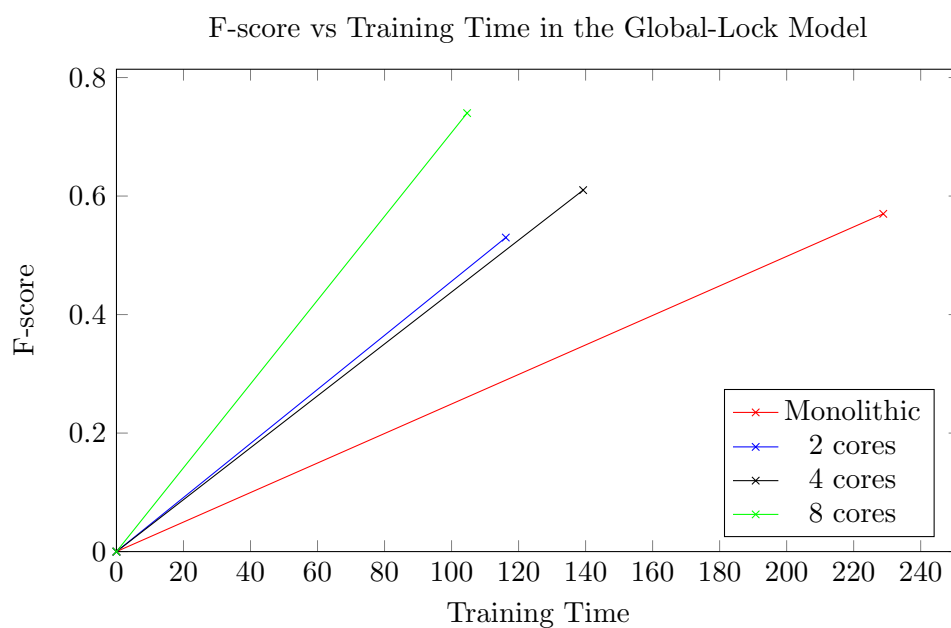
Σχήμα 5.4 – Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ολικού Κλειδώματος για το γεγονός meeting και εκτεταμένη επεξεργασία του Caviar

5.2.2 Moving Event

Ακολουθούν τα διαγράμματα σχετικά με τη μάθηση κανόνων σχετικά με το γεγονός moving σε απλή επεξεργασία του Caviar:

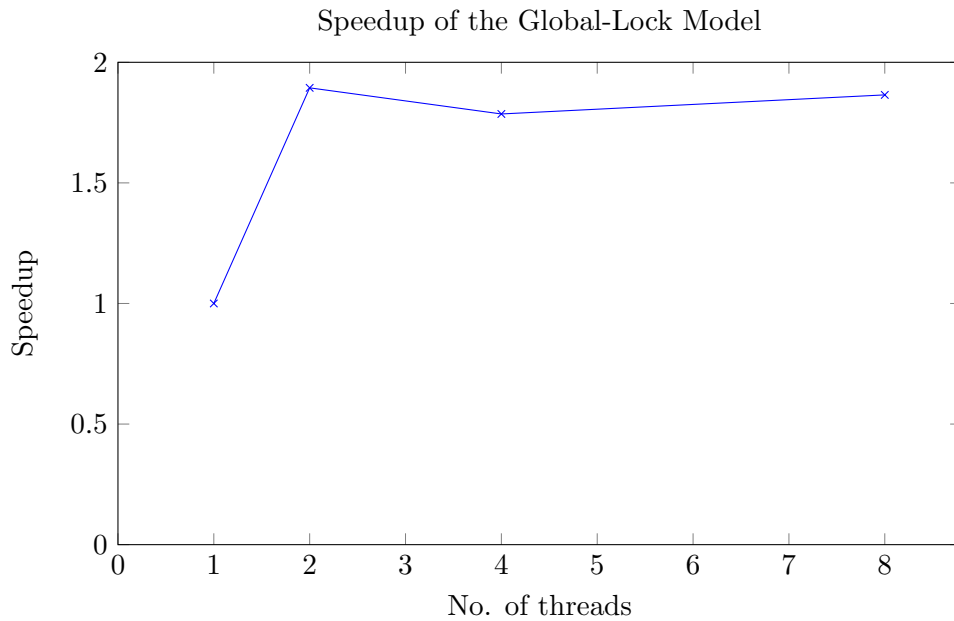


Σχήμα 5.5 – Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ολικού Κλειδώματος για το γεγονός moving και απλή επεξεργασία του Caviar

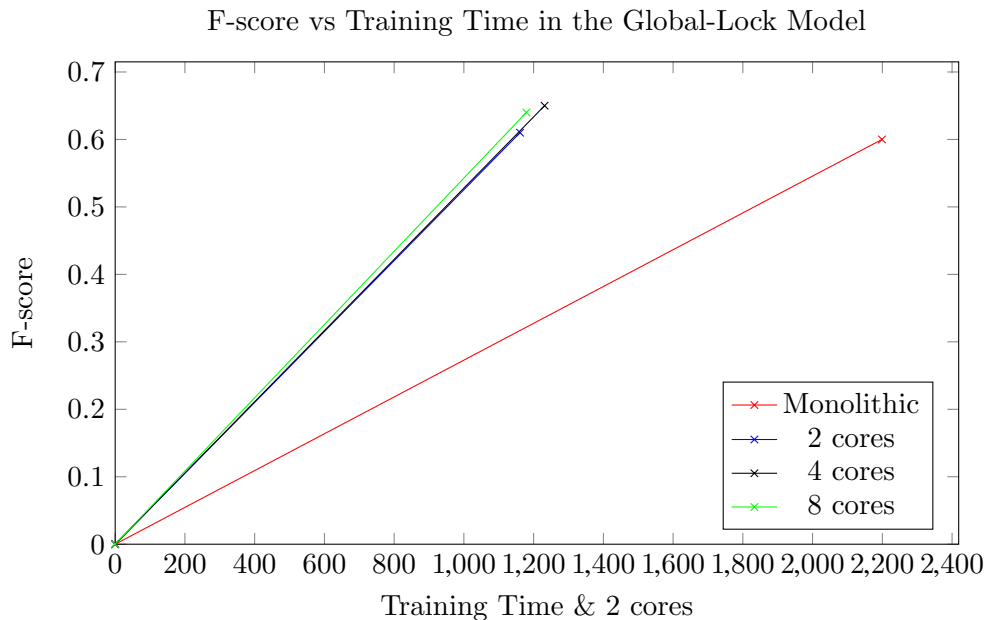


Σχήμα 5.6 – Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ολικού Κλειδώματος για το γεγονός moving και απλή επεξεργασία του Caviar

Ακολουθούν τα αντίστοιχα διαγράμματα για το ίδιο γεγονός, στην εκτεταμένη εκδοχή του Caviar, όπου γίνεται χρήση των ίδιων παραδειγμάτων 10 φορές:



Σχήμα 5.7 – Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ολικού Κλειδώματος για το γεγονός moving και εκτεταμένη επεξεργασία του Caviar



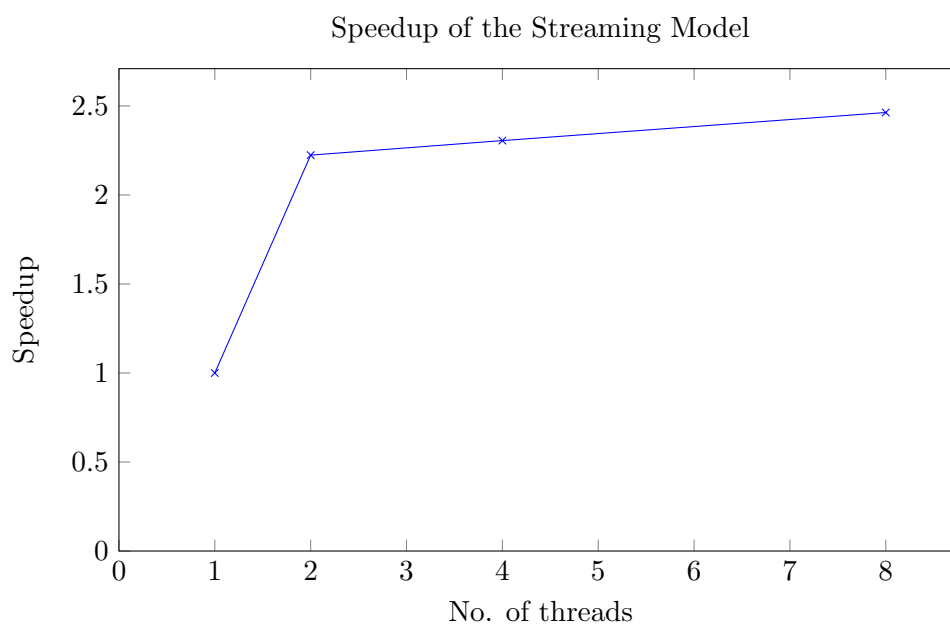
Σχήμα 5.8 – Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ολικού Κλειδώματος για το γεγονός moving και εκτεταμένη επεξεργασία του Caviar

5.3 Μοντέλο Ροής Πληροφορίας

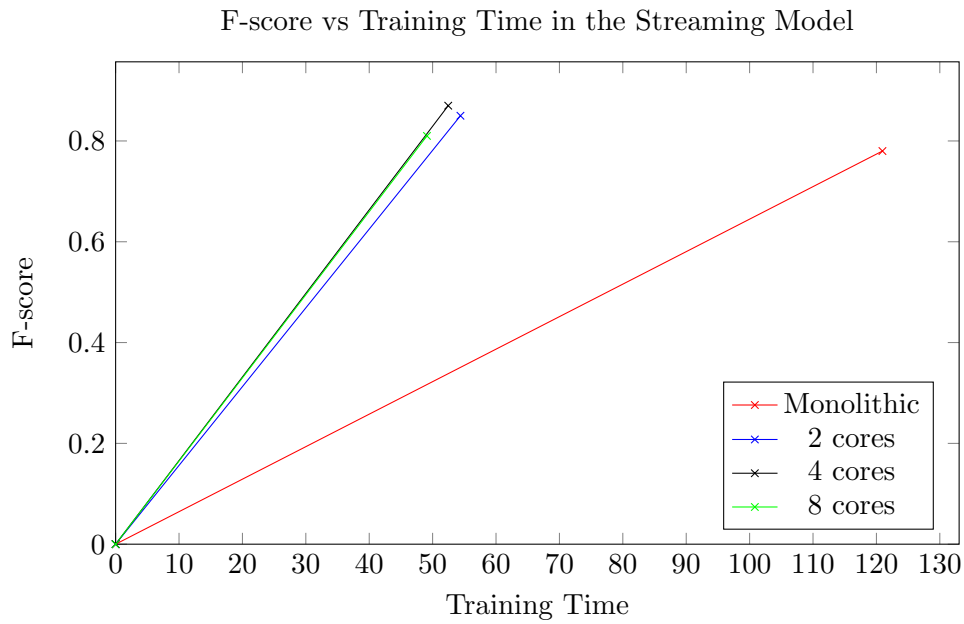
Στην παράγραφο αυτή παρουσιάζονται τα αποτελέσματα των πειραμάτων στο Μοντέλο Ροής Πληροφορίας, για τα 2 γεγονότα υψηλής κλίμακας Meeting και Moving.

5.3.1 Meeting Event

Στα ακόλουθα 2 διαγράμματα τα συστήματα μαθαίνουν κανόνες για να περιγράψουν το γεγονός meeting σε απλή επεξεργασία του Caviar, δηλαδή κάθε διαθέσιμο παράδειγμα χρησιμοποιείται ακριβώς μία φορά:

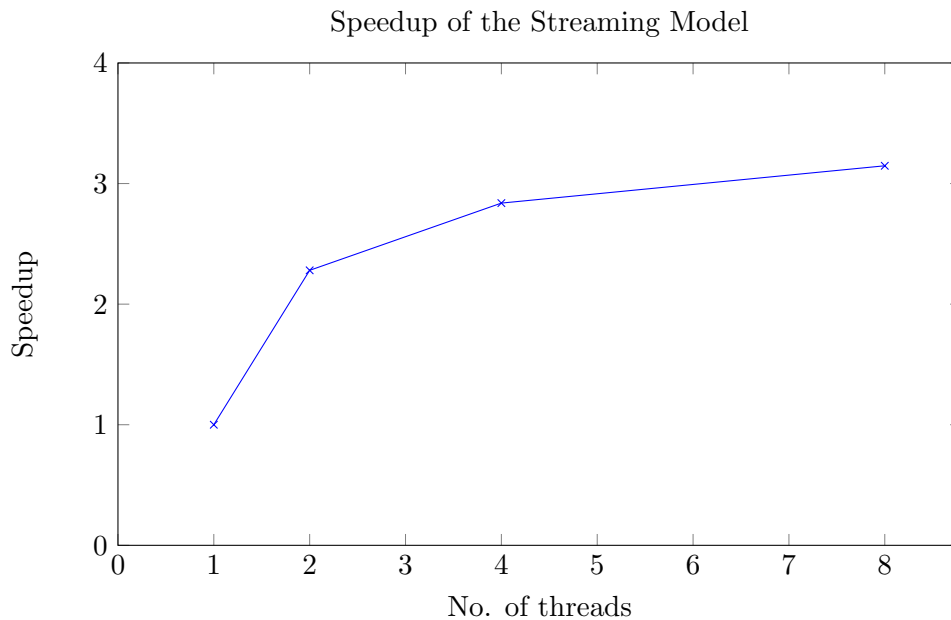


Σχήμα 5.9 – Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ροής Πληροφορίας για το γεγονός meeting και απλή επεξεργασία του Caviar

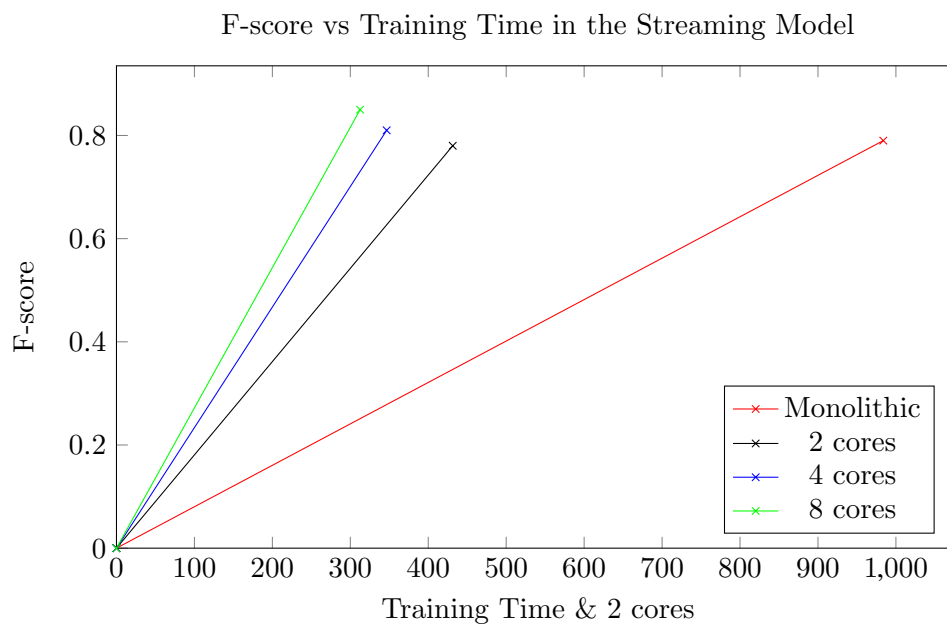


Σχήμα 5.10 – Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ροής Πληροφορίας για το γεγονός meeting και απλή επεξεργασία του Caviar

Ακολουθούν τα αντίστοιχα διαγράμματα για το ίδιο γεγονός, στην εκτεταμένη εκδοχή του Caviar, όπου γίνεται χρήση των ιδίων παραδειγμάτων 10 φορές:



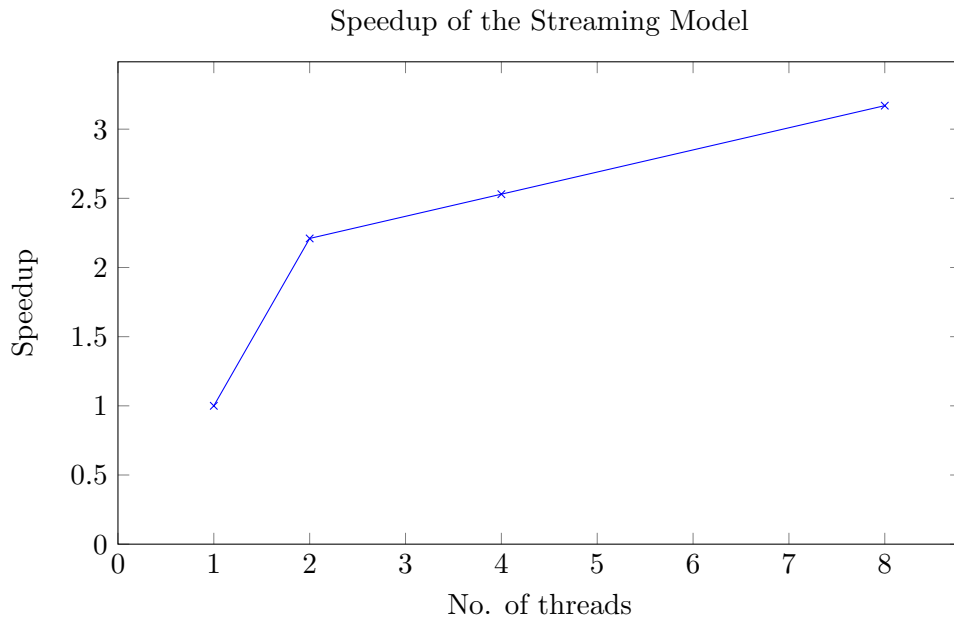
Σχήμα 5.11 – Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ροής Πληροφορίας για το γεγονός meeting και εκτεταμένη επεξεργασία του Caviar



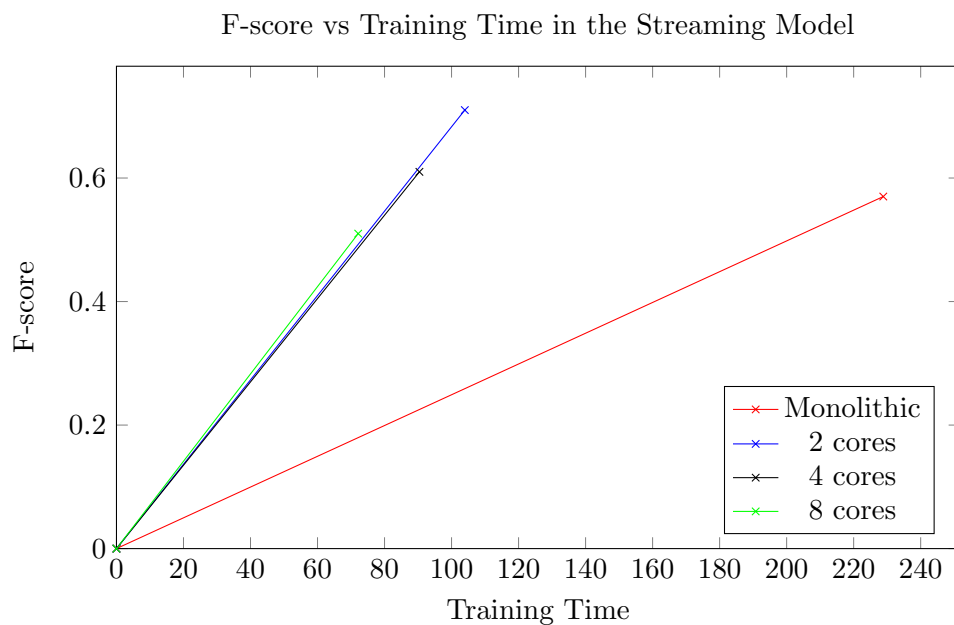
Σχήμα 5.12 – Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ροής Πληροφορίας για το γεγονός meeting και εκτεταμένη επεξεργασία του Caviar

5.3.2 Moving Event

Ακολουθούν τα διαγράμματα σχετικά με τη μάθηση κανόνων σχετικά με το γεγονός moving σε απλή επεξεργασία του Caviar:

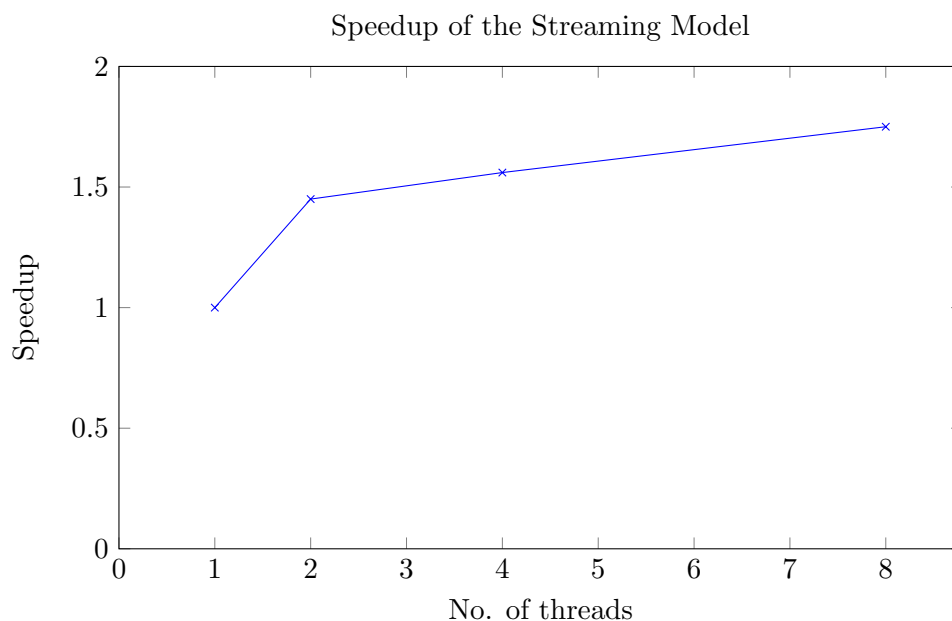


Σχήμα 5.13 – Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ροής Πληροφορίας για το γεγονός moving και απλή επεξεργασία του Caviar

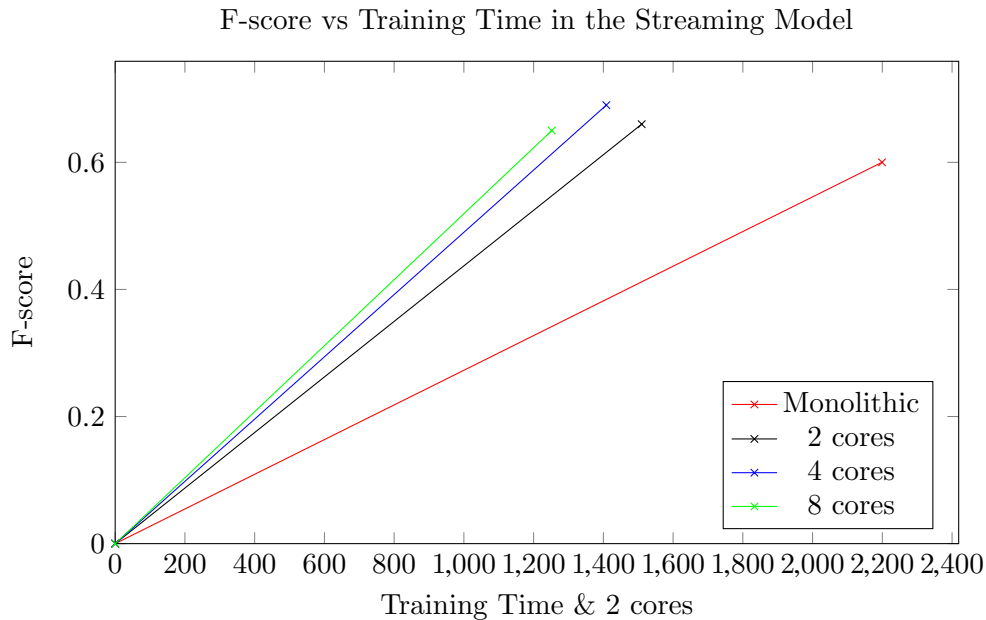


Σχήμα 5.14 – Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ροής Πληροφορίας για το γεγονός moving και απλή επεξεργασία του Caviar

Ακολουθούν τα αντίστοιχα διαγράμματα για το ίδιο γεγονός, στην εκτεταμένη εκδοχή του Caviar, όπου γίνεται χρήση των ίδιων παραδειγμάτων 10 φορές:



Σχήμα 5.15 – Μεταβολή του speedup συναρτήσει των υπολογιστικών νημάτων στο Μοντέλο Ροής Πληροφορίας για το γεγονός moving και εκτεταμένη επεξεργασία του Caviar



Σχήμα 5.16 – Συνδυαστικό αποτέλεσμα χρόνου και f-score στο Μοντέλο Ροής Πληροφορίας για το γεγονός moving και εκτεταμένη επεξεργασία του Caviar

5.4 Συγκεντρωτικός πίνακας αποτελεσμάτων

Ο ακόλουθος πίνακας συγκεντρώνει τα αποτελέσματα των μοντέλων Ολικού Κλειδώματος και Ροής Πληροφορίας ως προς το χρόνο μάθησης κανόνων, τη μεταβολή του speedup και την αποτελεσματικότητα (με κριτήριο την τιμή του f-score) τους.

	Global-Lock						Streaming						
	#cores	repeatFor1			repeatFor10			repeatFor1			repeatFor10		
		Time	Speedup	F-score	Time	Speedup	F-score	Time	Speedup	F-score	Time	Speedup	F-score
	1	120.95	1	0.78	983.79	1	0.79	120.95	1	0.78	983.79	1	0.79
Meeting	2	89.04	1.35	0.84	620.58	1.58	0.82	54.39	2.22	0.85	431.41	2.28	0.78
	4	64.29	1.88	0.81	671.82	1.46	0.79	52.46	2.31	0.87	346.66	2.83	0.81
	8	67.63	1.78	0.79	651.74	1.5	0.79	49.11	2.46	0.81	312.58	3.14	0.85
	1	228.84	1	0.57	2199.07	1	0.6	228.84	1	0.57	2199.07	1	0.6
Moving	2	116.19	1.96	0.53	1161.03	1.89	0.61	103.96	2.21	0.71	431.41	1.45	0.66
	4	139.27	1.64	0.61	1231.43	1.78	0.65	90.41	2.53	0.61	1408.44	1.56	0.69
	8	104.64	2.18	0.74	1179.21	1.86	0.64	72.14	3.17	0.51	1251.84	1.75	0.65

Πίνακας 5.1 – Συγκεντρωτικός πίνακας αποτελεσμάτων.

Κεφάλαιο 6

Επίλογος

Παρακάτω πραγματοποιείται μία σύνοψη της δουλειάς που πραγματοποιήθηκε κατά τη συγγραφή της εργασίας και γίνεται σχολιασμός των πειραματικών αποτελεσμάτων. Εν κατακλείδι, δίνονται προτάσεις οι οποίες μπορούν να αποτελέσουν τη βάση για τη βελτίωση και επέκταση της εργασίας αυτής.

6.1 Σύνοψη και Συμπεράσματα

Σκοπός της παρούσας εργασίας ήταν η σχεδίαση ενός συστήματος το οποίο να επεκτείνει τη λειτουργία του συστήματος Σύγχρονης Αναγνώρισης Σύνθετων Γεγονότων σε κατανεμημένα περιβάλλοντα επεξεργασίας, ώστε να παράγει κανόνες παρόμοιας ποιότητας με τους αντίστοιχους κανόνες που μαθαίνει το μονολιθικό σύστημα σε συγκριτικά μικρότερο χρονικό διάστημα επεξεργασίας των διαθέσιμων δεδομένων. Χρησιμοποιώντας ως βασικό εργαλείο της σχεδίασης το μαθηματικό μοντέλο των Δραστών, προτείνονται 2 διακριτά μοντέλα για την υλοποίηση της Κατανεμημένης Σύγχρονης Αναγνώρισης Σύνθετων Γεγονότων, το Μοντέλο Ολικού Κλειδώματος και το Μοντέλο Ροής Πληροφορίας.

Το Μοντέλο Ολικού Κλειδώματος διατηρεί σε μεγάλο βαθμό την ιεραρχία των βημάτων τα οποία θα ακολουθούσε το μονολιθικό σύστημα κατά την επεξεργασία ενός σετ δεδομένων. Μέσω ελέγχου της ροής των δεδομένων, αξιοποιούνται χρονικά διαστήματα στα οποία οι Δράστες παραμένουν ανενεργοί ως προς τον χειρισμό των παραδειγμάτων, προκειμένου η επεξεργασία τους σε κάθε σημείο της εκτέλεσης να πραγματοποιείται με τέτοιο τρόπο ώστε όλη η κρίσιμη πληροφορία να είναι διαθέσιμη σε κάθε Δράστη. Ως προς την ποιοτική σημασία των αποτελεσμάτων, μπορεί να γίνει

αντιληπτό πως η απόκλιση μεταξύ του μονολιθικού και κατανεμημένου συστήματος είναι ελάχιστη, γεγονός το οποίο επιβεβαιώνει σε μεγάλο βαθμό πως περνώντας από το μονολιθικό σύστημα στο MOK, τα βήματα επεξεργασίας διατηρούνται στο μέγιστο δυνατό βαθμό. Αντιθέτως, και παρότι ο χρόνος μάθησης κανόνων βελτιώνεται σημαντικά στο κατανεμημένο σύστημα συγκριτικά με το μονολιθικό, απόρροια των διαστημάτων αναμονής είναι το γεγονός πως μία πιθανή αύξηση του αριθμού των υπολογιστικών νημάτων δε συνεπάγεται απαραίτητα τη μείωση του χρόνου εκτέλεσης όσον αφορά αποκλειστικά το κατανεμημένο σύστημα.

Το Μοντέλο Ροής Πληροφορίας, εγκαταλείπει την τακτική των ανενεργών διαστημάτων και επιτρέπει στους Δράστες να μαθαίνουν κανόνες χωρίς να είναι δεδομένο πως σε κάθε χρονική στιγμή διαθέτουν όλη την πληροφορία που μπορεί να έχει δημιουργηθεί στο κατανεμημένο σύστημα. Κάθε Δράστης επεξεργάζεται αδιάκοπα τα παραδείγματα του και ενσωματώνει δεδομένα που παρήχθησαν από τους υπόλοιπους Δράστες του συστήματος ετεροχρονισμένα. Καθώς πλέον δεν τηρείται η ίδια αλληλουχία των βημάτων επεξεργασίας η ποιοτική απόκλιση των αποτελεσμάτων μεταξύ του μονολιθικού και του κατανεμημένου συστήματος είναι μεγαλύτερη σε σχέση με το MOK, παραμένοντας όμως σε υψηλά επίπεδα. Επιπλέον, αύξηση του αριθμού των υπολογιστικών νημάτων στο κατανεμημένο σύστημα συνεπάγεται και την ταυτόχρονη μείωση του χρόνου εκτέλεσης.

Αξίζει να παρατηρηθεί πως ανεξαρτήτως σχεδίασης, ο χρόνος που απαιτείται για να πραγματοποιηθεί η επεξεργασία των διαθέσιμων παραδειγμάτων μειώνεται δυσανάλογα με τη μείωση του πλήθους των παραδειγμάτων αυτών. Μετρήσεις που έγιναν στο μονολιθικό σύστημα, στις οποίες έγινε απότομη μείωση του αριθμού των παραδειγμάτων προς επεξεργασία, έδειξαν μικρή βελτίωση του χρονικού διαστήματος το οποίο απαιτούσε το σύστημα για να εξάγει την τελική θεωρία. Επομένως, ακόμα και σε ένα ιδανικό σύστημα κατανεμημένης επεξεργασίας, με μηδενική ανταλλαγή μηνυμάτων μεταξύ των Δραστών (η οποία συνεπάγεται και μηδενική χρονική καθυστέρηση για το χειρισμό πιθανών μηνυμάτων) η βελτίωση του χρόνου επεξεργασίας των παραδειγμάτων θα ήταν δυσανάλογη της αύξησης των διαθέσιμων υπολογιστικών νημάτων, γεγονός το οποίο σε μεγάλο βαθμό επιβεβαιώνεται και από τα αποτελέσματα του ΜΡΠ, κυρίως στην εκτεταμένη επεξεργασία του Caviar.

6.2 Μελλοντική Εργασία

Η βάση στην οποία σχεδιάστηκε η παρούσα εργασία ήταν η άμεση ενημέρωση όλων των μελών που απαρτίζουν το κατανεμημένο σύστημα με τις καινούριες πληροφορίες οι οποίες προέκυπταν σε μεμονωμένους κόμβους του συστήματος αυτού. Η προσέγγιση αυτή αναπόφευκτα οδήγησε στην

ανταλλαγή μεγάλου αριθμού μηνυμάτων μεταξύ των Δραστών που λειτουργούσαν το σύστημα και συνεπακόλουθα σε σημαντική χρονική επιβάρυνση του συστήματος, επιπρόσθετα της επιβάρυνσης την οποία συνεπάγεται μεμονωμένα η επεξεργασία των παραδειγμάτων, κάτι το οποίο έγινε εμφανές ιδιαίτερα στο Μοντέλο Ολικού Κλειδώματος. Η προσέγγιση αυτή μπορεί να θεωρηθεί εφάμιλλη των εφαρμογών κατανεμημένης μάθησης οι οποίες πραγματοποιούσαν ολικά επαναφορά του συστήματος όταν παρατηρούσαν απόκλιση μεταξύ ενός ολικού και ενός τοπικού μοντέλου αντίστοιχα, όπως παρουσιάστηκαν στο κεφάλαιο 2.

Σαν αποτέλεσμα, μία κατεύθυνση ως προς την οποία μπορεί να στραφεί η σχεδίαση μελλοντικών συστημάτων, είναι ο περιορισμός του πλήθους των μηνυμάτων με ταυτόχρονη χαλάρωση των κριτηρίων τα οποία οδηγούν στη δημιουργία και αποστολή μηνυμάτων από ένα Δράστη του συστήματος στους υπόλοιπους Δράστες οι οποίοι λειτουργούν στο σύστημα αυτό. Ενδεικτικά, τα κριτήρια επικοινωνίας θα μπορούσαν να προσαρμοστούν ως εξής:

- Οριοθέτηση ενός κάτω ορίου παραδειγμάτων μετά από την επεξεργασία των οποίων θα επιτρέπεται στους Δράστες του συστήματος να επικοινωνούν μεταξύ τους, ώστε παρόμοιες ποιοτικά πληροφορίες (όπως οι καινούριοι κανόνες παραδείγματος χάριν) να συμπτυχθούν σε μικρότερο πλήθος μηνυμάτων τα οποία θα επιφέρουν μικρότερη χρονική επιβάρυνση για τη μεταφορά τους στο σύστημα.
- Θέσπιση λιγότερο αυστηρών κριτηρίων επικοινωνίας τα οποία θα οδηγήσουν επίσης σε μικρότερο αριθμό μηνυμάτων. Ενδεικτικά μπορεί να τεθεί ένα κάτω όριο απόδοσης με τοπική εμβέλεια, με το οποίο ο κάθε Δράστης μεμονωμένα θα ελέγχει την απόδοση των κανόνων του, και εφόσον παραμένει πάνω από αυτό το όριο να μη κρίνεται απαραίτητη η επικοινωνία με τους υπόλοιπους Δράστες.

Ολοκληρώνοντας, και όσον αφορά το Μοντέλο Ολικού Κλειδώματος συγκεκριμένα, μπορεί να πραγματοποιηθεί επαναπροσδιορισμός του μοντέλου αξιοποιώντας τη δομή *become*¹ μέσω της οποίας ένας Δράστης μπορεί να υιοθετεί διαφορετικές ταυτότητες, και πρακτικά ένα υπολογιστικό νήμα μπορεί να λειτουργήσει σε διαφορετικές καταστάσεις οι οποίες στην αντίθετη περίπτωση θα απαιτούσαν περισσότερα υπολογιστικά νήματα. Σαν αποτέλεσμα, μπορούν να περιοριστούν οι αρμοδιότητες του Συντονιστή (και συνεπακόλουθα ο αριθμός των μηνυμάτων που αποστέλλει ανά τακτά χρονικά διαστήματα στους Learners του συστήματος), ή ακόμα και να παραληφθεί αυτό το επίπεδο με ενσωμάτωση περαιτέρω αρμοδιοτήτων στους Learners οι οποίοι δρουν σε ένα κατανεμημένο περιβάλλον βασισμένο στο σχεδιασμό του MOK.

¹<http://doc.akka.io/docs/akka/current/scala/actors.html>

Αυτή η σελίδα έχει σκοπίμως αφεθεί κενή.

Βιβλιογραφία

- [1] G. Agha. *Actors: a model of concurrent computation in distributed systems*. MIT Press Cambridge, MA, USA, 1986.
- [2] N. Aronsajn. Theory of reproducing kernels. *Transaction of the American Mathematical Society*, 68:(3):337–404, 1950.
- [3] A. Artikis, M. Sergot, , and G. Paliouras. An event calculus for event recognition. *Knowledge and Data Engineering, IEEE Transactions on* 27, 4:895–908, 2015.
- [4] A. Artikis, M. Sergot, and G. Paliouras. Behaviour recognition from video content:a logic programming approach. *International Journal on Artificial Intelligence Tools* 19, 02:193–209, 2010.
- [5] A. Artikis, A. Skarlatidis, F. Portet, and G. Paliouras. Logic-based event recognition. *The Knowledge Engineering Review* 27, 04:469–506, 2012.
- [6] H. Blockeel, L. De Raedt, N. Jacobs, and B. Demoen. scaling up inductive logic programming by learning from interpretations. *Data Mining and Knowledge discovery* 3, 1, 1999.
- [7] K. Clark. Negation as failure. in h. gallaire and j.minker, editors,. *Logic and Databases*, pages 293–322, 1978.
- [8] L. De Raedt. Logical and relational learning. *Springer Science and Business Media*, 2008.
- [9] A. Denecker and A. Kakas. Abduction in logic programming. *Computational Logic: Logic Programming and beyond*, pages 402–436.
- [10] P. Domingos and G. Hulten. Mining high-speed data streams. *In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 71–80, 2000.

- [11] O. Etzion and P. Niblett. Event processing in action. *Manning Publications Co.*, 2010.
- [12] M. Gabel, D. Keren, and A. Schuster. Monitoring least square models of distributed streams.
- [13] J. Gama. Knowledge discovery from data streams. *CRC Press.*, 2010.
- [14] N. Giatrakos, M. Deligiannakis, M. Garofalakis, I. Sharfman, and A. Schuster. Prediction-based geometric monitoring over distributed data streams. *In Proceedings of SIGMOD, ACM*, 2012.
- [15] Hewitt. Planner: a language for proving theorems in robots. *In Proceedings of the 1st international joint conference on Artificial intelligence (IJCAI'69)*, pages 295–301, 1969.
- [16] C. Hewitt, P. Bishop, and G. Steiger. A universal modular actor formalism for artificial intelligence. *In Proceedings of the 3rd international joint conference on Artificial intelligence*, pages 235–245, 1973.
- [17] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 301:13–30, 1958.
- [18] A. Horn. On sentences which are true of direct unions of algebras. *The journal of symbolic logic*, 16:14–21, 1951.
- [19] M. Kamp, M. Boley, D. Keren, A. Schuster, and I. Sharfman. Adaptive communication bounds for distributed online learning.
- [20] M. Kamp, M. Boley, D. Keren, A. Schuster, and I. Sharfman. Communication-efficient distributed online prediction by dynamic model synchronization. *Machine Learning and Knowledge Discovery in Databases*, pages 623–629, 2014.
- [21] M. Kamp, S. Bothe, M. Boley, and M. Mock. Communication-efficient distributed online learning with kernels. *Frasconi P., Landwehr N., Manco G., Vreeken J. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2016. Lecture Notes in Computer Science*, 9852, 2016.
- [22] N. Katzouris. Scalable relational learning for event recognition. *PhD Thesis, University of Athens*, 2017.
- [23] N. Katzouris, A. Artikis, and G. Paliouras. Incremental learning of event definitions. *Machine Learning*, 100:2–3,555–585, 2015.

-
- [24] N. Katzouris, A. Artikis, and G. Paliouras. Online learning of event definitions. *TPLP*, 16(5-6):817–833, 2016.
- [25] D. Keren, A. Sagy, A. Abboud, D. Ben-David, A. Schuster, I. Sharfman, and A. Deligianakis. Geometric monitoring of heterogeneous streams. In *IEEE Transactions of Knowledge and Data Engineering*, 2014.
- [26] R. Kowalski and M. Sergot. A logic-based of calculus events. *New Generation Computing*, 4:67–94, 1986.
- [27] M. Lesani and A. Lain. Semantics-preserving sharing actors. In *Proceedings of the 2013 workshop on Programming based on actors, agents, and decentralized control*, pages 69–80, 2013.
- [28] D. Luckham and R. Schulte. Event processing glossary - version 1.1. *Event Processing Technical Society.*, 2008.
- [29] J. R. Marden, H. Peyton Young, and L. Y. Pao. Achieving pareto optimality through distributed learning. *SIAM Journal on Control and Optimization*, 52, no. 5:2753–2770, 2014.
- [30] S. Muggleton. Inverse entailment and progol. *New Generation Computing*, 13:3–4,245–286, 1995.
- [31] M. Pontil. Learning in reproducing kernel hilbert spaces: a guide tour. *Bulletin of the Italian Artificial Intelligence Association – AI Notizie*, XVI(3):8–17, (2003).
- [32] B. Pradelski and H. Peyton Young. Learning efficient nash equilibria in distributed systems. *Games and Economics Behavior*, 75, no. 2:882–897, 2012.
- [33] B. Schölkopf and A. Ja Smola. Learning with kernels: Support vector machines, regularization, optimization, and beyond. *MIT press*, 2001.
- [34] M. Shanahan. The event calculus explained.
- [35] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press New York, NY, USA, 2008.
- [36] S. Tasharofi, P. Dinges, and R. E. Johnson. Why do scala developers mix the actor model with other concurrency models? In *ECOOP*, pages 302–326, 2013.

- [37] V. Vernon. *Reactive Messaging Patterns with the Actor Model*. Pearson Education, Inc., 2016.
- [38] H. Yang and S. Fong. Moderated vfdt in stream mining using adaptive tie threshold and incremental pruning. *Data Warehousing and Knowledge Discovery*, pages 471–483, 2011.