ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

# ΜΕΛΕΤΗ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ SCIKIT-MULTIFLOW ΓΙΑ ΤΗΝ ΕΞΟΡΥΞΗ ΓΝΩΣΗΣ ΣΕ ΔΕΔΟΜΕΝΑ ΡΟΗΣ

Διπλωματική Εργασία

του

Σταυρίδη Απόστολου

Θεσσαλονίκη, Σεπτέμβριος 2020

# ΜΕΛΕΤΗ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ SCIKIT-MULTIFLOW ΓΙΑ ΤΗΝ ΕΞΟΡΥΞΗ ΓΝΩΣΗΣ ΣΕ ΔΕΔΟΜΕΝΑ ΡΟΗΣ

Σταυρίδης Απόστολος

Πτυχίο Μαθηματικών Α.Π.Θ 2008

Διπλωματική Εργασία

υποβαλλόμενη για τη μερική εκπλήρωση των απαιτήσεων του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ

Επιβλέπων Καθηγητής
Ευαγγελίδης Γεώργιος

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 02/11/2020

| Ονοματεπώνυμο 1 | Ονοματεπώνυμο 2 | Ονοματεπώνυμο 3 |
|---|---|---|
| Ευαγγελίδης Γεώργιος | Γεωργία Κολωνιάρη | Δημήτρης Χρήστου-Βαρσακέλης |

Σταυρίδης Απόστολος

# Περίληψη

Στο υπολογιστικό μοντέλο των ροών δεδομένων, τα δεδομένα φθάνουν συνεχώς σε μια δυνητικά άπειρη ροή η οποίο πρέπει να υποβληθεί σε επεξεργασία από ένα σύστημα με περιορισμένους πόρους. Ο κύριος περιορισμός είναι ότι η κύρια μνήμη είναι μικρή και μπορεί να περιέχει μόνο ένα μικρό τμήμα του stream, επομένως τα περισσότερα δεδομένα πρέπει να απορρίπτονται αμέσως μετά την επεξεργασία. Η διαδικτυακή μάθηση ενημερώνει το μοντέλο της μετά από κάθε εμφάνιση δεδομένων χωρίς πρόσβαση σε όλα τα δεδομένα του παρελθόντος, εξ ου και ισχύουν οι περιορισμοί του υπολογιστικού μοντέλου ροής δεδομένων. Η ροή δεδομένων δεν είναι απλώς ένας τεχνικός περιορισμός στη μηχανική μάθηση, καθώς τα γρήγορα δεδομένα δεν αφορούν μόνο την ισχύ επεξεργασίας αλλά και τη γρήγορη σημασιολογία. Το scikit-multiflow αποτελεί ένα περιβάλλον μηχανικής μάθησης και εξόρυξης γνώσης ανοικτού κώδικα για δεδομένα πολλαπλών εξόδων / πολλαπλών ετικετών και ροών δεδομένων. Στην τρέχουσα κατάσταση του, το scikit-multiflow περιέχει γεννήτριες ροών δεδομένων, κατηγοριοποιητές πολλαπλών εξόδων / πολλαπλών ετικετών δεδομένων ροής, ανιχνευτές αλλαγής και μεθόδους αξιολόγησης. Σκοπός της μελέτης είναι η διερεύνηση και η σύγκριση υλοποιημένων αλγορίθμων κατηγοριοποίησης του scikit-multiflow πάνω σε ροές δεδομένων. Στην εργασία παρουσιάζεται ένα παράδειγμα, εργασίας ταξινόμησης στο οποίο χρησιμοποιείται η γεννήτρια SEA, της οποίας τα δεδομένα χρησιμοποιήθηκαν για την εκπαίδευση και τη σύγκριση ενός ταξινομητή Naive Bayes και ενός SGDClassifier. Παρόλο που ο NaiveBayes παρουσίασε καλύτερη απόδοση στην αρχή της ροής, ο SGDClassifier τελικά είχε λίγο καλύτερη απόδοση. Ο NaiveBayes ήταν γρηγορότερος και δέσμευσε ελαφρώς περισσότερη μνήμη, ωστόσο ο SGDClassifier αν και πιο αργός παρουσίασε μικρότερο αποτύπωμα μνήμης. Για την μελέτη της επίδρασης της απόκλισης στη μάθηση συγκρίθηκαν δύο δημοφιλή μοντέλα ροής, το HoeffdingTreeClassifier και η εκδοχή του HoeffdingAdaptiveTreeClassifier που λαμβάνει υπόψη την απόκλιση. Το HoeffdingAdaptiveTreeClassifier πέτυχε καλύτερη απόδοση ενώ απαιτούσε λιγότερο χώρο στη μνήμη. Αυτό δείχνει ότι έχει εφαρμοστεί ο μηχανισμός αντικατάστασης κλάδου που ενεργοποιείται από το ADWIN, με αποτέλεσμα μια λιγότερο περίπλοκη δομή δέντρου να αντιπροσωπεύει τα δεδομένα.

**Λέξεις Κλειδιά:** Ροές Δεδομένων, Scikit-Multiflow, Εξόρυξη Γνώσης

# Abstract

In the computational model of data flows, the data continuously reaches a potentially infinite flow that must be processed by a system with limited resources. The main limitation is that the main memory is small and may contain only a small part of the current, so most data should be discarded immediately after processing. Online learning updates its model after each data display without access to all past data, hence the limitations of the computer data flow model. Data flow is not just a technical constraint on machine learning, as fast data is not just about processing power but also about fast semantics. Scikit-multiflow is an open source machine learning and mining knowledge environment for multi-output / multi-tagged data and data streams. In its current state, scikit-multiflow contains feed generators, multiple output / multiple flow data tags, change detectors, and evaluation methods. The purpose of the study is to investigate and compare implemented scikit-multiflow categorization algorithms on data streams. The paper presents an example of a classification work using the SEA generator, the data of which were used to train and compare a Naive Bayes classifier and an SGDClassifier. Although NaiveBayes performed better at the beginning of the flow, the SGDClassifier eventually performed better, having difference from NaiveBayes. NaiveBayes was faster and took up slightly more memory, but the SGDClassifier, although slower, showed a smaller memory footprint. To study the effect of divergence on learning, two popular flow models were compared, the HoeffdingTreeClassifier and the HoeffdingAdaptiveTreeClassifier version that take divergence into account. The HoeffdingAdaptiveTreeClassifier performed better while requiring less memory space. This indicates that the ADWIN-enabled branch replacement mechanism has been implemented, resulting in a less complex tree structure representing the data.



**Keywords:** Data Flows, Scikit-Multiflow, Data Mining

# Περιεχόμενα

# Κατάλογος Σχημάτων

# 1 Εισαγωγή

## 1.1 Πρόβλημα – Σημαντικότητα του θέματος

Η διαδικασία της παραδοσιακής επεξεργασίας δεδομένων προϋποθέτει ότι τα δεδομένα είναι διαθέσιμα για πολλαπλή πρόσβαση, ακόμη και αν ορισμένες φορές βρίσκονται αποθηκευμένα σε κάποιο μέσο και δύνανται να επεξεργαστούν μόνο σε μεγαλύτερα μέρη, με τα συστήματα βάσεων δεδομένων να αποτελούν ένα χαρακτηριστικό παράδειγμα, καθώς αποθηκεύουν μεγάλες συλλογές δεδομένων και επιτρέπουν στον χρήστη να κάνει ερωτήματα και να εξάγει πληροφορίες. Ωστόσο, τα γρήγορα δεδομένα ή τα δεδομένα σε ροή επεξεργαζονται απο υπολογιστικά μοντέλά των ροών δεδομένων. Σε αυτό το μοντέλο, τα δεδομένα φθάνουν συνεχώς σε μια δυνητικά άπειρη ροή η οποίο πρέπει να υποβληθεί σε επεξεργασία από ένα σύστημα με περιορισμένους πόρους. Ο κύριος περιορισμός είναι ότι η κύρια μνήμη είναι μικρή και μπορεί να περιέχει μόνο ένα μικρό τμήμα του stream, επομένως τα περισσότερα δεδομένα πρέπει να απορρίπτονται αμέσως μετά την επεξεργασία. Η διαδικτυακή μάθηση ενημερώνει το μοντέλο της μετά από κάθε εμφάνιση δεδομένων χωρίς πρόσβαση σε όλα τα δεδομένα του παρελθόντος, εξ ου και ισχύουν οι περιορισμοί του υπολογιστικού μοντέλου ροής δεδομένων. Η ροή δεδομένων δεν είναι απλώς ένας τεχνικός περιορισμός στη μηχανική μάθηση, καθώς τα γρήγορα δεδομένα δεν αφορούν μόνο την ισχύ επεξεργασίας αλλά και τη γρήγορη σημασιολογία.

## 1.2 Σκοπός – Στόχοι

Σκοπός της μελέτης είναι η διερεύνηση και η σύγκριση υλοποιημένων αλγορίθμων κατηγοριοποίησης του scikit-multiflow πάνω σε ροές δεδομένων. Επιπλέον, θα αναλυθούν αλγόριθμοι απομείωσης δεδομένων σε πραγματικό χρόνο (π.χ., DRHC, AIB2) και θα αποτιμηθεί η απόδοση της κατηγοριοποίησης με ή χωρίς απομείωση των δεδομένων. Η μεθοδολογία που θα ακολουθηθεί για την πραγμάτωση του παραπάνω στόχου είναι η κριτική αποτίμηση των διεθνών βιβλιογραφικών πηγών που σχετίζονται με το εν λόγω ζήτημα.

## 1.3 Συνεισφορά

Μέσω της σύγκρισης των υλοποιημένων αλγορίθμων κατηγοριοποίησης του scikit-multiflow πάνω σε ροές δεδομένων και της ανάλυσης των αλγορίθμων

απομείωσης δεδομένων σε πραγματικό χρόνο πρόκειται να εξαχθούν σημαντικά συμπεράσματα που σχετίζονται με τις αλγοριθμικές τεχνικές και τους πόρους που απαιτούνται για την επεξεργασία γρήγορων δεδομένων στο υπολογιστικό μοντέλο των ροών δεδομένων.

## 1.4 Διάρθρωση της μελέτης

Η εργασία δομείται σε πέντε κεφάλαια, εκ των οποίων το πρώτο αποτελεί το κεφάλαιο της εισαγωγής όπου αναλύεται ο σκοπός και η σημαντικότητα της εργασίας. Το δεύτερο κεφάλαιο αποτελεί το κεφάλαιο της βιβλιογραφικής επισκόπησης του θεωρητικού υποβάθρου της εργασίας, όπου περιγράφονται βασικές έννοιες όπως η μηχανική μάθηση και τα είδη της, οι αλγόριθμοι κατηγοριοποίησης βασισμένοι σε δένδρα και η διαδικτυακή μηχανική μάθηση σε ροές μεγάλων δεδομένων. Στη συνέχεια, στο τρίτο κεφάλαιο παρουσιάζεται η μεθοδολογία που ακολουθήθηκε στην εργασία, ενώ στο τέταρτο κεφάλαιο αναλύονται τα αποτελέσματα της μελέτης. Τέλος, στο πέμπτο κεφάλαιο συνοψίζονται τα βασικά συμπεράσματα της εργασίας και περιγράφονται οι περιορισμοί που εμπεριέχει καθώς και οι προτάσεις για επέκταση της μελέτης στο μέλλον.

# 2 Βιβλιογραφική Επισκόπηση – Θεωρητικό Υπόβαθρο

## 2.1 Μηχανική Μάθηση & Αλγόριθμοι Κατηγοριοποίησης

### *2.1.1 Μηχανική Μάθηση*

Η μηχανική μάθηση είναι ένας υποτομέας της επιστήμης των υπολογιστών που ασχολείται με την κατασκευή αλγορίθμων οι οποίοι, για να είναι χρήσιμοι, βασίζονται σε μια συλλογή παραδειγμάτων κάποιου φαινομένου. Τα εν λόγω παραδείγματα δύνανται να προέρχονται από τη φύση, να κατασκευάζονται χειροποίητα από κάποιον άνθρωπο ή να παράγονται από άλλους αλγορίθμους.

Η διαδικασία της μηχανικής μάθησης ορίζεται επίσης ως η επίλυση ενός πρακτικού προβλήματος με α) τη διαδικασία συλλογής ενός συνόλου δεδομένων και β) τη διαδικασία της αλγοριθμικής δημιουργίας ενός στατιστικού μοντέλου που έχει ως βάση του το σύνολο των δεδομένων, με τον εν λόγω στατιστικό μοντέλο να θεωρείται πως χρησιμοποιείται για να επιλυθεί το πρακτικό πρόβλημα (Michie et al., 1994).

### *2.1.2 Είδη Μηχανικής Μάθησης*

#### *2.1.2.1. Μηχανική Μάθηση με Επίβλεψη*

Στη διαδικασία της εποπτευόμενης μάθησης, τα δεδομένα είναι η συλλογή επισημασμένων παραδειγμάτων$\{(x_i, y_i)\}_{i=1}^{N}$.Κάθε στοιχείο $x_i$ μεταξύ των N ονομάζεται feature vector, ενώ κάθε feature vector είναι ένα διάνυσμα στο οποίο κάθε διάσταση j = 1,. . . , D και περιέχει μια τιμή που περιγράφει με κάποιο τρόπο το παράδειγμα, με την εν λόγω τιμή να ονομάζεται χαρακτηριστικό και να δηλώνεται ως $x^{(j)}$. Για παράδειγμα, αν το κάθε παράδειγμα x στη συλλογή μας αντιπροσωπεύει ένα αυτοκίνητο, τότε το πρώτο χαρακτηριστικό, $x^{(1)}$, δύναται να περιέχει τα χιλιομετρα που έχει διανύσει σε Km, το δεύτερο χαρακτηριστικό, $x^{(2)}$, τον κυβισμό σε cc, το $x^{(3)}$το φύλλο του οδηγού που το οδηγούσε κτλ. Για το σύνολο των παραδειγμάτων στα δεδομένα, το χαρακτηριστικό στη θέση j στο διάνυσμα χαρακτηριστικών πάντα περιέχει το ίδιο είδος πληροφοριών, πράγμα που σημαίνει ότι αν $x_i^{(2)}$ περιέχει κυβισμό σε cc, σε κάποιο παράδειγμα $x_i$, τότε το $x_k^{(2)}$ θα περιέχει επίσης τον κυβισμό σε κάθε παράδειγμα $x_k$, k = 1,. . . , N. Η label $y_i$ δύναται να είναι είτε ένα στοιχείο που ανήκει σε ένα πεπερασμένο σύνολο κατηγοριών {1, 2,. . . , C}, είτε ένας πραγματικός αριθμός ή μια πιο περίπλοκη δομή, όπως πχ ένα διάνυσμα, ένας πίνακας, ένα δέντρο ή ένα γράφημα. Εκτός αν αναφέρεται διαφορετικά,

του$_i$ είναι είτε ένα πεπερασμένο σύνολο τάξεων είτε ένας πραγματικός αριθμός (Kotsiantisetal., 2007). Μια τάξη είναι η κατηγορία στην οποία έγκειται ένα παράδειγμα, ενώ αν τα παραδείγματα είναι μηνύματα ηλεκτρονικού ταχυδρομείου και το πρόβλημα έγκειται στην ανίχνευση των ανεπιθύμητων μηνυμάτων, τότε τίθενται δύο κατηγορίες {spam, not_spam}.

Ο σκοπός των αλγορίθμων εποπτευόμενης μάθησης είναι να χρησιμοποιούν το σύνολο των δεδομένων στην κατεύθυνση της παραγωγής ενός μοντέλου που παίρνει ένα διάνυσμα χαρακτηριστικών x ως είσοδο και εξάγει πληροφορία που επιτρέπει την εξαγωγή μιας label για το εν λόγω διάνυσμα χαρακτηριστικών. Για παράδειγμα, το μοντέλο που δημιουργήθηκε χρησιμοποιώντας το σύνολο δεδομένων των αυτοκινήτων θα μπορούσε να λάβει ως είσοδο ένα διάνυσμα χαρακτηριστικών που περιγράφει ένα αυτοκίνητο και να εξάγει μια πιθανότητα το αυτοκίνητο να έχει μηχανικό πρόβλημα.

#### *2.1.2.2. Μη Εποπτευόμενη Μηχανική Μάθηση*

Στην μη εποπτευόμενη μάθηση, το σύνολο δεδομένων είναι μια συλλογή μη χαρακτηρισμένων παραδειγμάτων$\{(x_i, y_i)\}_{i=1}^{N}$ . Και πάλι, το x είναι ένα διάνυσμα χαρακτηριστικών και σκοπός των αλγορίθμων μάθησης χωρίς επίβλεψη είναι να δημιουργήσουν μοντέλα που παίρνουν ένα διάνυσμα χαρακτηριστικών x ως είσοδο και ή το μετατρέπουν σε ένα άλλο διάνυσμα ή σε μια τιμή που δύναται να χρησιμοποιηθεί για να επιλυθεί ένα πρακτικό πρόβλημα. Για παράδειγμα, κατά τη διαδικασία της ομαδοποίησης, τα μοντέλα επιστρέφουν την ταυτότητα της ομάδας για κάθε διάνυσμα χαρακτηριστικών στα δεδομένα (Kassambara, 2017).

Στη διαδικασία μείωσης των διαστάσεων, η έξοδος των μοντέλων είναι ένα διάνυσμα χαρακτηριστικών που έχει λιγότερα χαρακτηριστικά από την είσοδο x, στην ανίχνευση των εξόδων (outlierdetection), η έξοδος είναι ένας πραγματικός αριθμός που υποδεικνύει τον τρόπο με τον οποίο το x είναι διαφορετικό από ένα "τυπικό" παράδειγμα στο σύνολο δεδομένων, ενώ η πλέον εξέχουσα τάξη μεθόδων μάθησης χωρίς επίβλεψη είναι η συσσωμάτωση όπου οι περιπτώσεις οφείλουν να διανέμονται σε ένα πεπερασμένο σύνολο συστάδων, προκειμένου οι περιπτώσεις μέσα στο cluster να είναι πιο παρόμοιες μεταξύ τους απ 'ό, τι οι άλλες σε διαφορετικές συστάδες, (Pang-Ning et al, 2006). Οι αλγόριθμοι batch aggregation έχουν μελετηθεί και χρησιμοποιηθεί ως εργαλεία ανάλυσης δεδομένων για δεκαετίες, (Jain et al, 1999). Μια συχνά εφαρμοζόμενη μέθοδος ομαδοποίησης είναι τα kmeans, (Hartigan & Hrtigan, 1975), όπου η επιλογή του

κέντρου συμπλέγματος και η αντιστοίχιση στα πλησιέστερα κέντρα εκτελούνται κατ'επανάληψη μέχρι τη σύγκλιση. Ένα άλλο είναι το DBSCAN, (Ester et al, 1996), που αποτελεί μια μέθοδο που έχει ως βάση της την πυκνότητα που συγκεντρώνει σημεία που έχουν στενή σύνδεση μεταξύ τους.

Οι αλγόριθμοι ηλεκτρονικής ομαδοποίησης εξετάζονται μεταξύ άλλων από τον Mahdiraji, (2009). Η πλειονότητα των πλέον σχετικών μεθόδων είναι οι εκδόσεις ρευμάτων δεδομένων των k-means ή των παραλλαγών τους όπως οι k-mediians (Zhang et al, 1996), ενώ ένα άλλο σύνολο αποτελεσμάτων περιγράφει την εφαρμογή ροής δεδομένων του DBSCAN, (Cao et al, 2006). Εν κατακλείδι, οι online αλγόριθμοι ιεραρχικής ομαδοποίησης που διατηρούν μέτρα ομοιότητας και ιεραρχικά συγχωνεύουν πιο κοντινές συστάδες περιγράφονται στην εργασία των Rodrigues et al, (2006).

Η εύρεση συχνών αντικειμένων, (Agrawal et al, 1993), αποτελεί μια άλλη κεντρική εργασία εξόρυξης δεδομένων δίχως επιτήρηση, στατική αλλά και συνεχούς ροής, για έναν πίνακα συναλλαγών και αντικειμένων δηλαδή στόχος είναι να βρεθούν όλα τα υποσύνολα στοιχείων που εμφανίζονται μαζί σε συναλλαγές με τουλάχιστον μια προκαθορισμένη συχνότητα, με πολλές παραλλαγές της εργασίας να περιγράφονται από τους Aggarwal & Han, (2014). Οι αλγόριθμοι εξόρυξης συχνών στοιχειωδών στοιχείων σε απευθείας σύνδεση εξετάζονται από τους Cheng et al, (2008). Αλγόριθμοι που έχουν ως βάση τους μετρήσεις όλων των προηγούμενων δεδομένων στο stream, (Chang & Lee, 2003a), μπορούν να ονομαστούν και προσεγγίσεις βασισμένες σε παράθυρα. Σε μερικούς από τους εν λόγω αλγορίθμους, παρατηρείται επίτευξη της προσαρμοστικότητας χρόνου με περισσότερη έμφαση στα πρόσφατα αντικείμενα, (Chang & Lee, 2003a). Οι προσεγγίσεις που βασίζονται σε συρόμενο παράθυρο, (Chang & Lee, 2003b) είναι ιδιαίτερα κατάλληλες για την επεξεργασία δεδομένων με ιδεατή μετατόπιση. Σε μια συγκριτική επισκόπηση, δύναται κάποιος να δει, για παράδειγμα, τον τρόπο με τον οποίο επιλέχθηκε ο αλγόριθμος του MOA (Quadrana et al, 2015). Αξίζει να σημειωθεί ότι μια ειδική υποδιαίρεση, η εύρεση δηλαδή συχνών στοιχείων σε ροές δεδομένων, θεωρείται ήδη προκλητική, απαιτώντας προσεγγιστικές δομές δεδομένων (Charikar et al, 2004).

Η διαδικασία της βασικής ανάλυσης συνιστωσών (PCA) αποτελεί ένα ισχυρό εργαλείο για να μειωθούν οι διαστάσεις, (Jolliffe, 1986), βάσει της παραγοντοποίησης μήτρας. Οι παραλλαγές σε απευθείας σύνδεση βασίζονται σε ιδέες για την επικαιροποιημένη κλιμάκωση της αποσύνθεσης του πίνακα, (Bunch & Nielsen, 1978). Οι πρώτοι από τους

αλγόριθμους PCA που θεωρούνται κατάλληλοι για τη διαδικασία της ηλεκτρονικής μάθησης έχουν ως βάση τους τα νευρωνικά δίκτυα, (Oja, 1982), ενώ παρομοίως με τα γραμμικά μοντέλα, ο PCA δύναται ακόμη να εφαρμόσει το κόλπο kernel προς την κατεύθυνση συμπερίληψης της μη γραμμικής μοντελοποίησης (Scholkopf et al, 1998). Ο επαναληπτικός kernel PCA περιγράφεται στην εργασία των Kim et al, (2005), και ο online kernel PCA στην εργασία του Honeine, (2012). Σημειώνεται ότι για την αναζήτηση πλησιέστερων γειτόνων στο χώρο χαμηλών διαστάσεων που παρέχεται από το PCA, τα ερευτικά στοιχεία για την επιλογή μεγάλων εσωτερικών προϊόντων εφαρμόζονται, (Teflioudi et al, 2015).

Η πιθανή μοντελοποίηση θέματος ταιριάζει με πολύπλοκα ιεραρχικά Bayesian μοντέλα σε μεγάλες συλλογές εγγράφων. Τα μοντέλα θέματος αποκαλύπτουν λανθάνουσα σημασιολογική δομή που δύναται να χρησιμοποιηθεί για πλήθος εφαρμογών και ενώ τα μοντέλα που μοιάζουν με PCA δύνανται επίσης να χρησιμοποιηθούν για λανθάνουσα σημασιολογική ανάλυση, (Deerwester et al, 1990), πρόσφατα απέκτησε δημοτικότητα ο αλγόριθμος Latent Dirichlet Allocation (LDA), (Blei et al, 2003). Οι πλειονότητα των παραμέτρων μοντέλου θέματος δύνανται να συναχθούν μόνο βάσει της δειγματοληψίας Markov Chain Monte Carlo, μια μέθοδος που δύσκολα εφαρμόζεται στην ηλεκτρονική μάθηση. Η συνεισφορά LDA είναι δυνατή με βάση είτε την online δειγματοληψία Gibbs, (Song et al, 2005), είτε τη στοχαστική βελτιστοποίηση στο διαδίκτυο με ένα βήμα φυσικής κλίσης, (Hoffman et al, 2010), ενώ διάφορες σε απευθείας σύνδεση παραλλαγές LDA περιγράφονται στην εργασία των Smola & Narayanamurthy, (2010).

#### *2.1.2.3. Διαδικασία Ημι-Εποπτευόμενης Μηχανικής Μάθησης*

Στη διαδικασία ημι-εποπτευόμενης μάθησης, το σύνολο δεδομένων περιέχει τόσο επισημασμένα όσο και μη επισημασμένα παραδείγματα και συνήθως, τα μη επισημασμένα παραδείγματα είναι πολύ περισσότερα από τα επισημασμένα παραδείγματα. Ο σκοπός ενός ημι-εποπτευόμενου αλγορίθμου μάθησης είναι ο ίδιος με τον στόχο του αλγόριθμου εποπτευόμενης μάθησης, με την ελπίδα εδώ να έγκειται στο ότι η χρήση πολλών μη επισημασμένων παραδειγμάτων δύναται να βοηθήσει τον αλγόριθμο μάθησης να βρει ("παράγει" ή "υπολογίσει") ένα καλύτερο μοντέλο (Zhu, 2005).

#### *2.1.2.4. Ενισχυμένη Μάθηση*

Η διαδικασία της ενισχυμένης μάθησης αποτελεί ένα υποπεδίο της μηχανικής μάθησης όπου η μηχανή "ζει" σε ένα περιβάλλον και είναι ικανή να αντιληφθεί την κατάσταση αυτού του περιβάλλοντος ως ένα διάνυσμα χαρακτηριστικών, με το μηχάνημα μπορεί να εκτελεί ενέργειες σε κάθε κατάσταση, ενώ οι διαφορετικές ενέργειες φέρνουν διαφορετικές ανταμοιβές και δύνανται επίσης να μετακινήσουν το μηχάνημα σε άλλη κατάσταση του περιβάλλοντος. Ο σκοπός ενός αλγόριθμου ενίσχυσης μάθησης είναι να μάθει μια πολιτική, δηλαδή μια συνάρτηση f (παρόμοια με το μοντέλο στην εποπτευόμενη μάθηση) που παίρνει το διάνυσμα χαρακτηριστικών μιας κατάστασης ως είσοδο και εξάγει μια βέλτιστη ενέργεια για να εκτελεστεί σε αυτή την κατάσταση, με τη δράση να είναι βέλτιστη αν μεγιστοποιεί την αναμενόμενη μέση ανταμοιβή (Lison, 2015).

Η διαδικασία ενισχυμένης μάθησης προβαίνει στην επίλυση ενός συγκεκριμένου είδους προβλημάτων όπου η λήψη αποφάσεων είναι διαδοχική και ο σκοπός μακροπρόθεσμος, όπως για παράδειγμα το παιχνίδι, η ρομποτική, η διαχείριση των πόρων ή η εφοδιαστική.

## 2.2 Αλγόριθμοι Κατηγοριοποίησης βασισμένοι σε Δένδρα

### *2.2.1 Δέντρα αποφάσεων*

Ένα δέντρο απόφασης είναι ένα ακυκλικό γράφημα που μπορεί να χρησιμοποιηθεί για τη λήψη αποφάσεων. Σε όλους τους κόμβους διακλάδωσης του γραφήματος, εξετάζεται ένα συγκεκριμένο χαρακτηριστικό j του διανύσματος χαρακτηριστικών και σε περίπτωση που η τιμή του χαρακτηριστικού είναι κάτω από ένα συγκεκριμένο όριο, τότε ακολουθείται ο αριστερός κλάδος, διαφορετικά, ακολουθείται ο δεξιός κλάδος. Καθώς φθάνεται ο τελικός κόμβος (leafnode), αποφασίζεται η τάξη στην οποία ανήκει το παράδειγμα. Ένα δέντρο απόφασης μπορεί να αντληθεί από τα δεδομένα (Quinlan, 1986).

Όπως και στο παρελθόν, υπάρχει μια συλλογή από επισημασμένα παραδείγματα. οι ετικέτες ανήκουν στο σετ {0, 1}. Πρέπει να φτιαχτεί ένα δέντρο αποφάσεων που θα επιτρέπει να προβλεφθεί η κλάση ενός παραδείγματος που δίνεται σε ένα διάνυσμα χαρακτηριστικών. Υπάρχουν διάφορες διαμορφώσεις του αλγορίθμου μάθησης δέντρων αποφάσεων. Θεωρείται μόνο ένας, που ονομάζεται ID3. Το κριτήριο βελτιστοποίησης, σε αυτήν την περίπτωση, είναι η μέση λογαριθμική πιθανότητα:

$$\frac{1}{N}\sum_{i=1}^{N} y_i ln f_{ID3}(\mathbf{x}_i) + (1-y_i)\ln(1-f_{ID3}(\mathbf{x}_i))$$

όπου το $f_{ID3}$ είναι ένα δέντρο απόφασης.

Μέχρι τώρα, μοιάζει πολύ με τον αλγόριθμο logisticregression. Ωστόσο, σε αντίθεση με τον αλγόριθμο μάθησης logisticregression που κατασκευάζει ένα παραμετρικό μοντέλο $f_{w^*, b^*}$ με την εξεύρεση βέλτιστης λύσης στο κριτήριο βελτιστοποίησης, ο αλγόριθμος ID3 το βελτιστοποιεί προσεγγιστικά κατασκευάζοντας ένα μη παραμετρικό μοντέλο

$$f_{ID3}(\mathbf{x}) \stackrel{\text{def}}{=} \Pr(y=1|\mathbf{x})$$

Ο αλγόριθμος εκμάθησης ID3 λειτουργεί ως εξής. Έστω ότι S είναι ένα σύνολο επισημασμένων παραδειγμάτων και στην αρχή, το δέντρο απόφασης έχει μόνο ένα κόμβο εκκίνησης που περιέχει όλα τα παραδείγματα:$S \stackrel{\text{def}}{=} \{(x_i, y_i)\}_{i=1}^{N}$ . Αρχίζει με ένα σταθερό μοντέλο $f_{ID3}^{S}$ :

$$f_{ID3}^{S} = \frac{1}{|S|}\sum_{(x,y)\in S} y \quad (1)$$

Η πρόβλεψη που δίνεται από το παραπάνω μοντέλο,$f_{ID3}^{S}(x)$ θα είναι η ίδια για κάθε είσοδο x..

Στη συνέχεια, αναζητούνται όλα τα χαρακτηριστικά j = 1,. . . , D και όλα τα όρια t, και διαιρείται το σύνολο S σε δύο υποσύνολα:

$$S_{-} \stackrel{\text{def}}{=} \{(x,y)|(x,y)\in S, x^{(j)} < t\}$$

και

$$S_{+} \stackrel{\text{def}}{=} \{(x,y)|(x,y)\in S, x^{(j)} \geq t\}$$

Τα δύο νέα υποσύνολα θα πάνε σε δύο νέους κόμβους φύλλων και αξιολογείται, για όλα τα πιθανά ζεύγη (j, t), πόσο καλή είναι η διάσπαση με τα κομμάτια S_ και S +. Τέλος, επιλέγονταιοι καλύτερες τέτοιες τιμές (j, t), διαιρείται το S σε S + και S_, σχηματίζονται δύο νέοι κόμβοι φύλλων και συνεχίζεται η διαδικασία αναδρομικώς με τα S + και S_ (ή

τερματίζεται εάν ένας διαχωρισμός δεν δημιουργεί ένα μοντέλο επαρκώς καλύτερο από το τρέχον ). Στον αλγόριθμο ID3, η αξία ενός διαχωρισμού εκτιμάται χρησιμοποιώντας το κριτήριο που ονομάζεται εντροπία. Η έννοια της εντροπίας αποτελεί ένα μέτρο αβεβαιότητας για μια τυχαία μεταβλητή, η οποία θα φτάσει στο μέγιστο όταν όλες οι τιμές των τυχαίων μεταβλητών είναι ισοπίθανες και στο ελάχιστό της όταν η τυχαία μεταβλητή μπορεί να έχει μόνο μία τιμή. Η εντροπία ενός συνόλου παραδειγμάτων S δίνεται από:

$$H(S) = -f_{ID3}^{S} ln f_{ID3}^{S} - (1 - f_{ID3}^{S}) \ln(1 - f_{ID3}^{S})$$

Όταν διαιρείται ένα σύνολο παραδειγμάτων με ένα συγκεκριμένο χαρακτηριστικό j και ένα όριο t, η εντροπία ενός διαχωρισμού, H (S_, S +), είναι απλώς ένα σταθμισμένο άθροισμα δύο εντροπιών:

$$H(S_-, S_+) = \frac{|S_-|}{|S|} H(S_-) + \frac{|S_+|}{|S|} H(S_+)$$

Έτσι, στον αλγόριθμο ID3, σε κάθε βήμα, σε κάθε κόμβο φύλλων, βρίσκεται ένας διαχωρισμός που ελαχιστοποιεί την εντροπία που δίνεται από την ανωτέρω εξίσωση ή γίνεται τερματισμός σε αυτόν τον κόμβο φύλλων.

Ο αλγόριθμος σταματά σε έναν κόμβο φύλλων σε οποιαδήποτε από τις παρακάτω περιπτώσεις:

- Όλα τα παραδείγματα στον κόμβο των φύλλων ταξινομούνται σωστά από το μοντέλο ενός τμήματος (εξίσωση 1).
- Δεν μπορεί να βρεθεί ένα χαρακτηριστικό για περεταίρω διαχωρισμό.
- Η διάσπαση μειώνει την εντροπία λιγότερο από κάποια τιμή "(η τιμή αυτή πρέπει να βρεθεί πειραματικά).
- Το δέντρο φτάνει σε κάποιο μέγιστο βάθος d (πρέπει επίσης να βρεθεί πειραματικά).

Επειδή στο ID3, η απόφαση να χωριστεί το σύνολο δεδομένων σε κάθε επανάληψη είναι τοπική (δεν εξαρτάται από μελλοντικούς διαχωρισμούς), ο αλγόριθμος δεν εγγυάται τη βέλτιστη λύση. Το μοντέλο μπορεί να βελτιωθεί με τη χρήση τεχνικών όπως η αναζήτηση προς τα πίσω – (backtracking) κατά την αναζήτηση του βέλτιστου δέντρου

αποφάσεων, με το κόστος να χρειαστεί περισσότερο χρόνο για την κατασκευή ενός μοντέλου.

### *2.2.2 Δέντρο Αποφάσεων και Παλινδρόμησης*

Στα δέντρα απόφασης ή παλινδρόμησης, οι εσωτερικοί κόμβοι αντιστοιχούν σε μια δοκιμή σε ένα χαρακτηριστικό, ενώ τα φύλλα περιέχουν πρόβλεψη για ταξινόμηση ή παλινδρόμηση. Για να οικοδομηθεί ένα δέντρο, οι αλγόριθμοι επαγωγής των δέντρων αποφάσεων επαναλαμβάνονται μέσω κάθε χαρακτηριστικού και υπολογίζουν μια θεωρητική συνάρτηση πληροφορίας όπως η εντροπία ή ο δείκτης Gini για την ταξινόμηση και τη διακύμανση για την παλινδρόμηση, (Pang-Ning et al, 2006).

Οι ηλεκτρονικοί αλγόριθμοι επαγωγής δέντρων αντιμετωπίζουν τη δυσκολία ότι τα βήματα της αναδρομικής δομής των δένδρων δεν μπορούν να διαβάσουν τα παρελθόντα δεδομένα. Αφότου ληφθεί η απόφαση διαίρεσης, οι αλγόριθμοι batch χωρίζουν το σύνολο των δεδομένων σε κόμβους παιδιού και υπολογίζουν τη θεωρητική συνάρτηση των απαιτούμενων πληροφοριών, ξεχωριστά για κάθε παιδικό κόμβο. Οι ηλεκτρονικοί αλγόριθμοι δεν μπορούν να διαμοιράσουν τα παρελθόντα δεδομένα. Αντιθέτως, εκμεταλλεύονται τα δυνητικά άπειρα δεδομένα και κάνουν χρήση των φρέσκων παραδειγμάτων μόνο στους νεοσύστατους κόμβους.

Ένα ακόμη πρόβλημα για την κατασκευή διαδικτυακών δέντρων είναι το ότι πρέπει να γίνει μια απόφαση διάσπασης σε μια δεδομένη χρονική στιγμή, δίχως να αποκαλύπτονται τα μελλοντικά δεδομένα, με μια δημοφιλή λύση να είναι η χρήση του κριτηρίου Hoeffding για στατιστική εγγύηση ότι ο επιλεγμένος διαχωρισμός είναι βέλτιστος και για τα μελλοντικά δεδομένα. Στο επονομαζόμενο δέντρο Hoeffding Tree ή Very Fast Decision Tree (VFDT), (Domingos & Hulten, 2000), έχουν αποδοθεί πληροφορίες που οι θεωρητικές λειτουργίες διατηρούν πάνω από το ρεύμα, ενώ αν πληρείται το κριτήριο Hoeffding, γίνεται μια απόφαση διάσπασης και τα στατιστικά στοιχεία χαρακτηριστικών στους νέους κόμβους παιδιών υπολογίζονται βάσει των νέων δεδομένων από τη ροή. Παρόμοιες μέθοδοι για δέντρα παλινδρόμησης περιγράφουν οι Alberg et al, (2012). Για online κατακόρυφα παράλληλα κατανεμημένα δέντρα Hoeffding, βλ. (Kourtellis et al, 2016). Ένα πρόβλημα στην κατασκευή δέντρων αποφάσεων σε δεδομένα ροής είναι το κόστος διατήρησης στατιστικών χαρακτηριστικών για πολλά χαρακτηριστικά. Για τέτοιου είδους ιδιότητες, πρέπει να

διατηρείται ένα ιστόγραμμα χαμηλής κοκκοποίησης, με την εργασία των Jin & Agrawal, (2003), να περιγράφει μια μέθοδο που χωρίζει το εύρος ενός αριθμητικού χαρακτηριστικού σε διαστήματα και χρησιμοποιεί στατιστικές δοκιμές για να κλαδεύει αυτά τα διαστήματα.

### *2.2.3 Τυχαία Δάση*

Υπάρχουν δύο παραδείγματα εκμάθησης του συνόλου: bagging και boosting. Το bagging αποτελείται από τη δημιουργία πολλών "αντιγράφων" των δεδομένων εκπαίδευσης (κάθε αντίγραφο είναι ελαφρώς διαφορετικό από το άλλο) και στη συνέχεια εφαρμόζεται «weaklearner» συνάρτηση σε κάθε αντίγραφο για να αποκτηθούν πολλαπλά «weak» μοντέλα όπου στη συνέχεια συνδυάζονται. Το παράδειγμα bagging βρίσκεται πίσω από τον αλγόριθμο μάθησηςrandomforest (Breiman, 2001).

Ο baggingαλγόριθμος "vanilla" λειτουργεί ως εξής. Με βάση ένα σετ εκπαίδευσης, δημιουργούνταιΒτυχαία δείγματα $S_b$ (για κάθε b = 1, ..., B) του σετ εκπαίδευσης και χτίζεται ένα μοντέλο δέντρων αποφάσεων $f_b$ χρησιμοποιώντας κάθε δείγμα $S_b$ ως το σετ εκπαίδευσης. Για να παρθούν τα$S_b$ για μερικά b, γίνεται δειγματοληψία με αντικατάσταση. Αυτό σημαίνει ότι γίνεται η εκκίνηση με ένα άδειο σετ και στη συνέχεια επιλέγεται τυχαία ένα παράδειγμα από το σετ εκπαίδευσης και τοποθετείται το ακριβές αντίγραφό του στο$S_b$ διατηρώντας το αρχικό παράδειγμα στο αρχικό σετ εκπαίδευσης. Λαμβάνονται τυχαία τα παραδείγματα μέχρι το | Sb | = N.

Μετά την εκπαίδευση, έχουμε B δέντρα αποφάσεων. Η πρόβλεψη για ένα νέο παράδειγμα x λαμβάνεται ως ο μέσος όρος των B προβλέψεων:

$$y \leftarrow \hat{f}(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{B}\sum_{b=1}^{B} f_b(\mathbf{x})$$

σε περίπτωση παλινδρόμησης, ή με τη λήψη της πλειοψηφίας στην περίπτωση ταξινόμησης.

Ο αλγόριθμος randomforest είναι διαφορετικός από τον αλγόριθμο vanillabagging με έναν μόνο τρόπο. Χρησιμοποιεί έναν τροποποιημένο αλγόριθμο εκμάθησης δέντρων που επιθεωρεί, σε κάθε διαχωρισμό στη διαδικασία εκμάθησης, ένα τυχαίο υποσύνολο των χαρακτηριστικών. Ο λόγος για να γίνει αυτό είναι να αποφευχθεί η συσχέτιση των δένδρων: εάν ένα ή λίγα χαρακτηριστικά είναι πολύ ισχυροί παράγοντες πρόβλεψης για τον στόχο, αυτά τα χαρακτηριστικά θα επιλεγούν για να χωρίσουν τα παραδείγματα σε

πολλά δέντρα. Αυτό θα είχε ως αποτέλεσμα πολλά συσχετισμένα δέντρα στο «δάσος». Οι συσχετισμένοι προγνωστικοί παράγοντες δεν μπορούν να βοηθήσουν στη βελτίωση της ακρίβειας της πρόβλεψης. Ο κύριος λόγος για την καλύτερη απόδοση του μοντέλου είναι ότι τα μοντέλα που είναι καλά θα συμφωνήσουν πιθανώς στην ίδια πρόβλεψη, ενώ τα κακά μοντέλα πιθανότατα διαφωνούν σε διαφορετικέςπροβλέψεις. Η συσχέτιση θα κάνει τα κακά μοντέλα πιο πιθανό να συμφωνήσουν, γεγονός που θα παρεμποδίσει την πλειοψηφία ή τον μέσο όρο.

Οι πιο σημαντικοί υπερπαράμετροι για να ρυθμιστούν είναι ο αριθμός των δέντρων, B, και το μέγεθος του τυχαίου υποσυστήματος των χαρακτηριστικών που πρέπει να ληφθούν υπόψη σε κάθε διάσπαση.

Ο αλγόριθμος randomforest είναι ένας από τους πιο ευρέως χρησιμοποιούμενους αλγόριθμους μάθησης του συνόλου. Είναι πολύ αποτελεσματικός γιατί με τη χρήση πολλαπλών δειγμάτων του αρχικού συνόλου δεδομένων, μειώνεται η διακύμανση του τελικού μοντέλου, λαμβάνοντας ότι η χαμηλή διακύμανση σημαίνει χαμηλή υπερπροσαρμογή.

Η υπερπροσαρμογή συμβαίνει όταν το μοντέλο προσπαθεί να εξηγήσει μικρές παραλλαγές στο σύνολο δεδομένων, επειδή το σύνολο δεδομένων είναι ένα μικρό δείγμα του πληθυσμού όλων των πιθανών παραδειγμάτων του φαινομένου που γίνεται προσπάθεια να μοντελοποιηθεί. Αν ο τρόπος δειγματοληψίας του εκπαιδευτικού σετ δεν ήταν σωστός, τότε θα μπορούσετο δείγμα να περιέχει κάποια ανεπιθύμητα (αλλά αναπόφευκτα) αντικείμενα: θόρυβο, υπερβολικές τιμές και υπερβολικά ή υποεκπροσωπούμενα παραδείγματα. Δημιουργώντας πολλαπλά τυχαία δείγματα με αντικατάσταση του εκπαιδευτικού σετ, μειώνεται η επίδραση αυτών των αντικειμένων.

### *2.2.4 Ensemble Methods*

Οι ensemble methods δημιουργούν πολλαπλά, δυνητικά διαφορετικά μοντέλα σε δυνητικά διαφορετικά υποσύνολα παρουσιών και χαρακτηριστικών, (Pang-Ning et al, 2006), με το πλέον απλό παράδειγμα να είναι το bagging, όπου πολλά base models εκπαιδεύονται με βάση το sampling with replacement, το οποίο μπορεί να προσομοιωθεί ηλεκτρονικά, (Oza, 2005), ενώ μια ειδική τεχνική του ensemble, ο online αλγόριθμος random forest περιγράφεται στην εργασία των Denil et al, (2013).

Μια ιδιαιτέρως επιτυχημένη τεχνική ensemble είναι το boosting, όπου παράγεται μια ακολουθία από base models ενσωματώνοντας το σφάλμα πρόβλεψης των προηγούμενων μοντέλων κατά την κατασκευή του επόμενου, όπως για παράδειγμα, στον AdaBoost, έναν αλγόριθμο που σχεδιάστηκε για την ηλεκτρονική μάθηση στην πρώτη του περιγραφή, (Freund & Shapire, 1995), όπου ο επόμενος ταξινομητής εκπαιδεύεται από weights με βάση το error function προηγούμενων ταξινομητών. Στο gradient boosting (Chen & Guestrin, 2016), το επόμενο μοντέλο εκπαιδεύεται στο υπόλοιπο, δηλαδή στη διαφορά της τιμής εκπαίδευσης και στη συνεχή προβλεπόμενη τιμή των προηγούμενων ταξινομητών.

Για τον αλγόριθμο online boosting, η διαφορά σε σχέση με το batch boosting είναι ότι η απόδοση ενός βασικού μοντέλου δεν μπορεί να παρατηρηθεί σε ολόκληρο το σετ εκπαίδευσης, παρά μόνο στα παραδείγματα που έχουν δειχθεί πριν, ενώ όπως και με τον online decision tree induction, οι αποφάσεις πρέπει να λαμβάνονται με βάση μόνο μέρος των δεδομένων εκπαίδευσης. Ποικίλοι online boosting αλγόριθμοι περιγράφονται στην εργασία του Oza, (2005), βάσει των ιδεών του parallel boosting μέσω της προσέγγισης, (Fan et al, 1999), ενώ για τα δέντρα με τον αλγόριθμο gradient boosting, (Chen & Guestrin, 2016), μια πρόσφατη, πιο επιτυχημένη μέθοδος ταξινόμησης, η online έκδοση περιγράφεται στην εργασία των Vasiloudis et al, (2017).

### *2.2.5 Μοντέλα Bayes*

Τα Bayesian δίκτυα είναι μία εκ των πρώτων εφαρμογών για ηλεκτρονική μάθηση, (Friedman & Goldszmidt, 1997), με τις πρώτες μεθόδους να χρησιμοποιούν κατά βάση ενημερώσεις μίνι-παρτίδων, (Buntine, 1991). Οι μέθοδοι μάθησης Bayesian διατηρούν τους πιθανούς πίνακες πιθανοτήτων P (x|y) με την καταμέτρηση, όπου το x είναι ένα διάνυσμα χαρακτηριστικών και το y είναι η label του. Έχοντας κατά νου τους πίνακες πιθανότητας και την κατανομή της τάξης ως προαπαιτούμενα, η προβλεπόμενη τάξη μιας στιγμής θα είναι αυτή που προβαίνει στη μεγιστοποίηση της posterior πιθανότητας που υπολογίζεται από τον κανόνα Bayes, (Pang-Ning et al, 2006).

Το απλούστερο, Naive Bayes μοντέλο κάνει την "αφηρημένη" παραδοχή ότι κάθε μεταβλητή εισόδου είναι υπό όρους ανεξάρτητη δεδομένης της labelς κλάσης, (Duda et al, 2012). Ενώ το εν λόγω μοντέλο λειτουργεί πάρα πολύ καλά, (Domingos & Pazzani, 1997), μια ασθενέστερη υπόθεση είναι απαραίτητη στα Bayesian δίκτυα (Pearl, 2014), στην οποία εκπροσωπείται κάθε μεταβλητή με έναν κόμβο σε ένα κατευθυνόμενο

ακυκλικό γράφημα, αντί δηλαδή να υποτίθεται μία γενική ανεξαρτησία, υποτίθεται ότι κάθε μεταβλητή είναι υπό όρους ανεξάρτητη δεδομένων των γονέων της στο γράφημα.
Για τη διαδικασία της εκμάθησης μέσα από το διαδίκτυο, είναι εύκολο να ενημερωθούν οι υπό συνθήκη πιθανότητες τόσο για τα Naive Bayes όσο και για τα Bayesian δίκτυα, (Friedman et al, 1997), εφόσον αυτά ταιριάζουν στην εσωτερική μνήμη, με τις μεθόδους ενημέρωσης της δομής του δικτύου στο διαδίκτυο να περιγράφονται, για παράδειγμα, στην εργασία των Friedman et al, (1997).

## 2.3 Online Machine Learning for Big Data Streams

Πριν από μερικά χρόνια, τα γρήγορα δεδομένα, (Lam et al, 2012), αντικατόπτριζαν την ιδέα ότι τα data streams δημιουργούνται με πολύ υψηλούς ρυθμούς και ότι αυτά πρέπει να αναλυθούν γρήγορα για να φθάσουν σε ευαίσθητες πληροφορίες, ενώ μπορούν να έρχονται από μετρήσεις δικτύου, εγγραφές κλήσεων, επισκέψεις σε ιστοσελίδες, αναγνώσεις αισθητήρων κ.ο.κ., (Bifet et al, 2011). Το γεγονός ότι τα δεδομένα αυτά φθάνουν συνεχώς σε πολλαπλές, ταχείες, χρονικά μεταβαλλόμενες, πιθανώς απρόβλεπτες και απεριόριστες ροές φαίνεται να αποδίδει ορισμένα θεμελιωδώς νέα ερευνητικά προβλήματα, με παραδείγματα τέτοιων εφαρμογών να είναι οι οικονομικές εφαρμογές, (Zhu & Shasha, 2002), η παρακολούθηση του δικτύου, (Babu & Widom, 2001), η ασφάλεια, τα δίκτυα αισθητήρων, (Morales et al, 2016), η ανάλυση Twitter, (Bifet & Frank, 2010), κτλ (Fontenla-Romero et al, 2013).
Η διαδικασία παραδοσιακής επεξεργασίας δεδομένων προϋποθέτει ότι τα δεδομένα είναι διαθέσιμα για πολλαπλή πρόσβαση, ακόμη και αν σε ορισμένες περιπτώσεις βρίσκονται σε δίσκο και μπορούν να επεξεργαστούν μόνο σε μεγαλύτερα κομμάτια, όντας σε κατάσταση ηρεμίας και μπορώντας να εκτελέσουν batch processing. Τα συστήματα βάσεων δεδομένων, για παράδειγμα, αποθηκεύουν μεγάλες συλλογές δεδομένων και επιτρέπουν στους χρήστες να κάνουν ερωτήματα και συναλλαγές, ενώ τα γρήγορα δεδομένα ή τα δεδομένα σε κίνηση είναι στενά συνδεδεμένα και σε ορισμένες περιπτώσεις χρησιμοποιούνται ως συνώνυμο του υπολογιστικού μοντέλου ροής δεδομένων, (Muthukrishnan et al, 2005). Στο εν λόγω μοντέλο, τα δεδομένα φθάνουν συνεχώς σε ένα δυνητικά άπειρο ρεύμα που πρέπει να επεξεργαστεί από ένα σύστημα με περιορισμένο πόρο. Ο κύριος περιορισμός είναι ότι η κύρια μνήμη είναι μικρή και μπορεί να περιέχει μόνο ένα μικρό τμήμα του ρεύματος, επομένως τα περισσότερα δεδομένα πρέπει να απορρίπτονται αμέσως μετά την επεξεργασία.

Σε μία από τις πρώτες εργασίες που περιγράφουν ένα σύστημα επεξεργασίας ροής δεδομένων, (Abadi et al, 2003), περιγράφονται οι ανάγκες των εφαρμογών παρακολούθησης. Οι εργασίες που σχετίζονται με τις εφαρμογές παρακολούθησης διαφέρουν από τη συμβατική επεξεργασία δεδομένων σε κατάσταση ηρεμίας, καθώς το σύστημα λογισμικού πρέπει να επεξεργάζεται και να αντιδρά σε συνεχείς εισροές από πολλαπλές πηγές. Οι Benczur et al, (2018), εισάγουν το ενεργό για δεδομένα παθητικό για ανθρώπους μοντέλο, στο οποίο το σύστημα επεξεργάζεται μόνιμα δεδομένα για την παροχή ειδοποιήσεων για τον άνθρωπο. Οι ανάγκες και οι ευκαιρίες για μηχανική μάθηση σε ταχείες ροές δεδομένων υποκινούνται από έναν ταχέως αυξανόμενο αριθμό βιομηχανικών εφαρμογών, εφαρμογών συναλλαγών, αισθητήρων και άλλων εφαρμογών, (Zliobaite et al, 2012). Η ηλεκτρονική μηχανική μάθηση συνοψίζεται σε μια εκ των πρώτων επισκοπήσεων του πεδίου (Widmer & Kubat, 1996).

Κύριο καθήκον αποτελεί η διαδικασία εκμάθησης μιας ιδέας διαδοχικά με επεξεργασία των επισημασμένων παραδειγμάτων training μία φορά σε κάθε περίπτωση. Μετά από κάθε εμφάνιση δεδομένων, μπορεί να ενημερωθεί το μοντέλο, και μετά απορρίπτεται το παράδειγμα. Στα υπολογιστικά μοντέλα ροής δεδομένων, μόνο ένα μικρό μέρος των δεδομένων μπορεί να διατηρηθεί διαθέσιμο για άμεση ανάλυση, (Henzinger et al, 1998), γεγονός που έχει τόσο αλγοριθμικές όσο και στατιστικές συνέπειες για την εκμάθηση μηχανών: Οι μη βέλτιστες αποφάσεις σε προηγούμενα μέρη των δεδομένων μπορεί να είναι δύσκολο να ανακληθούν και αν χρειαστεί, απαιτούν διαδικασίες δειγματοληψίας και σύνοψης χαμηλής μνήμης, ενώ για τις εφαρμογές ροής δεδομένων, η βαθμιδωτή ή η ηλεκτρονική μάθηση ταιριάζει καλύτερα.

### *2.3.1 Ενημέρωση Μοντέλου Διαδικτυακής Μηχανικής Μάθησης*

Η διαδικτυακή μάθηση ενημερώνει το μοντέλο της μετά από κάθε εμφάνιση δεδομένων χωρίς πρόσβαση σε όλα τα δεδομένα του παρελθόντος, εξ ου και ισχύουν οι περιορισμοί του υπολογιστικού μοντέλου ροής δεδομένων. Η ροή δεδομένων δεν είναι απλώς ένας τεχνικός περιορισμός στη μηχανική μάθηση, καθώς τα γρήγορα δεδομένα δεν αφορούν μόνο την ισχύ επεξεργασίας αλλά και τη γρήγορη σημασιολογία, ενώ οι μεγάλες βάσεις δεδομένων που είναι διαθέσιμες για την εξόρυξη στις μέρες μας έχουν συγκεντρωθεί σε μήνες ή χρόνια και οι υποκείμενες διαδικασίες που τις δημιουργούν έχουν αλλάξει κατά τη διάρκεια αυτής της περιόδου, μερικές φορές ριζικά (Hulten et al, 2001). Σε εργασίες ανάλυσης δεδομένων, οι θεμελιώδεις ιδιότητες των δεδομένων μπορεί να αλλάξουν

γρήγορα, γεγονός που καθιστά τις διαδικασίες σταδιακής χειροκίνητης ρύθμισης μοντέλων μη αποτελεσματικές και μάλιστα μη εφικτές, (Zliobaite et al, 2012). Ο παραδοσιακός αλγόριθμος μάθησης batch κατασκευάζει στατικά μοντέλα από πεπερασμένα, στατικά, πανομοιότυπα κατανεμημένα σύνολα δεδομένων. Αντίθετα, οι αλγόριθμοι μάθησης ροής πρέπει να δημιουργήσουν μοντέλα που εξελίσσονται με την πάροδο του χρόνου. Η διαδικασία της επεξεργασίας εξαρτάται εν πολλοίς από τη σειρά παραδειγμάτων που παράγονται από συνεχή, μη στατική ροή δεδομένων κι έτσι, η μοντελοποίηση δέχεται επιρροές από ενδεχόμενες μετακινήσεις ιδεών ή από αλλαγές στη διανομή, (Gama et al, 2013).

### *2.3.2 Adaptive Machine Learning Models*

Η adaptive learning επηρεάζει επίσης τον τρόπο με τον οποίο εκτελείται η αξιολόγηση. Ενδεχομένως σε κάθε μονάδα χρόνου, το σύστημα μπορεί να επιστρέψει προβλέψεις από διαφορετικά μοντέλα και ενδέχεται να ληφθούν πολύ λίγες προβλέψεις από ένα συγκεκριμένο μοντέλο για αξιολόγηση από παραδοσιακές μετρήσεις. Αντιθέτως, πρέπει να οριστούν μέτρα σφάλματος που μπορούν να ελαχιστοποιηθούν σε ένα σύστημα ανάδρασης, με τις προβλέψεις να γίνονται για μια ροή αντικειμένων ένα προς ένα και η σωστή απάντηση να λαμβάνεται αμέσως μετά, ενώ μία απόκλιση ανάμεσα στην πρόβλεψη και την παρατηρούμενη τιμή χρησιμεύει ως ανατροφοδότηση, η οποία μπορεί ευθύς να τροποποιήσει το μοντέλο (Widmer & Kubat, 1996).

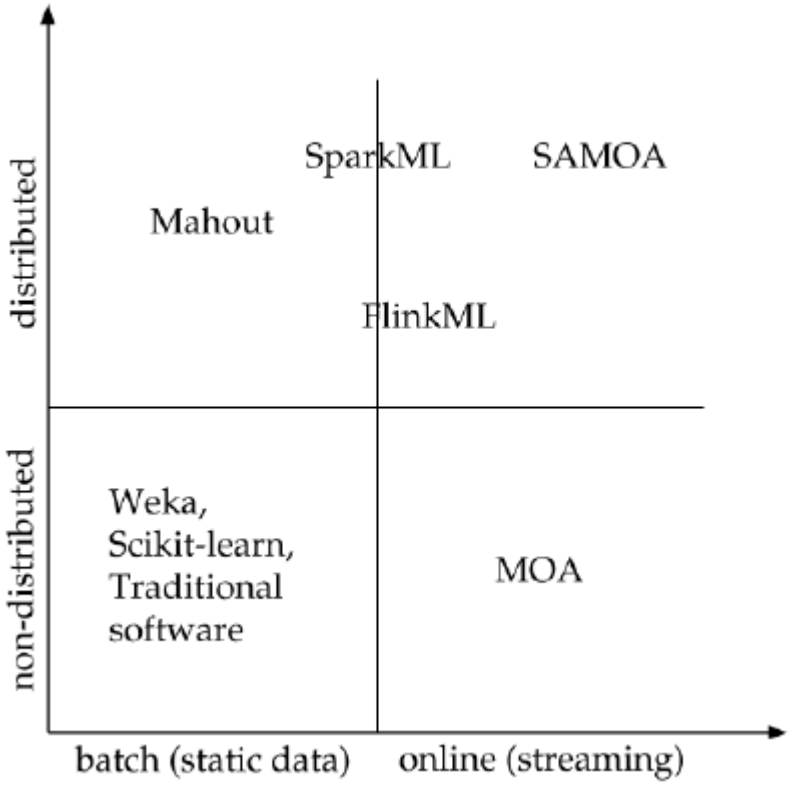
Τέλος, η τρίτη σημαντική πτυχή της διαδικτυακής μάθησης από τα μεγάλα δεδομένα είναι ο αλγόριθμος και προκειμένου να αντιμετωπιστεί ο όγκος των δεδομένων, η επεξεργασία πρέπει να διανεμηθεί, με τους machine clusters να είναι δύσκολο να τα διαχειριστούν και η αποτυχία υλικού να πρέπει να μετριαστεί στην περίπτωση μιας εφαρμογής που εκτελείται σε χιλιάδες διακομιστές. Το Map-Reduce, (Dean et al, 2008), ήταν η πρώτη προσπάθεια προγραμματισμού αφαίρεσης που σχεδιάστηκε για να διαχειρίζεται το cluster, να παρέχει ανεκτικότητα σφαλμάτων και να διευκολύνει την ανάπτυξη λογισμικού και την αποσφαλμάτωση. Αν και το Map-Reduce έχει σχεδιαστεί για batch processing, η διαδικασία επεξεργασία ροής κατανεμημένων δεδομένων απαιτεί άλλες λύσεις, όπως η επικοινωνία ανάμεσα στα στοιχεία επεξεργασίας, (Neumeyer et al, 2010), μέσα από μία τοπολογία διασύνδεσης (Toshniwal et al, 2014). Για μια προοπτική, οι περισσότερες λύσεις εξόρυξης δεδομένων κατά batches αναλύονται από τους Fan & Bifet, (2013).

### *2.3.3 Διαδικτυακή Μηχανική Μάθηση μέσω Κατανεμημένων Ροών Μεγάλων Δεδομένων*

Πολλές έρευνες για την ηλεκτρονική εκμάθηση μηχανών γενικά, (Gaber et al, 2007), και τα υποπεδία, (Widmer & Kubat, 1996), εμφανίστηκαν πρόσφατα. Η μελέτη των Benczur et al, (2018), είναι διαφορετική, καθώς εστιάζει σε τρεις πτυχές: το υπολογιστικό μοντέλο ροής δεδομένων, τις προσαρμοστικές μεθόδους για τη διαχείριση της αντίληψης ιδεών και τις λύσεις κατανεμημένης αρχιτεκτονικής λογισμικού για streaming και επεξεργάζεται συστήματα για μηχανική εκμάθηση με επεξεργασία κατανεμημένων ροών δεδομένων. Συγκεκριμένα, οι Benczur et al, (2018), διερευνούν την ιδέα της χρήσης παραμέτρων Servers. Εστιάζουν σε μη στάσιμα περιβάλλοντα και δεν δίνουν θεωρήματα σύγκλισης για απόδοση σε identically distributed streams.

Σύμφωνα με τους Benczur et al, (2018), η έρευνα τους είναι η πρώτη έρευνα σχετικά με την ηλεκτρονική εκμάθηση μηχανών με μια εκτενή συζήτηση για τα recommender systems. Οι recommenders είναι σημαντικοί δεδομένου ότι παρέχουν ένα σαφές, σχετικό με τη βιομηχανία παράδειγμα της Απαίτησης 2. Σημειώνεται ότι οι Zlobaite et al, (2012), παρατηρούν ότι τα adaptive learning models εξακολουθούν να χρησιμοποιούνται σπάνια στη βιομηχανία.

### *2.3.4 Ταξινόμηση Εργαλείων Διαδικτυακής Μηχανικής Μάθησης*




**Σχήμα 1.** Ταξινόμηση εργαλείων μηχανικής μάθησης

Στην εργασία των Benczur et al, (2018), εξετάζονται εργαλεία ηλεκτρονικής μάθησης για μεγάλα δεδομένα, με την κύρια εστίασή τους να είναι τα μοντέλα, οι αρχιτεκτονικές και οι βιβλιοθήκες λογισμικού που επεξεργάζονται stream data through shared-nothing distributed architectures. Όπως φαίνεται στο σχήμα 1, τα δύο βασικά διακριτικά χαρακτηριστικά των εργαλείων μηχανικής μάθησης είναι το κατά πόσο είναι κατανεμημένα και το αν βασίζονται σε στατικά δεδομένα ή σε δεδομένα συνεχούς ροής.

Ως το λιγότερο περιοριστικό των τεσσάρων τεταρτημορίων στο σχήμα 1, τα μη κατανεμημένα εργαλεία παρτίδων μπορούν να εφαρμόσουν οποιαδήποτε μέθοδο από τα άλλα τεταρτημόρια, ενώ αν το επιτρέπει η κλίμακα, τα data streams μπορούν πρώτα να αποθηκευτούν και στη συνέχεια να αναλυθούν σε ηρεμία και τα διανεμημένα βήματα επεξεργασίας μπορούν να ξεδιπλωθούν για να τρέξουν διαδοχικά σε ένα σύστημα ενός επεξεργαστή, με τα παραδοσιακά εργαλεία εκμάθησης μηχανών, για παράδειγμα, R, Weka και scikit-learn, να εμπίπτουν σε αυτό το τεταρτημόριο.

Τα κατανεμημένα συστήματα μηχανικής μάθησης batch, (Fan & Bifet, 2013), συνήθως εφαρμόζουν αλγόριθμους χρησιμοποιώντας την αρχή Map-Reduce, (Dean et al, 2008),

ενώ το πιο γνωστό σύστημα είναι το Mahout, (Owen et al, 2011), χτισμένο στην κορυφή του Hadoop, (White, 2010), αλλά με μια πολύ πλούσια ποικιλία λύσεων. Το GraphLab, (Low et al, 2012), που αποσύρθηκε από την αγορά το 2016, αποτέλεσε ένα άλλο εργαλείο που ήταν κυρίως εργαλείο batch και επηρέασε επίσης τα ηλεκτρονικά συστήματα.

Οι λύσεις ηλεκτρονικής μάθησης καλύπτουν αλγορίθμους που δημιουργούν αμέσως μοντέλα αφού δουν ένα σχετικά μικρό τμήμα των δεδομένων, ενώ με αυτήν την απαίτηση, αντιμετωπίζεται η δυσκολία να μην είναι αναγκαστικά σε θέση να ακυρωθεί μια υποβέλτιστη απόφαση που έγινε σε προγενέστερο στάδιο, με βάση δεδομένα που δεν είναι πλέον διαθέσιμα για τον αλγόριθμο. Η πρώτη βιβλιοθήκη για την μηχανική εκμάθηση στο διαδίκτυο, το MOA, (Bifet et al, 2010), συλλέγει μια ποικιλία μοντέλων κατάλληλων για online εκπαίδευση και παράλληλα, με βάση τις έννοιες του MOA, το SAMOA, (Morales & Bifet, 2015), είναι ένα κατανεμημένο πλαίσιο - μια ειδικού σκοπού DSPE και βιβλιοθήκη που παρέχει κατανεμημένη εφαρμογή για τους περισσότερους αλγορίθμους MOA. Τα recommender systems μάθησης μέσω διαδικτύου, τόσο διανεμημένα όσο και μη διανεμημένα, περιγράφονται επίσης στους Palovics et al, (2017), ενώ ένα άλλο πρόσφατο μη διανεμημένο ηλεκτρονικό εκπαιδευτικό εργαλείο περιγράφεται στους Kavitha & Punithavalli, (2010).

Για μια κατανεμημένη online αρχιτεκτονική λογισμικού εκμάθησης στο διαδίκτυο, το υποκείμενο σύστημα πρέπει να είναι ένα DSPE και εκτός από το SAMOA, το οποίο μπορεί να θεωρηθεί ως το ίδιο σαν DSPE, τα περισσότερα DSPE της προηγούμενης ενότητας μπορούν να χρησιμοποιηθούν για την κατανεμημένη ηλεκτρονική μάθηση, όπως για παράδειγμα, τα Flink, Samza και Storm που εφαρμόζουν διεπαφές για τη χρήση βιβλιοθηκών SAMOA. Ωστόσο, ένα DSPE μπορεί επίσης να εφαρμόσει αλγόριθμους μηχανικής μάθησης batch:

Για παράδειγμα, η βιβλιοθήκη μηχανικής εκμάθησης SparkML είναι ως επί το πλείστον batch και το FlinkML είναι εν μέρει batch, με τις συνδυασμένες λύσεις batch και ηλεκτρονικής εκμάθησης μηχανών να έχουν μεγάλη πρακτική σημασία. Μπορούν να εκπαιδευθούν τα μοντέλα με βάση ένα πρόχειρο δείγμα και στη συνέχεια να εφαρμοστεί πρόβλεψη για μια ζωντανή ροή δεδομένων, με τις λύσεις DSPE προχωρούν σε αυτόν τον τομέα. Ένα πρόσφατο αποτέλεσμα, το Clipper είναι ένα σύστημα πρόβλεψης με low latency, (Crankshaw et al, 2017), με ένα abstract layer model που διευκολύνει την

εξυπηρέτηση προ-εκπαιδευμένων μοντέλων βασισμένων σε μια μεγάλη ποικιλία πλαισίων μηχανικής μάθησης.

Μια άλλη προσέγγιση batch για την εκμάθηση από τα δεδομένα που αλλάζουν με το χρόνο είναι η επανειλημμένη εφαρμογή ενός παραδοσιακού εργαλείου μάθησης σε ένα συρόμενο παράθυρο με παραδείγματα: Όπως φτάνουν τα νέα παραδείγματα, εισάγονται στην αρχή του παραθύρου. Εν συνεχεία, αφαιρείται αντίστοιχο πλήθος παραδειγμάτων από το τέλος του παραθύρου και το εργαλείο εφαρμόζεται ξανά, όπως και σε πρώιμες έρευνες σχετικά με τη μετατόπιση ιδεών στη μάθηση από συνεχή δεδομένα, (Widmer & Kubat, 1996) και τέλος, σε συνδυασμό με τις μεθόδους μάθησης μέσω διαδικτύου, το μοντέλο που έχει εκπαιδευτεί σε παρτίδες μπορεί να ενημερωθεί σταδιακά με έναν αλγόριθμο ροής, (Frigo et al, 2017).

### *2.3.5 Ταξινόμηση και Παλινδρόμηση*

Ο στόχος της ταξινόμησης και της παλινδρόμησης είναι η πρόβλεψη της άγνωστης labelς ενός instance δεδομένων με βάση τα χαρακτηριστικά ή τις μεταβλητές, με τη label να είναι διακριτή για ταξινόμηση και συνεχής για παλινδρόμηση και την ταξινόμηση και την παλινδρόμηση σε ρυθμίσεις batch να είναι καλά καθιερωμένες, με πολλά εγχειρίδια διαθέσιμα στην εξόρυξη δεδομένων, (Pang-Ning et al, 2006) και στη μηχανική μάθηση, (Friedman et al, 2001).

Για την ταξινόμηση και την παλινδρόμηση σε δεδομένα συνεχούς ροής, οι δύο πρώτες από τις απαιτήσεις της Εισαγωγής διαδραματίζουν βασικό ρόλο, καθώς με την πρώτη απαίτηση, τα δεδομένα είναι παροδικά, επειδή δεν ταιριάζουν στην κύρια μνήμη, επομένως αποκλείονται αλγόριθμοι που επαναλαμβάνονται σε ολόκληρο το σύνολο δεδομένων πολλές φορές και με τη δεύτερη απαίτηση δεν μπορεί να υποτεθεί ότι τα παραδείγματα είναι ανεξάρτητα, διανέμονται πανομοιότυπα και παράγονται από μια στατική κατανομή, επομένως διαφορετικές προβλέψεις από διαφορετικά μοντέλα πρέπει να εφαρμόζονται και να αξιολογούνται σε διαφορετικούς χρόνους. Είναι πασιφανές πως απαιτούνται ειδικά εργαλεία μοντελοποίησης για την αντιμετώπιση των δύο προκλήσεων και οι γνωστές μέθοδοι αξιολόγησης και σύγκρισης δεν είναι βολικές (Gama et al, 2009).

Μια σημαντική, και ίσως και η πιο παλαιά, εφαρμογή της διαδικτυακής μάθησης είναι η single trial classification EEG, όπου το σύστημα διδάσκεται από την ηλεκτρονική ανατροφοδότηση των πειραματικών υποκειμένων, (Obermaier et al, 2001). Αυτά τα πρώτα πειράματα διαφέρουν από την ως τώρα προσέγγιση στο ότι η απόδοση μετρήθηκε

μόνο στο τέλος των πειραμάτων και δεν πραγματοποιήθηκε συστηματική ανάλυση της απόδοσης του ταξινομητή στο χρόνο.

### *2.3.6 Γραμμικό Μοντέλο Διαδικτυακής Μηχανικής Μάθησης*

Το γραμμικό μοντέλο στην διαδικτυακή μηχανική μάθηση χρονολογείται από τον αλγόριθμο perceptron, (Rosenblatt, 1958). Ο αλγόριθμος εκμάθησης perceptron μαθαίνει την τιμή y του d - διαστάσεων διανύσματος εισόδου x ως γραμμικό συνδυασμό των συντεταγμένων,

$$\hat{y} = w \cdot x$$

Ο μηχανισμός πρόβλεψης βασίζεται σε ένα n-διαστάσεων hyperplane της κατεύθυνσης w, το οποίο χωρίζει τον χώρο παρουσίας σε δύο μισά διαστήματα. Το margin ενός παραδείγματος, $\mathbf{y \cdot w \cdot x}$, αποτελεί μια προσημασμένη τιμή ανάλογη με την απόσταση μεταξύ του instance και του υπερεπιπέδου, ενώ αν το margin είναι θετικό, η περίπτωση είναι σωστά ταξινομημένη. διαφορετικά, δεν έχει ταξινομηθεί σωστά.

Το perceptron μπορεί να εκπαιδευτεί με τον αλγόριθμο gradient descent με την hinge loss ως objective function. Αν οι ετικέτες y παίρνουν τις τιμές $\mp 1$ και η πρόβλεψη y ορίζεται από την εξίσωση (2), η hinge loss είναι ίση με

$$l(\boldsymbol{w};(\boldsymbol{x},y)) = \begin{cases} 0 & \alpha\nu\ y \cdot \hat{y} \geq 1 \\ 1 - y \cdot \hat{y} & \alpha\lambda\lambda\iota\acute{\omega}\varsigma \end{cases}$$

από την οποία η κλίση μπορεί να υπολογιστεί ως εξής: Εάν η πρόβλεψη $\hat{\mathbf{y}}$ έχει το σωστό πρόσημο και η απόλυτη τιμή του είναι τουλάχιστον 1, δηλαδή η απώλεια του δεσμού είναι 0, τότε δεν υπάρχει αλλαγή, ο αλγόριθμος είναι παθητικός, διαφορετικά, η κλίση για το w είναι $-\mathbf{x} \cdot \hat{\mathbf{y}}$, και ο κανόνας ενημέρωσης για το ρυθμό εκμάθησης n είναι

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{n} \cdot \boldsymbol{x} \cdot \hat{y}\ \alpha\nu\ y \cdot \hat{y} < 1$$

Ο αλγόριθμος online gradient descent εφαρμόζει απλά το παραπάνω βήμα σε παραδείγματα εκπαίδευσης με σειρά, (Langford & Zhang, 2009), αντιθέτως, ο αλγόριθμος batch gradient descent διαβάζει την είσοδο πολλές φορές και συνήθως βελτιστοποιεί επανειλημμένα τους συντελεστές για την ίδια περίπτωση. Ο αλγόριθμος Batch gradient descent μπορεί να εξομοιωθεί με έναν online αλγόριθμο:

Πραγματοποιούνται τα παραδείγματα trainings ένα προς ένα με τρόπο online και επαναλαμβάνονται πολλές φορές τα δεδομένα εκπαίδευσης, (Cesa-Bianchi & Gentile, 2008).

Με βάση τη γενική ιδέα της εκμάθησης perceptron, έχουν προταθεί αρκετά online γραμμικά μοντέλα, για λεπτομερή επισκόπηση, βλ. (Fontenla=Romero et al, 2013), με τον ταξινομητή Passive Aggressive (PA), (Crammer et al, 2006), να είναι ένα δημοφιλές online γραμμικό μοντέλο που λειτουργεί καλά στην πράξη, για παράδειγμα, εφαρμόζεται ως φίλτρο ανεπιθύμητης αλληλογραφίας Gmail, (Aberdeen et al, 2010). Ο κύριος σκοπός του αλγόριθμου PA είναι η βελτίωση των ιδιοτήτων σύγκλισης του αλγορίθμου perceptron, που αποτελεί την απλούστερη διαδικασία αριθμητικής βελτιστοποίησης. Ορισμένες βελτιωμένες διαδικασίες βελτιστοποίησης στο διαδίκτυο προτάθηκαν πριν από την PA .: Οι Kivinen και Warmuth, (1997), δίνουν μια επισκόπηση πολλών παλαιότερων προσθέτων και πολλαπλασιαστικών ηλεκτρονικών αλγορίθμων, οι οποίοι είναι επιλύσιμοι από τον gradient descent αλγόριθμο, (Li & Yong, 2002). Με βάση τα πειράματα στην εργασία των Crammer & Singer, (2003), ο αλγόριθμος PA ξεπερνά τις περισσότερες από τις προηγούμενες μεθόδους γραμμικής ταξινόμησης σε απευθείας σύνδεση. Η διαδικασία εκπαίδευσης Perceptron και η πλειονότητα των διαδόχων της ισχύουν τόσο για την ταξινόμηση όσο και για την παλινδρόμηση.

### *2.3.7 Μάθηση από Ροές Δεδομένων*

Η ελάχιστη διοχέτευση (pipeline) στη μηχανική μάθηση αποτελείται από: (1) συλλογή και επεξεργασία δεδομένων, (2) εκπαίδευση μοντέλου και (3) ανάπτυξη μοντέλου. Συμβατικά, τα δεδομένα συλλέγονται και υποβάλλονται σε επεξεργασία κατά δεσμίδες. Αν και αυτή η προσέγγιση είναι υπερσύγχρονη σε πολλές εφαρμογές, δεν είναι κατάλληλη στο πλαίσιο των εξελισσόμενων ροών δεδομένων. Η προσέγγιση μάθησης κατά δεσμίδες προϋποθέτει ότι τα δεδομένα είναι επαρκώς μεγάλα και προσβάσιμα. Αυτό δεν συμβαίνει σε ροές δεδομένων καθώς τα δεδομένα είναι διαθέσιμα ένα δείγμα κάθε χρονική στιγμή και η αποθήκευσή τους δεν είναι πρακτική δεδομένης της (θεωρητικά) άπειρης φύσης τους. Επιπλέον, τα μη στατικά περιβάλλοντα απαιτούν την εκτέλεση της διοχέτευσης πολλές φορές προκειμένου να ελαχιστοποιηθεί η υποβάθμιση του μοντέλου, με άλλα λόγια να διατηρηθεί η βέλτιστη απόδοση. Αυτό είναι ιδιαίτερα δύσκολο σε γρήγορα μεταβαλλόμενα περιβάλλοντα όπου η αποδοτική και αποτελεσματική προσαρμογή είναι καίριας σημασίας.

Στην πραγματικότητα, πολλές εφαρμογές μηχανικής μάθησης πραγματικού κόσμου παρουσιάζουν τα χαρακτηριστικά των εξελισσόμενων ροών δεδομένων, ειδικότερα μπορούν να αναφερθούν:

- Οι χρηματοοικονομικές αγορές παράγουν τεράστιους όγκους δεδομένων καθημερινά. Για παράδειγμα, το Χρηματιστήριο Νέας Υόρκης καταγράφει 1 terabyte πληροφοριών κάθε μέρα. Ανάλογα με την κατάσταση αυτών των αγορών και πολλούς εξωτερικούς παράγοντες, τα δεδομένα μπορούν να καταστούν παρωχημένα καθιστώντας τα άχρηστα για τη δημιουργία ακριβών μοντέλων. Τα μοντέλα πρόβλεψης πρέπει να μπορούν να προσαρμόζονται γρήγορα ώστε να είναι χρήσιμα σε αυτό το δυναμικό περιβάλλον.
- Προγνωστική συντήρηση. Η συμβολή του Διαδικτύου των Πραγμάτων (Internet of Things – IoT) στο ψηφιακό σύμπαν είναι σημαντική. Το 2013 τα δεδομένα αποκλειστικά από ενσωματωμένα συστήματα αντιπροσώπευαν το 2% από τα παγκόσμια δεδομένα, ενώ το 2020 το ποσοστό αυτό αναμένεται φτασει στο 10%. Οι αισθητήρες IoT χρησιμοποιούνται για την παρακολούθηση της υγείας πολλών συστημάτων, από σύνθετα συστήματα όπως αεροπλάνα σε απλούστερα, όπως οικιακές συσκευές. Απαιτούνται συστήματα πρόβλεψης με γρήγορη αντίδραση για την αποφυγή διαταραχών από δυσλειτουργικά στοιχεία.
- Διαδικτυακή ανίχνευση απάτης. Η ταχύτητα αντίδρασης ενός αυτόματου συστήματος είναι επίσης ένας σημαντικός παράγοντας σε πολλές εφαρμογές. Για παράδειγμα, η VisaNet έχει την ικανότητα (από τον Ιούνιο του 2019) να χειρίζεται περισσότερες από 65.000 συναλλαγές ανά δευτερόλεπτο. Η ανίχνευση απάτης στις διαδικτυακές τραπεζικές συναλλαγές περιλαμβάνει επιπλέον προκλήσεις εκτός από τη συλλογή και την επεξεργασία δεδομένων. Τα συστήματα ανίχνευσης απάτης πρέπει να προσαρμόζονται γρήγορα σε αλλαγές όπως στη συμπεριφορά των καταναλωτών (για παράδειγμα κατά τη διάρκεια των διακοπών), στη σταθερότητα των χρηματοπιστωτικών αγορών, καθώς και στο γεγονός ότι οι εισβολείς αλλάζουν συνεχώς τη συμπεριφορά τους για να νικήσουν αυτά τα συστήματα.
- Εφοδιαστική αλυσίδα. Αρκετοί τομείς χρησιμοποιούν αυτόματα συστήματα στην αλυσίδα εφοδιασμού τους για να αντιμετωπίσουν αποτελεσματικά τη ζήτηση προϊόντων. Ωστόσο, η πανδημία COVID -19 έφερε στην προσοχή την

ευπάθεια αυτών των συστημάτων σε ξαφνικές αλλαγές, π.χ., σε λιγότερο από 1 εβδομάδα, προϊόντα που σχετίζονταν με την πανδημία όπως μάσκες προσώπου γέμισαν τους 10 κορυφαίους όρους αναζήτησης στο Amazon. Πολλά αυτόματα συστήματα δεν κατάφεραν να ανταπεξέλθουν στις αλλαγές με αποτέλεσμα τη διαταραχή της αλυσίδας εφοδιασμού.

- Η κλιματική αλλαγή. Τα περιβαλλοντολογικά επιστημονικά δεδομένα είναι ένα απόλυτο παράδειγμα των πέντε χαρακτηριστικών των μεγάλων δεδομένων (big data): όγκος, ταχύτητα, ποικιλομορφία, εγκυρότητα και αξία. Συγκεκριμένα, τα έργα Earth Science Data and Information System της NASA, διαθέτει 24 petabytes δεδομένων στο αρχείο του και διένειμε 1.3 δισεκατομμύρια αρχεία το 2017. Η κατανόηση των περιβαλλοντικών δεδομένων έχει πολλές επιπτώσεις στην καθημερινή ζωή, π.χ. η παραγωγή τροφίμων μπορεί να επηρεαστεί σοβαρά από την κλιματική αλλαγή , η διαταραχή του κύκλου του νερού είχε ως αποτέλεσμα την αύξηση των ισχυρών βροχοπτώσεων με τον σχετικό κίνδυνο πλημμυρών. Οι αισθητήρες IoT κάνουν τώρα τα περιβαλλοντικά δεδομένα διαθέσιμα με ταχύτερο ρυθμό και τα συστήματα μηχανικής μάθησης πρέπει να προσαρμοστούν σε αυτόν τον νέο κανόνα.

Όπως φαίνεται στα προηγούμενα παραδείγματα, τα δυναμικά περιβάλλοντα αποτελούν ένα επιπρόσθετο σύνολο προκλήσεων για τα συστήματα μάθησης κατά δεσμίδες. Η υποβάθμιση του μοντέλου είναι ένα κυρίαρχο πρόβλημα σε πολλές εφαρμογές του πραγματικού κόσμου. Καθώς έχουν δημιουργηθεί και συλλεχθεί αρκετά δεδομένα, οι προληπτικοί χρήστες ενδέχεται να αποφασίσουν να εκπαιδεύσουν τα μοντέλα τους για να βεβαιωθούν ότι συμφωνούν με τα τρέχοντα δεδομένα. Αυτό είναι περίπλοκο για δύο λόγους: Πρώτον, τα μοντέλα δεσμίδων (γενικά) δεν μπορούν να λάβουν υπόψη στη χρήση τα νέα δεδομένα, επομένως η διοχέτευση μηχανικής μάθησης πρέπει να εκτελεστεί πολλές φορές καθώς τα δεδομένα συλλέγονται με την πάροδο του χρόνου. Δεύτερον, η απόφαση για μια τέτοια ενέργεια δεν είναι τετριμμένη και περιλαμβάνει πολλαπλές πτυχές. Για παράδειγμα, πρέπει ένα καινούριο μοντέλο να εκπαιδεύεται μόνο σε νέα δεδομένα; Αυτό εξαρτάται από το μέγεθος της διακύμανσης στα δεδομένα. Μικρές διακυμάνσεις μπορεί να μην είναι αρκετές για να δικαιολογήσουν την επανεκπαίδευση και την εκ νέου ανάπτυξη ενός μοντέλου. Αυτός είναι ο λόγος για τον οποίο μια αντιδραστική προσέγγιση χρησιμοποιείται κυρίως στη βιομηχανία. Η

υποβάθμιση του μοντέλου παρακολουθείται και επιβάλλονται διορθωτικά μέτρα σε περίπτωση υπέρβασης ενός ορίου που καθορίζεται από το χρήστη (ακρίβεια, σφάλματα τύπου I και τύπου II κ.λπ.). Μια άλλη σημαντική πτυχή που πρέπει να ληφθεί υπόψη είναι η αμοιβαία ανταλλαγή μεταξύ της επένδυσης σε πόρους όπως η μνήμη και ο χρόνος (και το σχετικό κόστος) και το κέρδος στην προβλεπτική απόδοση. Στην μάθηση ροής, η αποδοτικότητα από άποψη πόρων είναι θεμελιώδης, τα μοντέλα πρόβλεψης όχι μόνο πρέπει να είναι ακριβή, αλλά και να μπορούν να χειρίζονται θεωρητικά άπειρες ροές δεδομένων. Τα μοντέλα πρέπει να χωράνε στη μνήμη, ανεξάρτητα από τον όγκο των δεδομένων που εμφανίζονται (σταθερή μνήμη). Επιπροσθέτως, ο χρόνος εκπαίδευσης προβλέπεται να αυξηθεί γραμμικώς σε συνάρτηση με τον όγκο των δεδομένων που υποβάλλονται σε επεξεργασία. Τα νέα δείγματα πρέπει να επεξεργάζονται μόλις καταστούν διαθέσιμα οπότε είναι ζωτικής σημασίας να γίνει η επεξεργασία όσο το δυνατόν γρηγορότερα ώστε να υπάρχει ετοιμότητα για το αμέσως επόμενο δείγμα στη ροή.

Επισήμως, το έργο της επιτηρούμενης μάθησης από εξελισσόμενες ροές δεδομένων ορίζεται ως εξής: Έστω μια ροή δεδομένων $S = \{(\overrightarrow{x_t}, y_t)\} \mid t = 1, \ldots, T$ με $T \to \infty$. Η είσοδος $\overrightarrow{x_t}$ αποτελεί ένα διάνυσμα χαρακτηριστικών και το $y_t$ ο αντίστοιχος στόχος όπου το $y$ είναι συνεχές σε περίπτωση παλινδρόμησης και διακριτό σε περίπτωση ταξινόμησης, ενώ σκοπός είναι να προβλεφθεί ο στόχος $y$ για ένα άγνωστο δείγμα $\overrightarrow{x}$.

Για επεξηγηματικούς σκοπούς, η εργασία του (Montiel, 2020) επικεντρώνεται μόνο στην ταξινόμηση.

Στην μάθηση ροής, τα μοντέλα εκπαιδεύονται αυξητικά, ένα δείγμα κάθε φορά, καθώς καθίστανται διαθέσιμα νέα δείγματα $(\overrightarrow{x_t}, y_t)$. Δεδομένου ότι οι ροές είναι θεωρητικά άπειρες, η φάση εκπαίδευσης είναι αδιάκοπη και τα μοντέλα πρόβλεψης ενημερώνουν συνεχώς την εσωτερική τους κατάσταση σε συμφωνία με τα εισερχόμενα δεδομένα. Αυτό είναι θεμελιωδώς διαφορετικό από τη προσέγγιση μάθησης κατά δεσμίδες, όπου τα μοντέλα έχουν πρόσβαση σε όλα τα (διαθέσιμα) δεδομένα κατά τη διάρκεια της εκπαίδευσης. Όπως αναφέρθηκε προηγουμένως, στο υπόδειγμα μάθησης ροής, τα μοντέλα πρόβλεψης πρέπει να είναι αποδοτικά από άποψη πόρων. Για το σκοπό αυτό, πρέπει να πληρούται ένα σύνολο απαιτήσεων μέσω των μεθόδων ροής:

- **Επεξεργασία ενός δείγματος κάθε χρονική στιγμή και επιθεώρησή του μόνο μία φορά.** Η υπόθεση είναι ότι δεν υπάρχει αρκετός χώρος ούτε χρόνος

για την αποθήκευση όλων των δειγμάτων, και εάν δεν τήρηθει αυτη απαίτηση υπάρχει ο κίνδυνος να χάνονται, πιθανώς σημαντικά, εισερχόμενα δεδομένα.

- **Χρήση περιορισμένης ποσότητας μνήμης.** Οι ροές δεδομένων θεωρούνται άπειρες, επομένως η αποθήκευση δεδομένων για περαιτέρω επεξεργασία δεν είναι εφικτή.
- **Εργασία σε ορισμένο χρονικό διάστημα.** Με άλλα λόγια, αποφυγή καθυστερήσεων που δημιουργούνται από χρονοβόρες εργασίες που μακροπρόθεσμα θα μπορούσαν να κάνουν τον αλγόριθμο να αποτύχει.
- **Προετοιμασία για πρόβλεψη ανά πάσα στιγμή.** Τα μοντέλα ροής *ενημερώνονται* συνεχώς και πρέπει να είναι έτοιμα να κάνουν πρόβλέψη όποια στιγμή τους ζητηθεί.

#### *2.3.7.1. Concept Drift*

Ένα στοιχείο πρόκληση των δυναμικών περιβαλλόντων είναι οι πιθανότητες ότι η υποκείμενη σχέση μεταξύ των χαρακτηριστικών $X$ και του στόχου $\vec{y}$ μπορεί να εξελιχθεί (αλλάξει) με την πάροδο του χρόνου. Το φαινόμενο αυτό είναι γνωστό ως «Εννοιολογική Απόκλιση» (Concept Drift). Η πραγματική concept drift ορίζεται ως οι αλλαγές στην οπίσθια (posterior) κατανομή των δεδομένων $p(\vec{y} \mid X)$. Η πραγματική concept drift σημαίνει ότι η κατανομή των δεδομένων χωρίς label δεν αλλάζει, ενώ η εξέλιξη των δεδομένων αναφέρεται στην απόλυτη κατανομή δεδομένων $p(X)$. Στην μάθηση κατά δεσμίδες, η από κοινού κατανομή δεδομένων $p(X, \vec{y})$ θεωρείται, γενικά, ότι παραμένει στάσιμη. Στο πλαίσιο των εξελισσόμενων ροών δεδομένων, η concept drift ορίζεται ανάμεσα σε δύο σημεία στον χρόνο $t_0$, $t_1$ ως:

$$p_{t0}(X, \vec{y}) \neq p_{t1}(X, \vec{y})$$

Η concept drift είναι γνωστό πως προκαλεί ζημία στη μάθηση. Τα ακόλουθα μοτίβα συνήθως λαμβάνονται υπόψη:

- **Απότομο (Abrupt)**. Όταν μια νέα έννοια παρουσιάζεται αμέσως, χωρίς να υπάρχει ή με ελάχιστο μεταβατικό στάδιο μεταξύ των εννοιών. Εδώ ο χρόνος προσαρμογής είναι ζωτικής σημασίας μιας και η παλιά έννοια δεν έχει πλέον ισχύ.
- **Αυξητικό (Incremental)**. Εδώ στην διάρκεια της μετάβασης από μια παλιά εννοια σε μια καινούργια, εμφανίζονται ενδιάμεσες έννοιες.

**Βαθμιαίο (Gradual)**. Όταν οι παλιές και νέες έννοιες συμπίπτουν κατά τη μεταβατική περίοδο, γεγονός που μπορεί να είναι δύσκολο, καθώς και οι δύο έννοιες είναι κάπως έγκυρες κατά τη μετάβαση.

- **Επαναλαμβανόμενο (Recurring)**. Εάν μια παλιά έννοια εμφανίζεται ξανά καθώς προχωρά η ροή, υπάρχει δηλαδή μια εποχικότητα, όπως για παράδειγμα στις καλλιέργιες σιτυρών κτλ.
- **Ακραίες τιμές (Outliers)**. Να μην συγχέεται με την πραγματική απόκλιση. Μια μέθοδος ανίχνευσης της απόκλισης πρέπει να είναι εύρωστη στον θόρυβο, με άλλα λόγια, να ελαχιστοποιεί τον αριθμό των ψευδών θετικών υπό την παρουσία ακραίων τιμών ή θορύβου.

Παρόλο που η συνεχής μαθησιακή φύση των μεθόδων ροής δίνει κάποια ευρωστία στην concept drift, έχουν γίνει προτάσεις για εξειδικευμένες μεθόδους για να ανιχνευθεί η απόκλιση. Πολλαπλές μέθοδοι έχουν προταθεί στη βιβλιογραφία, ο (Gama et al., 2014) παρέχει μια διεξοδική έρευνα αυτού του θέματος. Σε γενικές γραμμές, ο στόχος των μεθόδων ανίχνευσης απόκλισης είναι να εντοπίζει με ακρίβεια τις αλλαγές στην κατανομή δεδομένων, δείχνοντας παράλληλα την ευρωστία στο θόρυβο και την αποδοτικότητα από πλευράς πόρων. Οι μέθοδοι με γνώμονα την απόκλιση χρησιμοποιούν εξειδικευμένους μηχανισμούς ανίχνευσης για να αντιδρούν ταχύτερα και αποτελεσματικότερα στην απόκλιση, όπως για παράδειγμα, ο αλγόριθμος *Hoeffding Tree*, ένα είδος δέντρου αποφάσεων για ροές δεδομένων, που δεν χειρίζεται ρητά την concept drift, ενώ ο *Hoeffding Adaptive Tree* χρησιμοποιεί Προσαρμοστική Παραθυροποίηση (ADaptive WINdowing - ADWIN) (Bifet & Gavalda, 2007) για να ανιχνεύει αποκλίσεις. Εάν εντοπιστεί μια απόκλιση σε ένα συγκεκριμένο κλάδο, δημιουργείται ένας εναλλακτικός κλάδος και τελικά αντικαθιστά τον αρχικό κλάδο εάν εμφανίζει καλύτερη απόδοση σε νέα δεδομένα.

Η δημοφιλής μέθοδος ανίχνευσης απόκλισης, ADWIN, με μαθηματικές εγγυήσεις διατηρεί ένα παράθυρο μεταβλητού μήκους πρόσφατων αντικειμένων διασφαλίζοντας ότι δεν υπήρξε καμία αλλαγή στην κατανομή δεδομένων εντός του παραθύρου. Εσωτερικά, δύο δευτερεύοντα παράθυρα ($W_0$, $W_1$) χρησιμοποιούνται για να προσδιορίσουν εάν έχει συμβεί μια αλλαγή. Με κάθε νέο αντικείμενο που παρατηρείται, συγκρίνονται οι μέσες τιμές των αντικειμένων στα $W_0$ και $W_1$ για να επιβεβαιωθεί ότι αντιστοιχούν στην ίδια κατανομή. Εάν δεν ισχύει πλέον η ισότητα κατανομών, τότε στέλνεται ένα σήμα συναγερμού που δείχνει ότι έχει συμβεί απόκλιση. Στη διαδικασία

ανίχνευσης μιας απόκλισης, το $W_0$ αντικαθίσταται από το $W_1$ και αρχικοποιείται ένα νέο $W_1$.

#### *2.3.7.2. Αξιολόγηση Απόδοσης*

Η προβλεπτική απόδοση *P* ενός δοσμένου μοντέλου *h* μετράται συνήθως μέσα από τη χρήση κάποιας συνάρτησης απώλειας *l* η οποία υπολογίζει τη διαφορά ανάμεσα στις προσδοκώμενες (αληθείς) ετικέτες κλάσης *y* και τις προβλεπόμενες ετικέτες κλάσης $\hat{y}$.

$$P(h) = l(y, \hat{y})$$

Μια δημοφιλής και ευθύς συνάρτηση απώλειας για ταξινόμηση είναι η μηδέν-ένα συνάρτηση απώλειας (zero – one loss function) που αντιστοιχεί στην αντίληψη για το εάν το μοντέλο έκανε λάθος ή όχι κατά τη διαδικασία της πρόβλεψης.

$$l(y, \hat{y}) = \begin{cases} 0\,, & y = \hat{y} \\ 1\,, & y \neq \hat{y} \end{cases}$$

Λόγω της αυξητικής φύσης των μεθόδων μάθησης ροής, χρησιμοποιούνται ειδικές εκτιμήσεις για την αξιολόγηση της απόδοσής τους. Δύο επικρατούσες μέθοδοι στη βιβλιογραφία είναι οι *holdout* και *prequential* αξιολογήσεις. Η αξιολόγηση holdout είναι μια δημοφιλής μέθοδος τόσο σε μάθηση ροής όσο και σε κατά δεσμίδες, όπου η δοκιμή πραγματοποιείται σε ένα ανεξάρτητο σύνολο δειγμάτων. Από την άλλη πλευρά, η αξιολόγηση prequential, είναι εξειδικευμένη στις ρυθμίσεις ροής. Στην prequential αξιολόγηση, το μοντέλο τροφοδοτείται από τα νεα δειγματα δεδομένων και δίνει μια πρόβλεψη και στη συνέχεια τα ιδια δεδομένα χρησιμοποιούνται για να εκπαίδευσουν (ενημέρωσουν) το μοντέλο. Το πλεονέκτημα αυτής της προσέγγισης είναι ότι όλα τα δείγματα χρησιμοποιούνται τόσο για δοκιμές όσο και για εκπαίδευση.

Αυτή είναι μια σύντομη επισκόπηση της μηχανικής μάθησης για ροές δεδομένων. Ωστόσο, είναι σημαντικό να αναφερθεί ότι το πεδίο της μηχανικής μάθησης για ροή δεδομένων καλύπτει άλλες εργασίες όπως παλινδρόμηση, συσταδοποίηση, ανίχνευση ανωμαλιών, για να αναφερθούν μερικές. Στο (Gomes et al., 2019) παρέχεται μια εκτενής και βαθύτερη περιγραφή αυτού του πεδίου, της τελευταίας τεχνολογίας και των ενεργών προκλήσεών του.

# 3 Μεθοδολογία

Στο θεωρητικό μέρος της εργασίας πραγματοποιήθηκε η βιβλιογραφική επισκόπηση βασικών εννοιών που σχετίζονται με τη μηχανική μάθηση, τους αλγορίθμους κατηγοριοποίησης και τη διαδικτυακή μηχανική μάθηση σε ροές μεγάλων δεδομένων. Στη συνέχεια της εργασίας παρουσιάζεται το scikit-multiflow, ένα περιβάλλον μηχανικής μάθησης και εξόρυξης γνώσης ανοικτού κώδικα για δεδομένα πολλαπλών εξόδων / πολλαπλών ετικετών και ροών δεδομένων. Το scikit-multiflow είναι υλοποιημένο σε Python και παρουσιάζει αυξανόμενη δημοτικότητα. Συμπληρώνει το scikit-learn, του οποίου η κύρια εστίαση είναι η μάθηση κατά παρτίδες, επεκτείνοντας το σύνολο των εργαλείων εκμάθησης μηχανών σε αυτήν την πλατφόρμα. Στην τρέχουσα κατάσταση του, το scikit-multiflow περιέχει γεννήτριες ροών δεδομένων, κατηγοριοποιητές πολλαπλών εξόδων / πολλαπλών ετικετών δεδομένων ροής, ανιχνευτές αλλαγής και μεθόδους αξιολόγησης.

Σκοπός της μελέτης είναι η διερεύνηση και η σύγκριση υλοποιημένων αλγορίθμων κατηγοριοποίησης του scikit-multiflow πάνω σε ροές δεδομένων. Επιπλέον, θα αναλυθούν αλγόριθμοι απομείωσης δεδομένων σε πραγματικό χρόνο (π.χ., DRHC, AIB2) και θα αποτιμηθεί η απόδοση της κατηγοριοποίησης με ή χωρίς απομείωση των δεδομένων. Η μεθοδολογία που θα ακολουθηθεί για την πραγμάτωση του παραπάνω στόχου είναι η κριτική αποτίμηση των διεθνών βιβλιογραφικών πηγών που σχετίζονται με το εν λόγω ζήτημα. Συγκεκριμένα, αναλύεται η βιβλιοθήκη λογισμικού scikit-multiflow, η αρχιτεκτονική του scikit-multiflow και αναλύεται ένα παράδειγμα εργασίας ταξινόμησης, όπου χρησιμοποιείται η γεννήτρια SEA, η οποία με την κλάση SEAGenerator δημιουργεί δεδομένα που αντιστοιχούν σε ένα πρόβλημα δυαδικής ταξινόμησης, τα οποία χρησιμοποιούνται για την εκπαίδευση ενός ταξινομητή Naive Bayes. Για αυτό το παράδειγμα, θα παραχθεί μια συνθετική ροή δεδομένων και θα γίνει χρήση της Προσαρμοστικής Παραθυροποίησης (ADWIN) ως μεθόδου ανίχνευσης εννοιολογικής απόκλισης, με σκοπό να ανιχνευθεί η απόκλιση που συμβαίνει μετά από συγκεκριμένα δείγματα στη συνθετική ροή δεδομένων. Τέλος, για τη μελέτη της επίδρασης της απόκλισης στη μάθηση συγκρίνονται δύο δημοφιλή μοντέλα ροής, το HoeffdingTreeClassifier και η εκδοχή του HoeffdingAdaptiveTreeClassifier που λαμβάνει υπόψη την απόκλιση.

# 4 Η Βιβλιοθήκη Λογισμικού Scikit-Multiflow

Το scikit-multiflow (Montiel et al., 2018) είναι μια βιβλιοθήκη μηχανικής μάθησης για δεδομένα ροής πολλαπλών εξόδων/πολλαπλών ετικετών γραμμένα σε Python, που αναπτύχθηκε ως ελεύθερο και ανοιχτού κώδικα λογισμικό και διανέμεται με την άδεια BSD 3 - Clause. Ακολουθώντας τη φιλοσοφία του **SciKits**, το scikit-multiflow επεκτείνει το υπάρχον σύνολο εργαλείων για επιστημονικούς σκοπούς. Περιλαμβάνει μια συλλογή από υπερσύγχρονες μεθόδους για ταξινόμηση, παλινδρόμηση, ανίχνευση εννοιολογικής απόκλισης και ανίχνευση ανωμαλιών, μαζί με ένα σύνολο γεννητριών δεδομένων και αξιολογητών. Το scikit-multiflow έχει σχεδιαστεί για να αλληλεπιδρά απρόσκοπτα με το *NumPy* και το *SciPy*. Επιπροσθέτως, συμβάλλει στη διαδικασία εκδημοκρατισμού της μάθησης ροής, μέσα από την αξιοποίηση της δημοτικότητας της γλώσσας Python. Το scikit-multiflow γράφεται κυρίως σε Python και ορισμένα βασικά στοιχεία γράφονται σε *Cython* για απόδοση. Το scikit-multiflow προορίζεται για χρήστες με διαφορετικά επίπεδα εμπειρογνωμοσύνης. Η γέννηση και η ανάπτυξή του ακολουθούν δύο βασικούς στόχους:

1) Ευκολία στο σχεδιασμό και στην εκτέλεση πειραμάτων. Αυτό ακολουθεί την ανάγκη για μια πλατφόρμα που επιτρέπει τη γρήγορη πρωτοτυποποίηση και πειραματισμό. Πολύπλοκα πειράματα μπορούν να εκτελεστούν εύκολα χρησιμοποιώντας κλάσεις αξιολόγησης. Διαφορετικές ροές δεδομένων και μοντέλα μπορούν να αναλυθούν και να συγκριθούν σε πολλαπλές συνθήκες και η ποσότητα του κώδικα που απαιτείται από τον χρήστη διατηρείται στο ελάχιστο.

2) Ευκολία στην επέκταση υφιστάμενων μεθόδων. Οι προχωρημένοι χρήστες μπορούν να δημιουργήσουν νέες δυνατότητες επεκτείνοντας ή τροποποιώντας υπάρχουσες μεθόδους. Με αυτόν τον τρόπο οι χρήστες μπορούν να επικεντρωθούν στις λεπτομέρειες της εργασίας τους και όχι στο πρόσθετο κόστος (overhead) όταν εργάζονται από την αρχή.

Το scikit-multiflow δεν προορίζεται ως αυτόνομη λύση για μηχανική μάθηση. Ενσωματώνεται με άλλες βιβλιοθήκες Python όπως το *Matplotlib* για γραφική απεικόνιση, *scikit-learn* για αυξητική μάθηση συμβατή με τη ρύθμιση ροής, *Pandas* για χειρισμό δεδομένων, *Numpy* και *SciPy* για αριθμητικούς και επιστημονικούς υπολογισμούς. Ωστόσο, είναι σημαντικό να σημειωθεί ότι το scikit-multiflow δεν επεκτείνει το *scikit-learn*, του οποίου η κύρια εστίαση είναι η μάθηση κατά δεσμίδες.

Μια βασική διαφορά είναι ότι οι εκτιμητές στο scikit-multiflow είναι αυξητικοί εκ σχεδιασμού και η εκπαίδευση πραγματοποιείται καλώντας πολλές φορές τη μέθοδο partial_fit(). Η πλειοψηφία των εκτιμητών που υλοποιούνται στο scikit-multiflow είναι αυξητικοί κατά instance (instance - incremental), το οποίο δηλώνει ότι μεμονωμένα στιγμιότυπα χρησιμοποιούνται για την ενημέρωση της εσωτερικής τους κατάστασης. Ένας μικρός αριθμός εκτιμητών είναι αυξητικός κατά δεσμίδες, όπου χρησιμοποιούνται μίνι δεσμίδες δεδομένων. Από την άλλη πλευρά, η κλήση της fit() πολλές φορές σε έναν εκτιμητή scikit-learn θα έχει ως αποτέλεσμα την αντικατάσταση της εσωτερικής του κατάστασης σε κάθε κλήση.

Από την έκδοση 0.5.0, διατίθενται τα ακόλουθα δευτερεύοντα πακέτα:

- anomaly_detection: Μέθοδοι ανίχνευσης ανωμαλιών.
- data: Μέθοδοι ροής δεδομένων, συμπεριλαμβανομένων μεθόδων για μετατροπή δεσμίδας - σε - ροή και γεννήτριες.
- drift_detection: Μέθοδοι για την ανίχνευση εννοιολογικής απόκλισης.
- evaluation: Μέθοδοι αξιολόγησης για μάθηση ροής.
- lazy: Μέθοδοι στις οποίες η γενίκευση των δεδομένων εκπαίδευσης καθυστερεί έως ότου ληφθεί ένα ερώτημα, π.χ. μέθοδοι βάσει γειτόνων, όπως το kNN.
- meta: Μέθοδοι μετα-μάθησης (επίσης γνωστές ως συλλογική μάθηση).
- neural_networks: Μέθοδοι βασισμένες σε νευρωνικά δίκτυα.
- prototype: Μέθοδοι μάθησης βασισμένες σε πρωτότυπα.
- rules: Μέθοδοι μάθησης βασισμένες σε κανόνες.
- transform: Εκτελούν μετασχηματισμούς δεδομένων.
- trees: Μέθοδοι μάθησης βασισμένες σε δέντρα.

Τα τελευταία έτη έχει διαπιστωθεί ο πολλαπλασιασμός του Ελεύθερου Λογισμικού και του Λογισμικού Ανοικτού Κώδικα (FOSS) στην ερευνητική κοινότητα, στον τομέα της μηχανικής μάθησης δηλαδή οι ερευνητές επωφελήθηκαν από τη διαθεσιμότητα διαφορετικών πλαισίων που παρέχουν εργαλεία για ταχύτερη ανάπτυξη, επιτρέπουν τις διαδικασίες της αναπαραγωγικότητας και της αντιγραφής των αποτελεσμάτων και

ενθαρρύνουν τη συνεργασία. Ακολουθώντας τις αρχές του FOSS, οι Montiel et al, (2018), εισάγουν το scikit-multiow, ένα πλαίσιο της Python για την υλοποίηση αλγορίθμων και την πραγματοποίηση πειραμάτων στον τομέα της μηχανικής μάθησης στις εξελισσόμενες ροές δεδομένων.

Το scikit-multiow εμπνέεται από τα δημοφιλή πλαίσια scikit-learn, MOA και MEKA. Η Scikit-learn (Pedregosa et al., 2011) είναι η πιο δημοφιλής βιβλιοθήκη λογισμικού ανοικτού κώδικα μηχανικής μάθησης για τη γλώσσα προγραμματισμού Python. Διαθέτει διάφορους αλγόριθμους ταξινόμησης, παλινδρόμησης και ομαδοποίησης, συμπεριλαμβανομένων των support vector machines, random forest, gradient boosting, K-means και DBSCAN και έχει σχεδιαστεί για να αλληλεπιδρά με τα αριθμητικά και επιστημονικά πακέτα NumPy και SciPy της Python.

Το MOA (Bifet et al., 2010) αποτελεί το πλέον δημοφιλές πλαίσιο ανοικτού κώδικα για τη διαδικασία εξόρυξης ροής δεδομένων, με μια πολύ ενεργή αναπτυσσόμενη κοινότητα, που περιλαμβάνει μια συλλογή αλγορίθμων μηχανικής μάθησης (ταξινόμηση, παλινδρόμηση, ομαδοποίηση, ανίχνευση των εξωστρεφών, συστήματα ανίχνευσης ιδεών και συστήματα σύστασης) και εργαλεία αξιολόγησης. Σχετικά με το σχέδιο WEKA (Hall et al., 2009), το MOA γράφτηκε επίσης σε Java, ενώ εξελίσσεται σε πιο απαιτητικά προβλήματα. Το πρόγραμμα MEKA (Read et al., 2016) παρέχει μια ανοιχτού κώδικα εφαρμογή μεθόδων προς την κατεύθυνση της εκμάθησης και αξιολόγησης πολλαπλών ετικετών, διαδικασία κατά την οποία στόχος είναι η πρόβλεψη πολλαπλών μεταβλητών εξόδου για κάθε περίπτωση εισαγωγής, πράγμα που διαφέρει από την περίπτωση standard '(δυαδική ή πολυταξική ταξινόμηση) η οποία περιλαμβάνει μόνο μία μεταβλητή στόχο.

Ως πλατφόρμα πολλαπλών εξόδων, το scikit-multiow χρησιμεύει ως γέφυρα ανάμεσα σε ερευνητικές κοινότητες που έχουν αναπτυχθεί γύρω από τα προαναφερθέντα δημοφιλή πλαίσια, παρέχοντας ένα κοινό έδαφος όπου μπορούν να ευδοκιμήσουν. Το scikit-multiow βοηθά στον εκδημοκρατισμό της Μάθησης Stream φέρνοντας αυτόν τον τομέα έρευνας πιο κοντά στην κοινότητα της μηχανικής μάθησης, δεδομένης της αυξανόμενης δημοτικότητας της γλώσσας προγραμματισμού της Python, με τον στόχο να είναι διπλός, πρώτον, δηλαδή να γεμίζει το κενό στην Python για ένα πλαίσιο μάθησης ροών το οποίο μπορεί επίσης να αλληλεπιδρά με διαθέσιμα εργαλεία όπως το scikit-learn και να επεκτείνει το σύνολο των διαθέσιμων μεθόδων τελευταίας τεχνολογίας σε αυτήν την πλατφόρμα και δεύτερον, να παρέχει ένα σύνολο εργαλείων στην κατεύθυνση της

διευκόλυνσης της ανάπτυξης της έρευνας για την εκμάθηση ροών, με χαρακτηριστικό το παράδειγμα των Montiel et al., (2018).

Είναι σημαντικό να παρατηρηθεί ότι το scikit-multiow συμπληρώνει το scikit-learn, του οποίου πρωταρχικός στόχος είναι η μαζική μάθηση, επεκτείνοντας το σύνολο των εργαλείων ελεύθερης και ανοιχτής πηγής για μάθηση ροών. Επιπλέον, το scikit-multiow μπορεί να χρησιμοποιηθεί μέσα στα Jupyter Notebooks, ένα δημοφιλές περιβάλλον επικοινωνίας στην κοινότητα της επιστήμης δεδομένων. Ειδική εστίαση στο σχεδιασμό του Scikit-multiow είναι να γίνει φιλικό προς τους νέους χρήστες και οικείο στους έμπειρους.

Το scikit-multiow περιέχει γεννήτριες ροών, μεθόδους μάθησης, ανιχνευτές αλλαγής και μεθόδους αξιολόγησης. Οι γεννήτριες ροών περιλαμβάνουν: Agrawal, Hyperplane, Led, Mixed, Random-RBF, Random-RBF with drift, Random Tree, SEA, SINE, SEA, STAGGER, Waveform, Multi-label, Regression and Concept-Drift. Οι διαθέσιμοι αξιολογητές αντιστοιχούν στις αξιολογήσεις Prequential και Hold-Out και υποστηρίζουν ταυτοχρόνως πολλές μετρήσεις απόδοσης για την ταξινόμηση (ACCURACY, Kappa Coefficient, Kappa T, Kappa M), Ταξινόμηση πολλαπλών αποτελεσμάτων (Hamming Score, Hamming Loss, Exact Match, Jaccard Index) Παλινδρόμηση (Μέσο τετραγωνικό σφάλμα, Μέσο απόλυτο σφάλμα) και παλινδρόμηση πολλαπλών εξόδων (μέσο τετραγωνικό σφάλμα, μέσο απόλυτο σφάλμα, μέσο ριζικό τετραγωνικό σφάλμα).

## 4.1 Σήμανση και Υπόβαθρο Εξόρυξης Ροής Δεδομένων

Εξετάζεται μια συνεχής ροή δεδομένων A = $\{\vec{x}_t, y_t\}|t = 1, \ldots, T$ όπου $T \to \infty$. Η εισαγωγή $\vec{x}_t$ είναι ένα διάνυσμα χαρακτηριστικών και yt ο αντίστοιχος στόχος όπου το y είναι συνεχές σε περίπτωση παλινδρόμησης και διακριτό για ταξινόμηση, ενώ σκοπός είναι να προβλεφθεί ο στόχος y για ένα άγνωστο $\vec{x}$. Στα παραδοσιακά μοντέλα μονής εξόδου, ασχολούνται με μια μόνο μεταβλητή στόχου για την οποία παράγεται μία αντίστοιχη έξοδος ανά δοκιμαστική περίπτωση, ενώ τα μοντέλα πολλαπλών εξόδων μπορούν να παράγουν πολλαπλές εξόδους για να εκχωρηθούν σε πολλαπλές μεταβλητές στόχου $\vec{y}$ για κάθε περίπτωση δοκιμής.

Διαφορετικά από τη μάθηση κατά παρτίδες, όπου όλα τα δεδομένα (X, y) είναι διαθέσιμα για εκπαίδευση, στην εκμάθηση ροών, η εκπαίδευση γίνεται σταδιακά καθώς

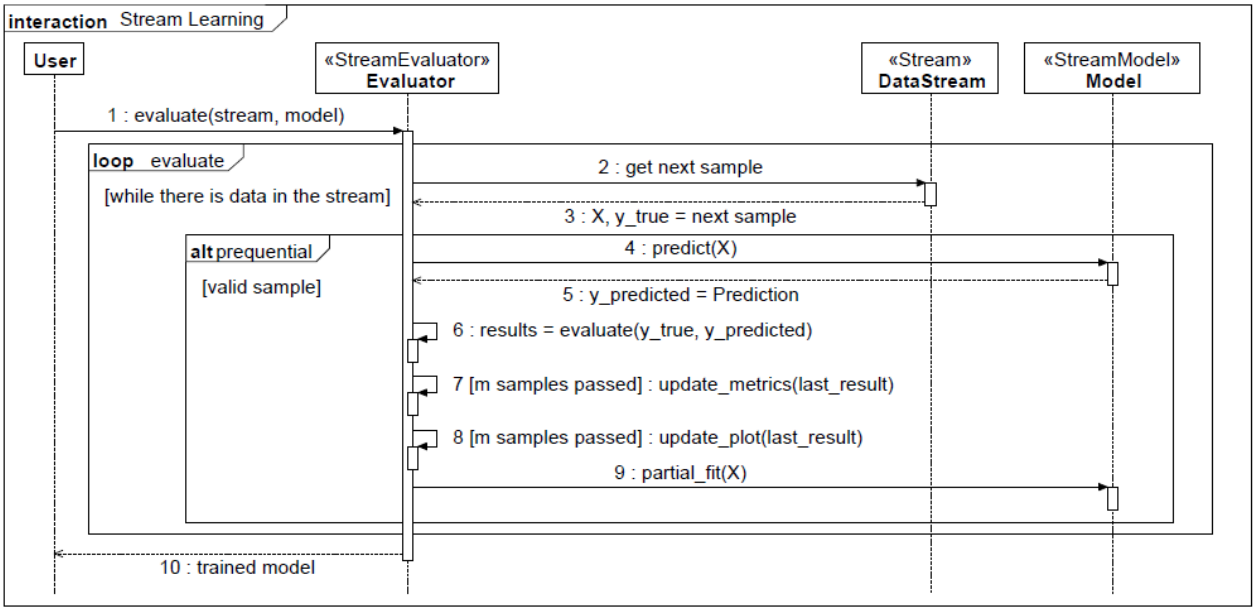
είναι διαθέσιμα νέα δεδομένα ($\vec{x}_i, y_i$). Η απόδοση P ενός δεδομένου μοντέλου μετράται σύμφωνα με κάποια συνάρτηση απώλειας που αξιολογεί τη διαφορά μεταξύ του συνόλου των αναμενόμενων ετικετών Y και των προβλεπόμενων $\hat{Y}$. Η αξιολόγηση Hold-out είναι μια δημοφιλής μέθοδος αξιολόγησης της απόδοσης για batch settings και ροής, όπου οι δοκιμές πραγματοποιούνται σε ξεχωριστό σετ δοκιμών. Η Prequential αξιολόγηση (Dawid, 1984) ή η αξιολόγηση interleaved-test then-train είναι μια δημοφιλής μέθοδος αξιολόγησης της απόδοσης για τη ρύθμιση ροής μόνο, όπου οι δοκιμές εκτελούνται σε νέα δεδομένα πριν τη χρησιμοποιήσουν για την εκπαίδευση του μοντέλου.

## 4.2 Αρχιτεκτονική του Scikit-Multiow

Η βασική κλάση στο scikit-multiow είναι το StreamModel που περιέχει τις ακόλουθες αφηρημένες μεθόδους που πρέπει να εφαρμοστούν από τις υποκατηγορίες του: _ fit | Εκπαιδεύει ένα μοντέλο κατά batch τρόπο και λειτουργεί ως διεπαφή σε μεθόδους batch που εφαρμόζουν μια συνάρτηση fit () όπως οι μέθοδοι scikit-learn. _ partial fit | Αυξανόμενα εκπαιδεύει ένα μοντέλο ροής.

_predict |Προβλέπει την αξία του στόχου σε μεθόδους μάθησης υπό επίβλεψη. _predict_proba| Υπολογίζει πιθανότητες ανά τάξη σε προβλήματα ταξινόμησης.

Σε ένα αντικείμενο StreamModel παρατηρείται αλληλεπίδραση με δύο άλλα αντικείμενα: ένα αντικείμενο Stream και (προαιρετικά) ένα αντικείμενο StreamEvaluator, με το αντικείμενο Stream να παρέχει συνεχή ροή δεδομένων κατόπιν αιτήματος και το αντικείμενο StreamEvaluator να εκτελεί πολλαπλές εργασίες: ερωτά τη ροή δεδομένων, εκπαιδεύει και δοκιμάζει το μοντέλο στα εισερχόμενα δεδομένα και παρακολουθεί συνεχώς την απόδοση του μοντέλου. Η ακολουθία για την training ενός μοντέλου Stream και την παρακολούθηση της απόδοσής του με βάση την προηγούμενη αξιολόγηση στο scikit-multiow περιγράφεται στο σχήμα 2.




**Σχήμα 2.** Εκπαίδευση και δοκιμή ενός μοντέλου ροών χρησιμοποιώντας το scikit-multiow. Αυτή η ακολουθία αντιστοιχεί στην prequential αξιολόγηση.

Η κλάση BaseSKMObject είναι η βασική κλάση. Όλοι οι εκτιμητές στο scikit-multiflow δημιουργούνται με την επέκταση της βασικής κλάσης και των αντίστοιχων εξειδικευμένων ως προς την εργασία μίξεων: ClassifierMixin, RegressorMixin, MetaEstimatorMixin και MultiOutputMixin.

Η ClassifierMixin ορίζει τις ακόλουθες μεθόδους:

- partial_fit – Εκπαιδεύει αυξητικά τον εκτιμητή με τα παρεχόμενα δεδομένα labels.
- fit – Διεπαφή που χρησιμοποιείται για την διαβίβαση δεδομένων εκπαίδευσης ως δεσμίδες.
- predict – Προβλέπει την τιμή κλάσης για τα δεδομένα που έχουν περάσει χωρίς label.
- predict_proba – Υπολογίζει την πιθανότητα για ένα δείγμα να υπάγεται σε μια δοσμένη κλάση.

Κατά τη διάρκεια μιας διεργασίας μάθησης, εκτελούνται τρεις κύριες εργασίες: τα δεδομένα παρέχονται από τη ροή, ο εκτιμητής εκπαιδεύεται στα εισερχόμενα δεδομένα, αξιολογείται η απόδοση του εκτιμητή. Στο scikit-multiflow, τα δεδομένα αντιπροσωπεύονται από την κλάση Stream, όπου χρησιμοποιείται η μέθοδος

next_sample() για να ζητήσει νέα δεδομένα. Η κλάση StreamEvaluator παρέχει έναν εύκολο τρόπο ρύθμισης πειραμάτων. Υπάρχουν διαθέσιμες υλοποιήσεις για μεθόδους αξιολόγησης holdout και prequential. Μια ροή και ένας ή περισσότεροι εκτιμητές μπορούν να διαβιβαστούν σε έναν αξιολογητή.

## 4.3 Εργασία ταξινόμησης

Σε αυτό το παράδειγμα, θα χρησιμοποιηθεί η γεννήτρια SEA. Μια γεννήτρια ροής δεν αποθηκεύει δεδομένα αλλά τα παράγει κατά ζήτηση. Η κλάση SEAGenerator δημιουργεί δεδομένα που αντιστοιχούν σε ένα πρόβλημα δυαδικής ταξινόμησης. Τα δεδομένα περιέχουν 3 αριθμητικά χαρακτηριστικά, από τα οποία μόνο 2 σχετίζονται με την μάθηση. Θα χρησιμοποιηθούν τα δεδομένα από τη γεννήτρια για την εκπαίδευση ενός ταξινομητή Naive Bayes. Για συνέπεια, τα ακόλουθα παραδείγματα δεν περιλαμβάνουν δηλώσεις εισαγωγής (import statements) και οι εξωτερικές βιβλιοθήκες αναφέρονται από πρότυπα ψευδώνυμα (aliases).

Όπως αναφέρθηκε προηγουμένως, μια δημοφιλής μέθοδος για την παρακολούθηση της απόδοσης των μεθόδων μάθησης ροής είναι η αξιολόγηση prequential. Όταν ένα νέο δείγμα δεδομένων (X, y) καταφτάσει: 1. Προβλέψεις λαμβάνονται για το νέο δείγμα (X) ώστε να αξιολογηθεί το κατά πόσο καλά αποδίδει το μοντέλο. 2. Κατόπιν το νέο δείγμα δεδομένων (X, y) χρησιμοποιείται για να εκπαιδεύσει το μοντέλο οπότε και ενημερώνει την εσωτερική του κατάσταση. Η αξιολόγηση prequential μπορεί εύκολα να υλοποιηθεί ως βρόχος:

```
stream = SEAGenerator(random_state=1)
classifier = NaiveBayes()

n_samples = 0
correct_cnt = 0
max_samples = 2000

# Βρόχος αξιολόγησης μεθόδου prequential
while n_samples < max_samples and \
```

```
stream.has_more_samples():
  X, y = stream.next_sample()
  # Πρόβλεψη κλάσης για τα νέα δεδομένα
  y_pred = classifier.predict(X)
  if y[0] == y_pred[0]:
    correct_cnt += 1
  # Partially fit (εκπαίδευση) μοντέλου με νέα δεδομένα
  classifier.partial_fit(X, y)
  n_samples += 1

print('{} samples analyzed.'.format(n_samples))
print('Accuracy: {}'.format(correct_cnt / n_samples))

> 2000 samples analyzed.
> NaiveBayes classifier accuracy: 0.9395
```

Το προηγούμενο παράδειγμα δείχνει ότι ο ταξινομητής Naive Bayes επιτυγχάνει ακρίβεια 93.95% μετά την επεξεργασία όλων των δειγμάτων.

Ωστόσο, η μάθηση από ροές δεδομένων είναι μια συνεχής εργασία και μια βέλτιστη πρακτική είναι η παρακολούθηση της απόδοσης του μοντέλου σε διαφορετικά σημεία της ροής. Σε αυτό το παράδειγμα, χρησιμοποιείται ένα instance της κλάσης Stream καθώς παρέχει τη μέθοδο next_sample() για την αίτηση δεδομένων και τα δεδομένα που επιστρέφονται είναι μια πλειάδα του numpy.ndarray.

Έτσι, ο παραπάνω βρόχος μπορεί εύκολα να τροποποιηθεί για ανάγνωση από άλλες δομές δεδομένων όπως οι numpy.ndarray ή pandas.DataFrame.

Για εφαρμογές σε πραγματικό χρόνο όπου τα δεδομένα αντιπροσωπεύονται εξ' ολοκλήρου ως ροή (π.χ. buffers πρωτοκόλλου της Google), η κλάση Stream μπορεί να

επεκταθεί για να «τυλίξει» (wrap) τον απαραίτητο κώδικα για να αλληλεπιδράσει με τη ροή.

Η μέθοδος αξιολόγησης prequential υλοποιείται στην EvaluatePrequential κλάση. Αυτή η κλάση παρέχει επιπλέον λειτουργικότητες συμπεριλαμβανομένων των:

- Εύκολη εγκατάσταση διαφορετικών διαμορφώσεων αξιολόγησης
- Επιλογή διαφορετικών μετρικών απόδοσης
- Οπτικοποίηση της απόδοσης έναντι του χρόνου
- Δυνατότητα συγκριτικής αξιολόγησης πολλαπλών μοντέλων ταυτόχρονα
- Αποθήκευση αποτελεσμάτων αξιολόγησης σε αρχείο csv

Το ίδιο πείραμα μπορεί να εκτελεστεί στα δεδομένα SEA. Αυτή τη φορά θα συγκριθούν οι δύο ταξινομητές: NaiveBayes και SGDClassifier (γραμμική SVM με SGD εκπαίδευση). Θα χρησιμοποιηθεί ο SGDClassifier προκειμένου να επιδειχθεί η συμβατότητα με τις αυξητικές μεθόδους από το scikit-learn.

```
stream = SEAGenerator(random_state=1)
nb = NaiveBayes()
svm = SGDClassifier()
# Ρύθμιση της αξιολόγησης
metrics = ['accuracy', 'kappa',
           'running_time', 'model_size']
eval = EvaluatePrequential(show_plot=True,
                           max_samples=20000,
                           metrics=metrics)
# Εκτέλεση της αξιολόγησης
eval.evaluate(stream=stream, model=[nb, svm],
                      model_names=['NB', 'SVM']);
```

Ορίζονται δύο μετρικές για την μέτρηση της προβλεπτικής απόδοσης: Ακρίβεια και στατιστικά Κάππα (Cohen, 1960) (για συγκριτική αξιολόγηση της ακρίβειας ταξινόμησης κάτω από ανισορροπία κλάσης, συγκρίνει την ακρίβεια των μοντέλων με εκείνη ενός τυχαίου ταξινομητή). Κατά τη διάρκεια της αξιολόγησης, μια δυναμική γραφική αναπαράσταση απεικονίζει την απόδοση και των δύο εκτιμητών έναντι της ροής. Μόλις ολοκληρωθεί η αξιολόγηση, εμφανίζεται μια αναφορά στο τερματικό. Για αυτό το παράδειγμα και λαμβάνοντας υπόψη τη διαμόρφωση αξιολόγησης:

Processed samples: 20000

Mean performance:

NB - Accuracy : 0.9430

NB - Kappa : 0.8621

NB - Training time (s) : 0.56

NB - Testing time (s) : 1.31

NB - Total time (s) : 1.87

NB - Size (kB) : 6.8076

SVM - Accuracy : 0.9560

SVM - Kappa : 0.8984

SVM - Training time (s) : 4.70

SVM - Testing time (s) : 1.73

SVM - Total time (s) : 6.43

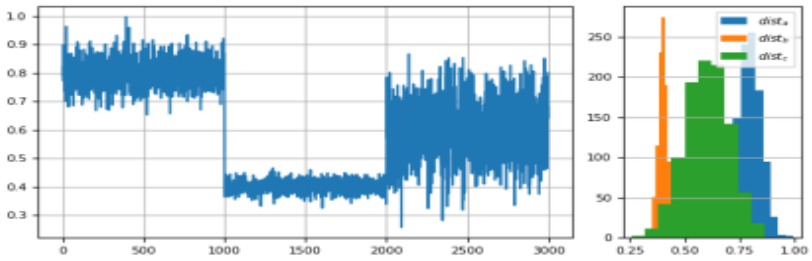
SVM - Size (kB) : 3.4531

Παρόλο που ο NaiveBayes έχει καλύτερη απόδοση στην αρχή της ροής, ο SGDClassifier τελικά τον ξεπερνά. Βέβαια δεν μπορούμε να πούμε με βεβαιότητα ότι ο SGDClassifier πέτυχε τελικά καλύτερη απόδοση, μιας και η διαφορά τους είναι πολύ μικρή. Στη γραφική αναπαράσταση παρουσιάζεται η απόδοση σε πολλά σημεία, μετρούμενη από τη δεδομένη μετρική (ακρίβεια, Κάππα κ.λπ.) με δύο τρόπους: Ο *μέσος όρος (mean)*

αντιστοιχεί στην μέση απόδοση όλων των δεδομένων που εμφανίστηκαν προηγουμένως, με αποτέλεσμα μια ομαλή γραμμή. Το *τρέχον (current)* δείχνει την απόδοση σε ένα συρόμενο παράθυρο με τα πιο πρόσφατα δεδομένα από τη ροή. Το μέγεθος του συρόμενου παραθύρου μπορεί να οριστεί από τον χρήστη και είναι χρήσιμο να αναλυθεί η «τρέχουσα» απόδοση ενός εκτιμητή. Σε αυτό το πείραμα μετρούνται επίσης πόροι σε όρους χρόνου (εκπαίδευση + δοκιμή) και μνήμης. Ο NaiveBayes είναι γρηγορότερος και δεσμεύει ελαφρώς περισσότερη μνήμη. Από την άλλη, ο SGDClassifier είναι αργότερος και έχει μικρότερο αποτύπωμα μνήμης.

## 4.4 Ανίχνευση Εννοιολογικής Απόκλισης

Για αυτό το παράδειγμα, θα παραχθεί μια συνθετική ροή δεδομένων. Τα πρώτα 1000 δείγματα της ροής περιέχουν μια ακολουθία από μια κανονική κατανομή με $\mu_\alpha = 0.8$, $\sigma_\alpha = 0.05$, ακολουθούμενα από 1000 δείγματα από μια κανονική κατανομή με $\mu_b = 0.4$, $\sigma_b = 0.2$ και τα τελευταία 1000 δείγματα από μια κανονική κατανομή με $\mu_c = 0.\ 6$, $\sigma_c = 0.1$. Η κατανομή των δεδομένων στην περιγραφόμενη συνθετική ροή φαίνεται στο σχήμα 3.




**Σχήμα 3.** Συνθετικά δεδομένα που προσομοιώνουν την απόκλιση. Η ροή αποτελείται από δύο κατανομές των 500 δειγμάτων.

```
random_state = np.random.RandomState(12345)

dist_a = random_state.normal(0.8, 0.05, 1000)

dist_b = random_state.normal(0.4, 0.02, 1000)

dist_c = random_state.normal(0.6, 0.1, 1000)

stream = np.concatenate((dist_a, dist_b, dist_c))
```

Θα γίνει χρήση της Προσαρμοστικής Παραθυροποίησης (ADWIN) ως μεθόδου ανίχνευσης εννοιολογικής απόκλισης. Ο σκοπός είναι να ανιχνευθεί η απόκλιση που συμβαίνει μετά τα δείγματα 1000 και 2000 στη συνθετική ροή δεδομένων.

```
drift_detector = ADWIN()


for i, val in enumerate(stream_int):
    drift_detector.add_element(val)
    if drift_detector.detected_change():
        print('Change detected at index {}'.format(i))


        drift_detector.reset()
```

```
> Change detected at index 1055
> Change detected at index 2079
```

## 4.5 Επίδραση της απόκλισης στην μάθηση

Η concept drift μπορεί να έχει σημαντικό αντίκτυπο στην προβλεπτική απόδοση εάν δεν αντιμετωπιστεί σωστά. Τα περισσότερα μοντέλα δεσμίδων θα αποτύχουν υπό την παρουσία απόκλισης, καθώς ουσιαστικά εκπαιδεύονται σε διαφορετικά δεδομένα. Από την άλλη πλευρά, οι μέθοδοι μάθησης ροής ενημερώνονται συνεχώς και μπορούν να προσαρμοστούν σε νέες έννοιες. Επιπλέον, οι μέθοδοι με επίγνωση της απόκλισης χρησιμοποιούν μεθόδους ανίχνευσης αλλαγών για να πυροδοτήσουν μηχανισμούς μετριασμού εάν ανιχνευτεί αλλαγή στην απόδοση.

Στο παράδειγμα αυτό, συγκρίνονται δύο δημοφιλή μοντέλα ροής: Το HoeffdingTreeClassifier και η εκδοχή του HoeffdingAdaptiveTreeClassifier που λαμβάνει υπόψη την απόκλιση.

Για αυτό το παράδειγμα, θα φορτωθούν δεδομένα από αρχείο csv χρησιμοποιώντας την κλάση FileStream. Τα δεδομένα αντιστοιχούν στην έξοδο του AGRAWALGenerator με 3 βαθμιαίες αποκλίσεις στα 5k, 10k, και 15k δείγματα. Μια βαθμιαία απόκλιση σημαίνει ότι η παλιά έννοια σταδιακά αντικαθίσταται από μια καινούρια, με άλλα λόγια, υπάρχει μια μεταβατική περίοδος στην οποία και οι δύο έννοιες είναι παρούσες.

```
stream = FileStream("agr_a_20k.csv")
ht = HoeffdingTreeClassifier(),
hat = HoeffdingAdaptiveTreeClassifier()
# Ρύθμιση της αξιολόγησης
metrics = ['accuracy', 'kappa', 'model_size']
eval = EvaluatePrequential(show_plot=True,
metrics=metrics,
n_wait=100)
# Εκτέλεση της αξιολόγησης
eval.evaluate(stream=stream, model=[hy, hat],
model_names=['HT', 'HAT']);
```

Τα αποτελέσματα της αξιολόγησης είναι:

```
Processed samples: 20000
Mean performance:
HT - Accuracy    : 0.7279
HT - Kappa       : 0.4530
HT - Size (kB)   : 175.8711
HAT - Accuracy   : 0.8070
HAT - Kappa      : 0.6122
HAT - Size (kB)  : 122.0986
```

Κατά τη διάρκεια των πρώτων 5k δειγμάτων, παρατηρείται ότι και οι δύο μέθοδοι συμπεριφέρονται με παρόμοιο τρόπο, κάτι που αναμένεται καθώς το HoeffdingAdaptiveTreeClassifier λειτουργεί ουσιαστικά σαν HoeffdingTreeClassifier όταν δεν υπάρχει απόκλιση. Στο σημείο 5k, η πρώτη απόκλιση εμφανίζεται από την ξαφνική πτώση της απόδοσης και των δύο εκτιμητών. Ωστόσο, να παρατηρηθεί ότι το HoeffdingAdaptiveTreeClassifier έχει το πλεονέκτημα και ανακάμπτει γρηγορότερα. Η

ίδια συμπεριφορά παρατηρείται μετά την απόκλιση στο σημείο των 15k. Είναι ενδιαφέρον, μετά την απόκλιση στα 10k, το HoeffdingTreeClassifier είναι καλύτερο για μια μικρή περίοδο αλλά μετά γρήγορα ξεπερνιέται. Σε αυτό το πείραμα, μπορεί επίσης να ιδωθεί ότι η «τρέχουσα» αξιολόγηση απόδοσης παρέχει πιο πλούσιες πληροφορίες για την απόδοση κάθε εκτιμητή. Αξίζει να σημειωθεί η διαφορά στη μνήμη μεταξύ αυτών των εκτιμητών. Το HoeffdingAdaptiveTreeClassifier επιτυγχάνει καλύτερη απόδοση ενώ απαιτεί λιγότερο χώρο στη μνήμη. Αυτό δείχνει ότι έχει εφαρμοστεί ο μηχανισμός αντικατάστασης κλάδου που ενεργοποιείται από το ADWIN, με αποτέλεσμα μια λιγότερο περίπλοκη δομή δέντρου να αντιπροσωπεύει τα δεδομένα.

# 5 Επίλογος

## 5.1 Σύνοψη και συμπεράσματα

Τα «γρήγορα δεδομένα» προέκυψαν για να συλλάβουν την ιδέα ότι οι ροές δεδομένων δημιουργούνται με πολύ υψηλούς ρυθμούς και ότι αυτά πρέπει να αναλυθούν γρήγορα για να φθάσουν σε ευαίσθητες πληροφορίες, ενώ μπορούν να έρχονται από μετρήσεις δικτύου, εγγραφές κλήσεων, επισκέψεις σε ιστοσελίδες, αναγνώσεις αισθητήρων κ.λ.π. Το γεγονός ότι τα εν λόγω δεδομένα φθάνουν συνεχώς σε πολλαπλές, ταχείες, χρονικά μεταβαλλόμενες, πιθανώς απρόβλεπτες και απεριόριστες ροές φαίνεται να αποδίδει ορισμένα θεμελιωδώς νέα ερευνητικά προβλήματα, με παραδείγματα τέτοιων εφαρμογών να είναι οι οικονομικές εφαρμογές, η παρακολούθηση του δικτύου, η ασφάλεια, η διαχείριση δικτύων αισθητήρων, η ανάλυση δεδομένων που προέρχονται από μέσα κοινωνικής δικτύωσης όπως το Twitter κ.λ.π.

Η διαδικασία της παραδοσιακής επεξεργασίας δεδομένων προϋποθέτει ότι τα δεδομένα είναι διαθέσιμα για πολλαπλή πρόσβαση, ακόμη και αν σε ορισμένες περιπτώσεις βρίσκονται αποθηκευμένα σε κάποιο μέσο και μπορούν να επεξεργαστούν μόνο σε μεγαλύτερα μέρη, περίπτωση κατά την οποία λέγεται ότι τα δεδομένα είναι σε κατάσταση ηρεμίας και μπορεί να εκτελεστεί επεξεργασία κατά τμήματα. Τα συστήματα βάσεων δεδομένων, αποτελούν ένα χαρακτηριστικό παράδειγμα αυτού, αποθηκεύοντας μεγάλες συλλογές δεδομένων και επιτρέποντας στους χρήστες να υποβάλλουν ερωτήματα και να εξάγουν πληροφορίες, ωστόσο, τα γρήγορα δεδομένα ή τα δεδομένα σε ροή έχουν στενή σύνδεση και σε ορισμένες περιπτώσεις γίνεται χρήση αυτών ως συνώνυμα του υπολογιστικού μοντέλου των ροών δεδομένων. Σε αυτό το μοντέλο, τα δεδομένα φθάνουν συνεχώς σε μια δυνητικά άπειρη ροή η οποίο πρέπει να υποβληθεί σε επεξεργασία από ένα σύστημα με περιορισμένους πόρους. Ο κύριος περιορισμός είναι ότι η κύρια μνήμη είναι μικρή και μπορεί να περιέχει μόνο ένα μικρό τμήμα του ρεύματος, επομένως τα περισσότερα δεδομένα πρέπει να απορρίπτονται αμέσως μετά την επεξεργασία. Η διαδικτυακή μάθηση ενημερώνει το μοντέλο της μετά από κάθε εμφάνιση δεδομένων χωρίς πρόσβαση σε όλα τα δεδομένα του παρελθόντος, εξ ου και ισχύουν οι περιορισμοί του υπολογιστικού μοντέλου ροής δεδομένων. Η ροή δεδομένων δεν είναι απλώς ένας τεχνικός περιορισμός στη μηχανική μάθηση: Τα γρήγορα δεδομένα δεν αφορούν μόνο την ισχύ επεξεργασίας αλλά και τη γρήγορη σημασιολογία.

Στην παρούσα εργασία πραγματοποιήθηκε η μελέτη και η σύγκριση υλοποιημένων αλγορίθμων κατηγοριοποίησης του scikit-multiflow πάνω σε ροές δεδομένων. Το scikit-multiflow, αποτελεί ένα περιβάλλον μηχανικής μάθησης και εξόρυξης γνώσης ανοικτού κώδικα για δεδομένα πολλαπλών εξόδων / πολλαπλών ετικετών και ροών δεδομένων. Το scikit-multiflow είναι υλοποιημένο σε Python και παρουσιάζει αυξανόμενη δημοτικότητα. Συμπληρώνει το scikit-learn, του οποίου η κύρια εστίαση είναι η μάθηση κατά παρτίδες, επεκτείνοντας το σύνολο των εργαλείων εκμάθησης μηχανών σε αυτήν την πλατφόρμα. Στην τρέχουσα κατάσταση του, το scikit-multiflow περιέχει γεννήτριες ροών δεδομένων, κατηγοριοποιητές πολλαπλών εξόδων / πολλαπλών ετικετών δεδομένων ροής, ανιχνευτές αλλαγής και μεθόδους αξιολόγησης.

Στην εργασία παρουσιάστηκε ένα παράδειγμα, εργασίας ταξινόμησης στο οποίο χρησιμοποιήθηκε η γεννήτρια SEA, μια γεννήτρια ροής δεν αποθηκεύει δεδομένα αλλά τα παράγει κατά ζήτηση. Χρησιμοποιήθηκαν τα δεδομένα από τη γεννήτρια για την εκπαίδευση ενός ταξινομητή Naive Bayes, ο οποίος πέτυχε ακρίβεια 93.95% μετά την επεξεργασία όλων των δειγμάτων. Στη συνέχεια συγκρίθηκαν οι ταξινομητές NaiveBayes και SGDClassifier. Στη γραφική αναπαράσταση παρουσιάζεται η απόδοση σε πολλά σημεία, μετρούμενη από τη δεδομένη μετρική (ακρίβεια, Κάππα κ.λπ.) με δύο τρόπους: Ο μέσος όρος (mean) αντιστοιχεί στην μέση απόδοση όλων των δεδομένων που εμφανίστηκαν προηγουμένως, με αποτέλεσμα μια ομαλή γραμμή. Το τρέχον (current) δείχνει την απόδοση σε ένα συρόμενο παράθυρο με τα πιο πρόσφατα δεδομένα από τη ροή. Το μέγεθος του συρόμενου παραθύρου μπορεί να οριστεί από τον χρήστη και είναι χρήσιμο να αναλυθεί η «τρέχουσα» απόδοση ενός εκτιμητή. Σε αυτό το πείραμα μετρούνται επίσης πόροι σε όρους χρόνου (εκπαίδευση + δοκιμή) και μνήμης. Ο NaiveBayes είναι γρηγορότερος και δεσμεύει ελαφρώς περισσότερη μνήμη. Από την άλλη, ο SGDClassifier είναι αργότερος και έχει μικρότερο αποτύπωμα μνήμης.

Για την μελέτη της επίδρασης της απόκλισης στη μάθηση συγκρίθηκαν δύο δημοφιλή μοντέλα ροής, το HoeffdingTreeClassifier και η εκδοχή του HoeffdingAdaptiveTreeClassifier που λαμβάνει υπόψη την απόκλιση. Κατά τη διάρκεια των πρώτων 5k δειγμάτων, παρατηρήθηκε ότι και οι δύο μέθοδοι συμπεριφέρονται με παρόμοιο τρόπο. Ωστόσο, στο σημείο 5k, η πρώτη απόκλιση εμφανίζεται από την ξαφνική πτώση της απόδοσης και των δύο εκτιμητών, με το HoeffdingAdaptiveTreeClassifier να έχει το πλεονέκτημα και να ανακάμπτει γρηγορότερα. Η ίδια συμπεριφορά παρατηρείται μετά την απόκλιση στο σημείο των 15k.

Είναι ενδιαφέρον, μετά την απόκλιση στα 10k, το HoeffdingTreeClassifier παρουσίασε καλύτερο για μια μικρή περίοδο αλλά μετά γρήγορα ξεπερνιέται. Το HoeffdingAdaptiveTreeClassifier πέτυχε καλύτερη απόδοση ενώ απαιτούσε λιγότερο χώρο στη μνήμη. Αυτό δείχνει ότι έχει εφαρμοστεί ο μηχανισμός αντικατάστασης κλάδου που ενεργοποιείται από το ADWIN, με αποτέλεσμα μια λιγότερο περίπλοκη δομή δέντρου να αντιπροσωπεύει τα δεδομένα.

## 5.2 Περιορισμοί της έρευνας και Μελλοντικές Επεκτάσεις

Η συγκεκριμένη εργασία περιορίστηκε σε δεδομένα που αναζητήθηκαν από μελέτες της διεθνούς βιβλιογραφίας που σχετίζονται με το θέμα, χωρίς την παραγωγή πρωτογενών δεδομένων. Περιορίστηκε στην παρουσίαση, την ανάλυση και την κριτική αξιολόγηση παραδειγμάτων και αλγοριθμικών τεχνικών που χρησιμοποιήθηκαν σε άλλες σχετικές μελέτες. Ως επέκταση της εργασίας, στο μέλλον θα μπορούσε να πραγματοποιηθεί διερεύνηση και η σύγκριση πρωτογενών αλγορίθμων κατηγοριοποίησης του scikit-multiflow πάνω σε ροές δεδομένων.

# Βιβλιογραφία


Abadi, D. J., Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Lee, S., ... & Zdonik, S. (2003). Aurora: a new model and architecture for data stream management. *the VLDB Journal*, *12*(2), 120-139.

Aberdeen, D., Pacovsky, O., & Slater, A. (2010). The learning behind gmail priority inbox.

Aggarwal, C. C., Bhuiyan, M. A., & Al Hasan, M. (2014). Frequent pattern mining algorithms: A survey. In *Frequent pattern mining* (pp. 19-64). Springer, Cham.

Agrawal, R., Imieliński, T., & Swami, A. (1993, June). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data* (pp. 207-216).

Alberg, D., Last, M., & Kandel, A. (2012). Knowledge discovery in data streams with regression tree methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2*(1), 69-78.

Andras A. Benczur and Levente Kocsis and Robert Palovics. ΄΄Online Machine Learning in Big Data Streams΄΄

Babu, S., & Widom, J. (2001). Continuous queries over data streams. *ACM Sigmod Record*, *30*(3), 109-120.

Bifet, A., & Frank, E. (2010, October). Sentiment knowledge discovery in twitter streaming data. In *International conference on discovery science* (pp. 1-15). Springer, Berlin, Heidelberg.

Bifet, A., & Gavalda, R. (2007, April). Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining* (pp. 443-448). Society for Industrial and Applied Mathematics.

Bifet, A., Holmes, G., Kirkby, R., & Pfahringer, B. (2010). Moa: Massive online analysis. *Journal of Machine Learning Research*, *11*(May), 1601-1604.



Bifet, A., Holmes, G., Kirkby, R., & Pfahringer, B. (2010). Moa: Massive online analysis. *Journal of Machine Learning Research*, *11*(May), 1601-1604.

Bifet, A., Holmes, G., Kirkby, R., & Pfahringer, B. (2011). MOA data stream mining-A practical approach. *COSI (Centre for Open Software Innovation)*.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993-1022.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Bunch, J. R., & Nielsen, C. P. (1978). Updating the singular value decomposition. *Numerische Mathematik*, *31*(2), 111-129.

Buntine, W. (1991). Theory refinement on Bayesian networks. In *Uncertainty Proceedings 1991* (pp. 52-60). Morgan Kaufmann.

Cao, F., Estert, M., Qian, W., & Zhou, A. (2006, April). Density-based clustering over an evolving data stream with noise. In *Proceedings of the 2006 SIAM international conference on data mining* (pp. 328-339). Society for Industrial and Applied Mathematics.

Cesa-Bianchi, N., & Gentile, C. (2008). Improved risk tail bounds for on-line algorithms. *IEEE Transactions on Information Theory*, *54*(1), 386-390.

Chang, J. H., & Lee, W. S. (2003, August). Finding recent frequent itemsets adaptively over online data streams. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 487-492).

Chang, J. H., & Lee, W. S. (2003, November). estWin: adaptively monitoring the recent change of frequent itemsets over online data streams. In *Proceedings of the twelfth international conference on Information and knowledge management* (pp. 536-539).

Charikar, M., Chen, K., & Farach-Colton, M. (2004). Finding frequent items in data streams. *Theoretical Computer Science*, *312*(1), 3-15.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

Cheng, J., Ke, Y., & Ng, W. (2008). A survey on algorithms for mining frequent itemsets over data streams. *Knowledge and Information Systems*, *16*(1), 1-27.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, *20*(1), 37-46.

Collobert, R., & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning* (pp. 160-167).

Crammer, K., & Singer, Y. (2003). Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, *3*(Jan), 951-991.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online passive-aggressive algorithms. *Journal of Machine Learning Research*, *7*(Mar), 551-585.

Crankshaw, D., Wang, X., Zhou, G., Franklin, M. J., Gonzalez, J. E., & Stoica, I. (2017). Clipper: A low-latency online prediction serving system. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)* (pp. 613-627).

Dawid, A. P. (1984). Present position and potential developments: Some personal views statistical theory the prequential approach. *Journal of the Royal Statistical Society: Series A (General)*, *147*(2), 278-290.

De Francisci Morales, G., Bifet, A., Khan, L., Gama, J., & Fan, W. (2016, August). Iot big data stream mining. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 2119-2120).

Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, *51*(1), 107-113.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., ... & Le, Q. V. (2012). Large scale distributed deep networks. In *Advances in neural information processing systems* (pp. 1223-1231).

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), 391-407.

Denil, M., Matheson, D., & Freitas, N. (2013, February). Consistency of online random forests. In *International conference on machine learning* (pp. 1256-1264).

Domingos, P., & Hulten, G. (2000, August). Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 71-80).

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning*, *29*(2-3), 103-130.

Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, No. 34, pp. 226-231).

Fan, W., & Bifet, A. (2013). Mining big data: current status, and forecast to the future. *ACM sIGKDD Explorations Newsletter*, *14*(2), 1-5.

Fan, W., Stolfo, S. J., & Zhang, J. (1999, August). The application of AdaBoost for distributed, scalable and on-line learning. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 362-366).

Fontenla-Romero, Ó., Guijarro-Berdiñas, B., Martinez-Rego, D., Pérez-Sánchez, B., & Peteiro-Barral, D. (2013). Online machine learning. In *Efficiency and Scalability Methods for Computational Intellect* (pp. 27-54). IGI Global.

Freund, Y., & Schapire, R. E. (1995, March). A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory* (pp. 23-37). Springer, Berlin, Heidelberg.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1, No. 10). New York: Springer series in statistics.

Friedman, N., & Goldszmidt, M. (2013). Sequential update of Bayesian network structure. *arXiv preprint arXiv:1302.1538*.

Frigó, E., Pálovics, R., Kelen, D., Kocsis, L., & Benczúr, A. (2017). Online ranking prediction in non-stationary environments.

Gaber, M. M., Zaslavsky, A., & Krishnaswamy, S. (2007). A survey of classification methods in data streams. In *Data streams* (pp. 39-59). Springer, Boston, MA.

Gama, J., Sebastião, R., & Rodrigues, P. P. (2009, June). Issues in evaluation of stream learning algorithms. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 329-338).

Gama, J., Sebastião, R., & Rodrigues, P. P. (2013). On evaluating stream learning algorithms. *Machine learning*, *90*(3), 317-346.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, *46*(4), 1-37.

Gomes, H. M., Read, J., Bifet, A., Barddal, J. P., & Gama, J. (2019). Machine learning for streaming data: state of the art, challenges, and opportunities. *ACM SIGKDD Explorations Newsletter*, *21*(2), 6-22.

Hartigan, J. A. (1975). *Clustering algorithms*. John Wiley & Sons, Inc..

Haselsteiner, E., & Pfurtscheller, G. (2000). Using time-dependent neural networks for EEG classification. *IEEE transactions on rehabilitation engineering*, *8*(4), 457-463.

Haykin, S. (1994). *Neural networks: a comprehensive foundation*. PrenticeHall PTR.

Henzinger, M. R., Raghavan, P., & Rajagopalan, S. (1998). Computing on data streams. *External memory algorithms*, *50*, 107-118.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, *29*(6), 82-97.

Hoffman, M., Bach, F. R., & Blei, D. M. (2010). Online learning for latent dirichlet allocation. In *advances in neural information processing systems* (pp. 856-864).

Honeine, P. (2011). Online kernel principal component analysis: A reduced-order model. *IEEE transactions on pattern analysis and machine intelligence*, *34*(9), 1814-1826.

Hulten, G., Spencer, L., & Domingos, P. (2001, August). Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 97-106).

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264-323.

Jin, R., & Agrawal, G. (2003, August). Efficient decision tree construction on streaming data. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 571-576).

Jolliffe, I.T.: Principal component analysis and factor analysis. In: Principal component analysis, pp. 115–128. Springer (1986)

Juang, C. F., & Lin, C. T. (1998). An online self-constructing neural fuzzy inference network and its applications. *IEEE transactions on Fuzzy Systems*, *6*(1), 12-32.

Jacob Montiel Lopez, thesis "Apprentissage automatique rapide et lent"

Kassambara, A. (2017). *Practical guide to cluster analysis in R: Unsupervised machine learning* (Vol. 1). STHDA.

Kavitha, V., & Punithavalli, M. (2010). Clustering time series data stream-a literature survey. *arXiv preprint arXiv:1005.4270*.

Kivinen, J., & Warmuth, M. K. (1997). Exponentiated gradient versus gradient descent for linear predictors. *information and computation*, *132*(1), 1-63.

Kotsiantis, S. B., Zaharakis, I., &Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*, 3-24.

Kourtellis, N., Morales, G.D.F., Bifet, A., Murdopo, A.: Vht: Vertical hoeffding tree. In: Big Data (Big Data), 2016 IEEE International Conference on, pp. 915–922. IEEE (2016)

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Lam, W., Liu, L., Prasad, S., Rajaraman, A., Vacheri, Z., & Doan, A. (2012). Muppet: MapReduce-style processing of fast data. *arXiv preprint arXiv:1208.4175*.

Langford, J., Li, L., & Zhang, T. (2009). Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, *10*(Mar), 777-801.

LeCun, Y., Bottou, L., Orr, G. B., & Müller, K. R. (1998). Neural networks: Tricks of the trade. *Springer Lecture Notes in Computer Sciences*, *1524*(5-50), 6.

Li, Y., Long, P.M.: The relaxed online maximum margin algorithm. Machine Learning 1(46), 361–387 (2002)

Lison, P. (2015). An introduction to machine learning. *Language Technology Group (LTG)*, *1*, *35*.

Low, Y., Gonzalez, J., Kyrola, A., Bickson, D., Guestrin, C., & Hellerstein, J. M. (2012). Distributed graphlab: A framework for machine learning in the cloud. *arXiv preprint arXiv:1204.6078*.

Mahdiraji, A. R. (2009). Clustering data stream: A survey of algorithms. *International Journal of Knowledge-based and Intelligent Engineering Systems*, *13*(2), 39-44.

Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). Machine learning. *Neural and Statistical Classification*, *13*.

Montiel, J., Read, J., Bifet, A., & Abdessalem, T. (2018). Scikit-multiflow: A multi-output streaming framework. *The Journal of Machine Learning Research*, *19*(1), 2915-2914.

Morales, G. D. F., & Bifet, A. (2015). SAMOA: scalable advanced massive online analysis. *Journal of Machine Learning Research*, *16*(1), 149-153.

Muthukrishnan, S. (2005). Data streams: Algorithms and applications. *Foundations and Trends® in Theoretical Computer Science*, *1*(2), 117-236.

Neumeyer, L., Robbins, B., Nair, A., & Kesari, A. (2010, December). S4: Distributed stream computing platform. In *2010 IEEE International Conference on Data Mining Workshops* (pp. 170-177). IEEE.

Obermaier, B., Guger, C., Neuper, C., & Pfurtscheller, G. (2001). Hidden Markov models for online classification of single trial EEG data. *Pattern recognition letters*, *22*(12), 1299-1309.

Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, *15*(3), 267-273.

Owen, S., Anil, R., Dunning, T., Friedman, E., & ACTION, M. I. (2011). Manning Publications.

Oza, N. C. (2005, October). Online bagging and boosting. In *2005 IEEE international conference on systems, man and cybernetics* (Vol. 3, pp. 2340-2345). Ieee.

Pálovics, R., Kelen, D., & Benczúr, A. A. (2017, August). Tutorial on open source online learning recommenders. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (pp. 400-401).

Pang-Ning, T., Steinbach, M., Kumar, V., et al.: Introduction to data mining. WP Co (2006)

Pearl, J. (2014). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, *12*(Oct), 2825-2830.

Quadrana, M., Bifet, A., & Gavalda, R. (2015). An efficient closed frequent itemset miner for the MOA stream mining system. *AI Communications*, *28*(1), 143-158.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81-106.

Read, J., Reutemann, P., Pfahringer, B., & Holmes, G. (2016). Meka: a multi-label/multi-target extension to weka. *The Journal of Machine Learning Research*, *17*(1), 667-671.

Reutemann, G. H. B. P. P., Hall, I. H. W. M., Frank, E., & Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explorations*, *11*(1), 10-18.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Schölkopf, B., Smola, A., & Müller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, *10*(5), 1299-1319.

Smola, A., & Narayanamurthy, S. (2010). An architecture for parallel topic models. *Proceedings of the VLDB Endowment*, *3*(1-2), 703-710.

Song, X., Lin, C. Y., Tseng, B. L., & Sun, M. T. (2005, August). Modeling and predicting personal information dissemination behavior. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining* (pp. 479-488).

Teflioudi, C., Gemulla, R., & Mykytiuk, O. (2015, May). Lemp: Fast retrieval of large entries in a matrix product. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (pp. 107-122).

Toshniwal, A., Taneja, S., Shukla, A., Ramasamy, K., Patel, J.M., Kulkarni, S., Jackson, J., Gade, K., Fu, M., Donham, J., et al.: Storm @ Twitter. In: Proceedings of the 2014 ACM SIGMOD international conference on Management of data, pp. 147–156. ACM (2014)

Vasiloudis, T., Beligianni, F., & De Francisci Morales, G. (2017, November). BoostVHT: Boosting distributed streaming decision trees. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (pp. 899-908).

White, T. (2010). Hadoop: The Definitive Guide, Yahoo.

Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine learning*, *23*(1), 69-101.

Zhu, X. J. (2005). *Semi-supervised learning literature survey*. University of Wisconsin-Madison Department of Computer Sciences.

Zhu, Y., & Shasha, D. (2002, January). Statstream: Statistical monitoring of thousands of data streams in real time. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases* (pp. 358-369). Morgan Kaufmann.

Zliobaite, I., Bifet, A., Gaber, M., Gabrys, B., Gama, J., Minku, L., & Musial, K. (2012). Next challenges for adaptive learning systems. *ACM SIGKDD Explorations Newsletter*, *14*(1), 48-55.