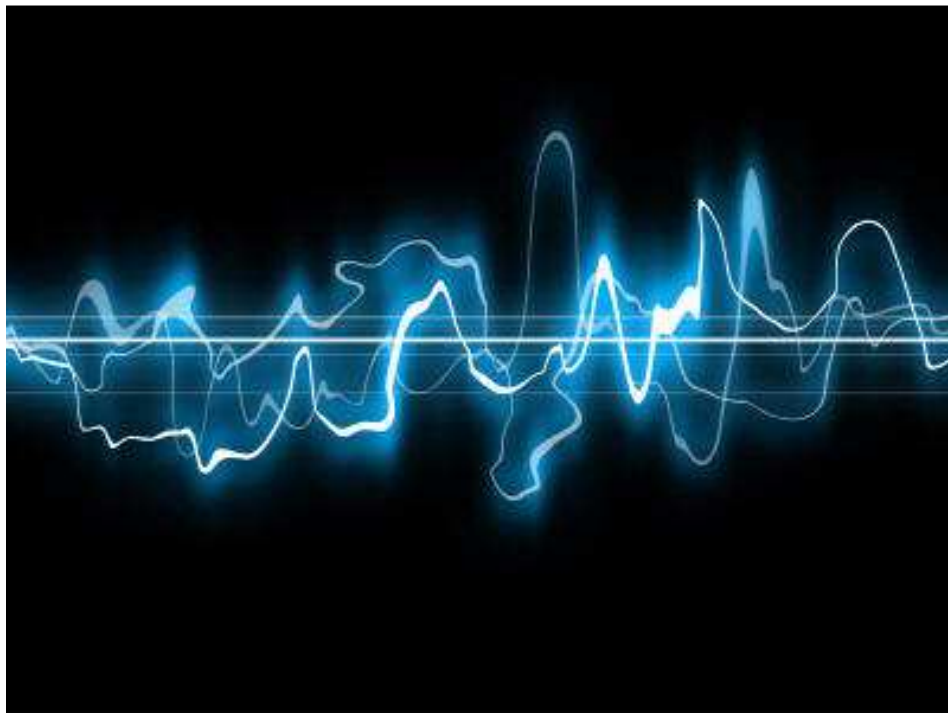


ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
ΠΑΡΑΡΤΗΜΑ ΡΕΘΥΜΝΟΥ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΜΟΥΣΙΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΑΚΟΥΣΤΙΚΗΣ Τ.Ε.



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΑΣ ΑΠΟ ΒΑΣΕΙΣ ΜΟΥΣΙΚΗΣ
ΠΛΗΡΟΦΟΡΙΑΣ ΜΕΓΑΛΗΣ ΚΛΙΜΑΚΑΣ**

ΣΠΟΥΔΑΣΤΗΣ

ΡΟΥΣΣΟΣ ΝΙΚΟΣ Α.Μ. 894

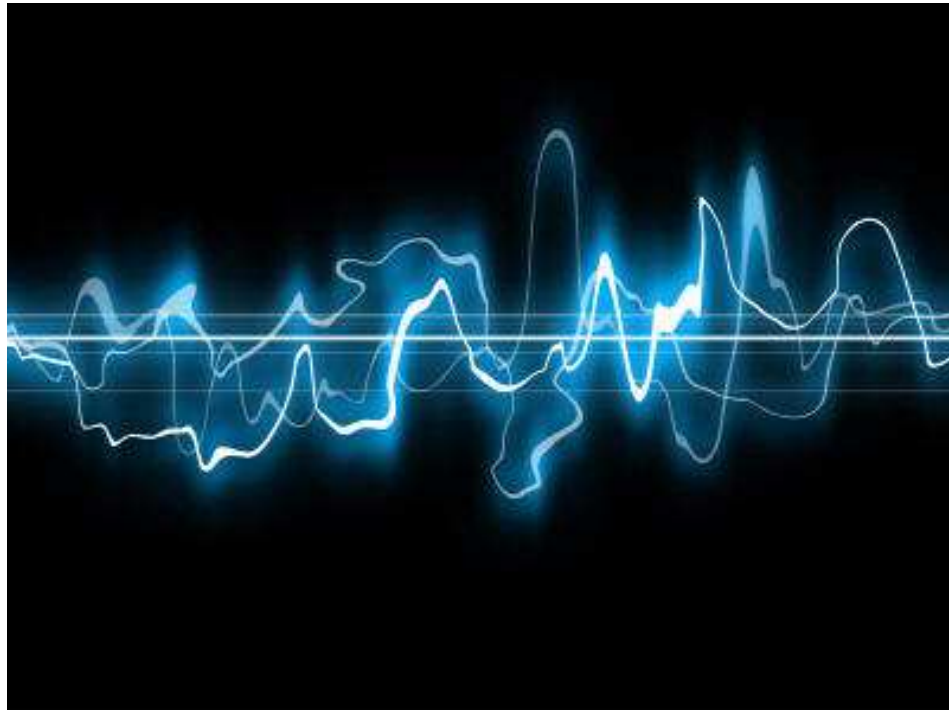
ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ

ΔΡ. ΖΕΡΒΑΣ ΠΑΝΑΓΙΩΤΗΣ

ΡΕΘΥΜΝΟ 2015



TECHNOLOGICAL EDUCATIONAL INSTITUTE OF CRETE
RETHYMNON BRANCH
DEPARTMENT OF MUSIC TECHNOLOGY & ACOUSTICS



DISSERTATION PROJECT

**INFORMATION RETRIEVAL FROM LARGE SCALE
MUSIC DATA BASES**

STUDENT

ROUSSOS NIKOS 894

PROFESSOR SUPERVISOR

DR. ZERVAS PANAGIOTIS

RETHIMNO 2015

Περίληψη

Με την πάροδο των χρόνων έχει υπάρξει μια θεαματική εξέλιξη της τεχνολογίας. Αυτό φυσικά δε θα μπορούσε να αφήσει ανεπηρέαστο και τον τομέα της μουσικής και του ήχου. Όσο η τεχνολογία εξελίσσεται και προοδεύει τόσο μεγαλώνει και το μέγεθος των ψηφιακών δεδομένων που πρέπει να διαχειριστούν. Έτσι δημιουργήθηκε η ανάγκη αυτά τα δεδομένα να κατηγοριοποιηθούν και να αποθηκευτούν ανά κατηγορίες στις λεγόμενες ψηφιακές βάσεις δεδομένων. Αυτή η αυξανόμενη ζήτηση και χρήση των ψηφιακών δεδομένων, με τα μουσικά δεδομένα να είναι αυτά που απασχολούν περισσότερο, έχει οδηγήσει στην ανάγκη για αναγνώριση, οργάνωση και αναζήτηση αυτού του τεράστιου υλικού. Αυτό είχε σαν αποτέλεσμα τη δημιουργία και ανάπτυξη ενός επιστημονικού και βιομηχανικού ενδιαφέροντος πάνω στην αναγνώριση και ανάκτηση των ψηφιακών δεδομένων. Έχει δημιουργηθεί λοιπόν ένας πολύ μεγάλος αριθμός εφαρμογών και συστημάτων πάνω σε αυτόν τον τομέα, όπου τα πιο διαδεδομένα αφορούν την μουσική και τον ήχο. Αυτά είναι και τα συστήματα τα οποία πραγματεύεται η παρούσα πτυχιακή.

Στην παρούσα πτυχιακή αναλύεται η εξέλιξη των συστημάτων αυτών, τις κατηγορίες τους και τον τρόπο λειτουργίας τους. Ξεκινώντας από την μηχανική μάθηση και συνεχίζοντας στην ανάκτηση μουσικών πληροφοριών και το ακουστικό αποτύπωμα, εστιάζει στις online βάσεις δεδομένων, την λειτουργία τους και το πώς μπορούν να χρησιμοποιηθούν και να αξιοποιηθούν από προγραμματιστές και απλούς χρήστες, καθώς και τους αλγόριθμους, τα εργαλεία και τις εφαρμογές που χρησιμοποιούνται για την ανάκτηση μουσικών δεδομένων από αυτές. Σκοπός της εργασίας είναι η κατανόηση της τεχνολογίας πάνω στην οποία βασίζεται η ανάκτηση ψηφιακών δεδομένων και πληροφοριών και η δημιουργία μιας web εφαρμογής που θα χρησιμοποιεί αυτήν την τεχνολογία. Για την δημιουργία αυτής της εφαρμογής θα χρησιμοποιηθούν εργαλεία δοσμένα μέσα από συγκεκριμένες εταιρίες που ειδικεύονται πάνω στο ζήτημα της ανάκτησης πληροφορίας και ο σκοπός των εταιριών αυτών είναι να αποθηκεύουν στις βάσεις τους αυτά τα δεδομένα και να βοηθούν τους χρήστες των εργαλείων αυτών στην εξέλιξη της τεχνολογίας της ανάκτησης ψηφιακής πληροφορίας.

Λέξεις κλειδιά : Ανάκτηση πληροφορίας, βάσεις δεδομένων, ακουστικό αποτύπωμα, μουσική αναγνώριση, δημιουργία ιστοσελίδας, αλγόριθμοι.

Abstract

With the passing of time there has been a spectacular evolution of technology. The fields of sound and music of course couldn't be unaffected from that evolution. As technology grows and advances, the size of the digital data that have to be managed grows too. Thus the need for categorization and storage of these data was created. For that reason digital data bases were created. That ongoing demand and use of the digital data, with music digital data to be the ones with the biggest demand, has led to the need for recognition, organizing and identification of this enormous material. That had as a result the creation and development of a scientific and industrial interest on the identification and retrieval of digital data. There has been created a number of applications and systems on that field, where the most common have to do with sound and music. These are the systems the presence thesis is about to analyze.

In the presence thesis we analyze the evolution of these systems, their categories and the way they work. Starting from the field of machine learning and continuing to music information retrieval and audio fingerprinting It focuses on the online data bases, their operation and how they can be used and exploited by developers and ordinary users, their functionality and how they can be used and be utilized by developers and simple users. Also the algorithms, tools and applications that are used to retrieve music data from them. Aim of this thesis is the understanding of the technology on which the digital data and information retrieval is based. Also the development of a web application that is going to use that technology. For the development of the application we are going to use tools given by specific companies that specialize on the matter of information retrieval and they aim on storing data in their data bases and also help the developers and users of these tools with the development of the digital information retrieval technology.

Key words : Information retrieval, databases, audio fingerprint, music recognition, web design, algorithms.

Πίνακας Περιεχομένων

Εισαγωγή	5
ΚΕΦΑΛΑΙΟ 1	6
1. Μηχανική Μάθηση (Machine Learning).....	6
1.1. Σχεδιασμός συστήματος Μηχανικής Μάθησης.....	8
1.2. Αλγόριθμοι.....	10
1.3. Μηχανική μάθηση, Βάσεις δεδομένων.....	13
1.4. Ηχητικά χαρακτηριστικά	13
1.5. Open source εφαρμογές	14
2. Ανάκτηση μουσικής πληροφορίας (Music Information Retrieval M.I.R.).....	15
2.1. Διεργασίες	16
2.2. Σχεδιασμός και αλγόριθμοι.....	18
2.3. Βάσεις δεδομένων ανάκτησης μουσικής πληροφορίας.....	19
2.4. Συστήματα	20
2.5. ISMIR και MIREX	21
ΚΕΦΑΛΑΙΟ 2.....	22
1. Ακουστικό αποτύπωμα (Audio Fingerprinting AF).....	22
1.1. Περιγραφή	22
1.2. Ορισμός του ακουστικού αποτυπώματος.....	22
1.3. Παράμετροι των ακουστικών αποτυπωμάτων	23
1.4. Αλγόριθμοι, Δομή και σχεδιασμός	25
1.4.1. Αποτελέσματα	31
1.4.2. Αλγόριθμος εξαγωγής.....	31
1.4.3. Προϋποθέσεις	32
1.4.4. Σχεδιασμός	33
1.5. State of the Art.....	34
1.5.1. Χρήσεις και εφαρμογές	34
1.5.2. Open source εφαρμογές	37
ΚΕΦΑΛΑΙΟ 3.....	42
1. Echo nest - Echoprint	42
1.1.1. Echo Nest.....	42
1.2. Εμπορική και ακαδημαϊκή χρήση.....	42
1.3. Υπηρεσίες.....	43
1.4. Τι είναι το echoprint.....	43
1.4.1. Περιγραφή λειτουργίας.....	44
1.4.2. Λειτουργίες του echoprint	46
2. Online Music Services and APIs.....	48
2.1. Κατηγορίες	48
2.2. Λειτουργία και παραδείγματα	49
3. Musicbrainz και audio fingerprinting.....	50

3.1. Musicbrainz	50
3.2. Βάση δεδομένων	51
3.3. Picard	52
3.4. Musicbrainz API και audio fingerprinting.....	52
3.5. Συνεργασίες.....	53
ΚΕΦΑΛΑΙΟ 4.....	55
1. ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ.....	55
1.1. Εισαγωγή.....	55
1.2. Doreso.....	55
1.2.1. Doreso API & API key.....	55
1.3. Εργαλεία	57
1.4. Υλοποίηση.....	59
2. Στατιστικά.....	63
2.1. Διασκευές (covers).....	71
1.3. Συμπεράσματα.....	74
ΚΕΦΑΛΑΙΟ 5.....	77
Επίλογος.....	77
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	78
Παράρτημα.....	83

Εισαγωγή

Η μουσική έχει διεισδύσει σε σχεδόν κάθε τομέα της ανθρώπινης δραστηριότητας σήμερα. Διαδίδεται μέσω του ραδιοφώνου και μέσω του διαδικτύου, χρησιμοποιείται ως υπόβαθρο σε βίντεο, τηλεόραση και ταινίες. Υπάρχει σε οτιδήποτε ένας άνθρωπος κάνει στην προσωπική του ζωή και μπορεί να την ανακαλύψει μέσα από τα εκατομμύρια τραγούδια που υπάρχουν στο διαδίκτυο, να την μεταφέρει μέσα από ηλεκτρονικές συσκευές αναπαραγωγής και κινητά τηλέφωνα. Όλες αυτές οι συσκευές δίνουν την δυνατότητα ο χρήστης τους να μπορεί να δει και να ξέρει ανά πάσα στιγμή πιο μουσικό κομμάτι ακούει. Τι γίνεται όμως όταν βρίσκεται σε ένα περιβάλλον που δεν του δίνει αυτή τη δυνατότητα; Είναι πολύ συχνό κάποιος να ακούει ένα τραγούδι και να ψάχνει διακαώς να μάθει πιο είναι ή να προσπαθεί να θυμηθεί τον καλλιτέχνη που το ερμηνεύει. Με τον τεράστιο όγκο τραγουδιών που υπάρχει είναι αδύνατον για κάποιον να μπορεί να ξέρει και να αναγνωρίζει κάθε τραγούδι που ακούει. Για αυτόν τον λόγο αυτή η εργασία ανατέθηκε στους υπολογιστές.

Η δημιουργία εφαρμογών που σχετίζονται με τη μουσική μέσω των υπολογιστών πάει πίσω στην εποχή που τα πρώτα ερευνητικά περιβάλλοντα έγιναν διαθέσιμα. Η μουσική ανάκτηση πληροφορίας (music information retrieval) είναι ένα παράδειγμα και σχετίζεται με την χρήση υπολογιστών ώστε να αποθηκεύονται και να αναλύονται ψηφιακές συλλογές μουσικής σε κάθε μορφή και είδος και μέσα από την ανάλυση να επιστρέφονται τα ζητούμενα δεδομένα. Κατά την διαδικασία του ακουστικού αποτυπώματος (audio fingerprinting) ένας αλγόριθμος αναλύει διαφορετικά μέρη ενός μουσικού κομματιού, αναγνωρίζει μοναδικά χαρακτηριστικά του που μπορούν να χρησιμοποιηθούν, του συγκεκριμένου κομματιού και όχι μόνο. Μόλις δημιουργηθεί ένα σύνολο αποτυπωμάτων ο αλγόριθμος μπορεί να τα χρησιμοποιήσει ώστε να δημιουργήσει το ολοκληρωμένο αποτύπωμα του αγνώστου κλιπ ήχου. Στην πορεία γίνεται αναζήτηση σε μια βάση δεδομένων για αποτυπώματα που είναι παρόμοια με αυτό ώστε να επιστραφούν οι πληροφορίες για την συγκεκριμένη ηχογράφιση. Όλες αυτές οι εφαρμογές βεβαία ξεκινάν από τον τομέα της μηχανικής μάθησης (machine learning) δηλαδή της επιστήμης που ως σκοπό έχει την εκμάθηση των μηχανών να λειτουργούν όπως ένας άνθρωπος. Όλοι αυτοί οι διαφορετικοί τομείς αναλύονται στα επόμενα κεφάλαια της συγκεκριμένης πτυχιακής.

Κεφάλαιο 1

1. Μηχανική Μάθηση (Machine Learning)

Ο τομέας της μηχανικής μάθησης εξελίχθηκε μέσα από το ευρύ πεδίο της τεχνητής νοημοσύνης, και ως σκοπό της έχει οι μηχανές να μπορέσουν να μιμηθούν τις λογικές ικανότητες του ανθρώπου. Η μηχανική μάθηση (ML) είναι ένα σύνολο μεθόδων στατιστικής ανάλυσης που καταφέρνει να επιτύχει τα οποιαδήποτε καθήκοντα μαθαίνοντας μέσω παραδειγμάτων. Το πεδίο είναι συνδεδεμένο με τομείς όπως το Data Mining, τεχνικές που ανακαλύπτουν άγνωστες δομές από δεδομένα, και τεχνικές αναγνώρισης μοτίβων (Pattern Recognition), τα οποία αναγνωρίζουν μοτίβα μέσα από δοσμένα στοιχεία βασισμένα σε μια πιθανότητα να ταιριάζουν με προϋπάρχοντα δεδομένα. οι μέθοδοι της μηχανικής μάθησης διακρίνονται από τους τρόπους μάθησης που επιτρέπουν το συμπέρασμα και τη γενίκευση.

Η μηχανική μάθηση έχει αναπτυχθεί γύρω από τις ακόλουθες ερευνητικές γραμμές:

- Εξειδικευμένες μελέτες(task-oriented studies) π.χ. ανάπτυξη συστημάτων μάθησης για τη βελτίωση των επιδόσεων σε μια προκαθορισμένη σειρά εργασιών.
- Γνωστική προσομοίωση(Cognitive simulation), είναι η έρευνα και η προσομοίωση των διαδικασιών μάθησης των ανθρώπων μέσω υπολογιστή.
- Θεωρητική ανάλυση(Theoretical analysis), π.χ. η θεωρητική διερεύνηση των πιθανών αλγορίθμων και μεθόδων μάθησης, ανεξαρτήτου εφαρμοσμένου πεδίου.

Ένας πιο γενικός ορισμός δίνεται από τον Tom Mitchell το 1997 όπου ορίζει τη μηχανική μάθηση ως : «Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από μια εμπειρία E σε σχέση μια σειρά από έργα T και μια μέτρηση της απόδοσης P ή οποία βελτιώνεται με την εμπειρία E » [3]

Η μηχανική μάθηση χρησιμοποιεί τη θεωρία της στατιστικής ώστε να χτίζει μαθηματικά μοντέλα, αφού το βασικό ζήτημα είναι να βγάξει συμπεράσματα χρησιμοποιώντας δείγματα. Έχει οριστεί λοιπόν ένα μοντέλο βάση κάποιων παραμέτρων. Η μάθηση είναι η εκτέλεση ενός προγράμματος υπολογιστή ώστε μέσα από αυτό να βελτιστοποιηθούν οι παράμετροι του μοντέλου χρησιμοποιώντας τα εκπαιδευτικά δεδομένα και την παλαιότερη εμπειρία. Το μοντέλο μπορεί να είναι προγνωστικό ώστε να κάνει προγνώσεις στο μέλλον ή περιγραφικό ώστε να αποκτά γνώση από τα δεδομένα, ή και τα δυο.

Η μηχανική μάθηση ανάλογα το επιθυμητό αποτέλεσμα του αλγορίθμου ή τον τύπο της εισόδου που χρησιμοποιείται κατά τη διάρκεια της εκπαίδευσης χωρίζεται στις εξής κατηγορίες :[11]

Επιτηρούμενη μάθηση, επιβλεπόμενη μάθηση ή μάθηση με επίβλεψη (supervised learning), όπου ο αλγόριθμος κατασκευάζει μια συνάρτηση που απεικονίζει δεδομένες εισόδους σε γνωστές επιθυμητές εξόδους (σύνολο

εκπαίδευσης), με απώτερο στόχο τη γενίκευση της συνάρτησης αυτής και για εισόδους με άγνωστη έξοδο (σύνολο ελέγχου).

Μη επιτηρούμενη μάθηση, ανεπίβλεπτη μάθηση ή μάθηση χωρίς επίβλεψη (unsupervised learning), όπου ο αλγόριθμος κατασκευάζει ένα μοντέλο για κάποιο σύνολο εισόδων χωρίς να γνωρίζει επιθυμητές εξόδους για το σύνολο εκπαίδευσης.

Ενισχυτική μάθηση (reinforcement learning), όπου ο αλγόριθμος μαθαίνει μια στρατηγική ενεργειών για μια δεδομένη παρατήρηση.

Ημι-επιβλεπόμενη μάθηση (Semi-supervised learning) συνδυάζει περιγραφικά και μη περιγραφικά παραδείγματα για να παράγει τη κατάλληλη λειτουργία ή ταξινόμηση.

Μεταγωγή ή μεταβιβαστική συμπερασματολογία (Transduction ή transductive inference), προσπαθεί να προβλέψει νέες εξόδους σε συγκεκριμένες και σταθερές (test) περιπτώσεις από συγκεκριμένες παρατηρημένες (training) περιπτώσεις.

Ενισχυτική μάθηση (Reinforcement learning) ασχολείται με το πώς οι λογικοί παράγοντες πρέπει να ενεργούν σε ένα περιβάλλον ώστε να μεγιστοποιούν κάποια πιθανότητα ανταμοιβής. Οι παράγοντες εκτελούν ενέργειες οι οποίες προκαλούν την αισθητή κατάσταση του περιβάλλοντος να αλλάξει. Μέσα από μια σειρά ενεργειών, ο παράγοντας προσπαθεί να συγκεντρώσει τη γνώση για το πώς το περιβάλλον ανταποκρίνεται στις ενέργειες του, και επιχειρεί να συνθέσει μια σειρά από ενέργειες που μεγιστοποιεί μια αθροιστική ανταμοιβή.

Μαθαίνοντας να μαθαίνει (Learning to learn) μαθαίνει τη δικιά του επαγωγική κλίση βασισμένο σε προηγούμενη εμπειρία.

Αναπτυξιακή μάθηση (Developmental learning), εκπονημένη για ρομποτική μάθηση, δημιουργεί τις δικές της σειρές από καταστάσεις μάθησης ώστε να αποκτήσει αθροιστικά ρεπερτόρια από νέες δεξιότητες μέσα από αυτόνομη αυτό-αναζήτηση, κοινωνική αλληλεπίδραση με ανθρώπους εκπαιδευτικούς, και με τη χρήση μηχανισμών επαγγελματικού προσανατολισμού, όπως η ενεργός μάθηση, η ωρίμανση και η μίμηση.

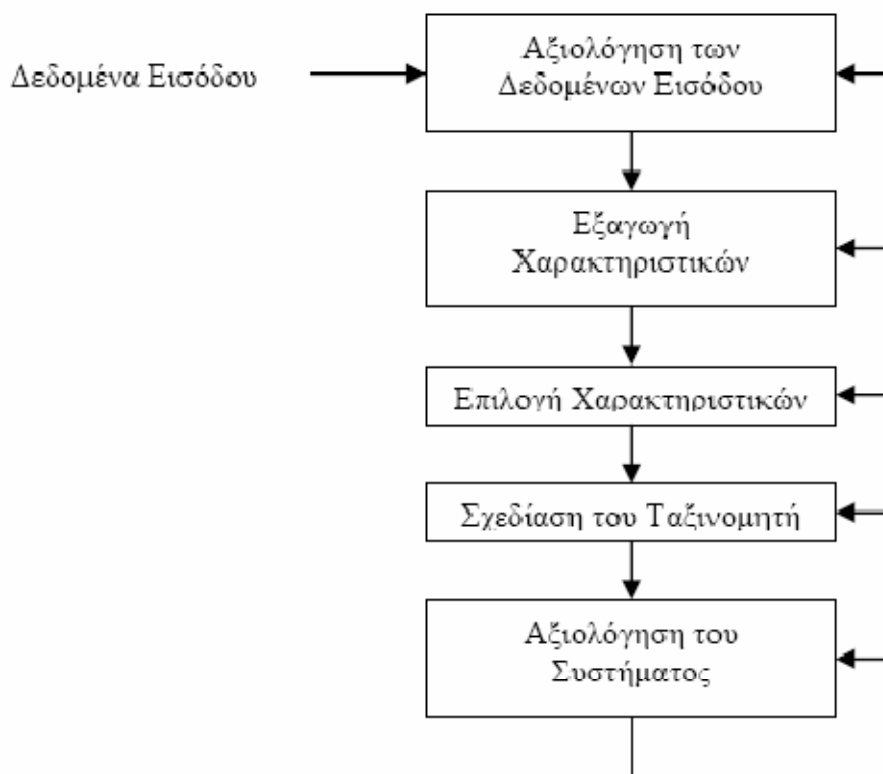
Με την επιτηρούμενη και τη μη επιτηρούμενη μάθηση να είναι οι πιο βασικές και δημοφιλείς τεχνικές, καθώς και αυτές για τις οποίες θα γίνει αναφορά παρακάτω.

Η μηχανική μάθηση βρίσκει εφαρμογή σε ένα μεγάλο εύρος ζητημάτων και εφαρμογών όπως ιατρική διάγνωση, αναγνώριση λόγου και γραφικού χαρακτήρα, μηχανές αναζήτησης, βιοπληροφορική, αναγνώριση και ανάλυση συναισθήματος. Η οπτική αναγνώριση χαρακτήρων, στην οποία τυπωμένοι χαρακτήρες αναγνωρίζονται αυτόματα βασισμένοι σε προηγούμενα παραδείγματα είναι ένα κλασικό παράδειγμα μηχανικής μάθησης.

1.1. Σχεδιασμός συστήματος Μηχανικής Μάθησης

Το πρώτο στάδιο της σχεδίασης ενός συστήματος μηχανικής μάθησης πραγματεύεται τον προσδιορισμό της γνώσης που θα χρησιμοποιηθεί κατά την εκπαίδευση του. Όπως είναι φυσικό η απόδοση του συστήματος εξαρτάται άμεσα από αυτές τις επιλογές. Μια από τις σημαντικότερες επιλογές όσον αφορά τη παροχή της γνώσης είναι το κατά πόσο η γνώση αυτή προσφέρει στο σύστημα άμεση ή έμμεση πληροφόρηση όσον αφορά την εκμάθηση, αναγκάζοντας το να προχωρήσει μόνο του στην ανακάλυψη της. Ιδιαίτερη σημασία δίνεται στην κατανομή των παραδειγμάτων από τα όποια θα αντληθεί η γνώση από το σύστημα μηχανικής μάθησης, καθώς θα πρέπει να είναι όμοια με την αντίστοιχη κατανομή των περιπτώσεων που θα χρησιμοποιηθούν κατά τον έλεγχο της απόδοσης, και ακόμα περισσότερο κατά τη λειτουργία του. Η συνηθέστερη μορφή αναπαράστασης των παραδειγμάτων είναι η διανυσματική. Κάθε παράδειγμα εκπαίδευσης ή στιγμιότυπο (instance) αποτελείται από ένα σύνολο χαρακτηριστικών (features), που έχουν επιλεγεί με τέτοιο τρόπο από τα δεδομένα εκπαίδευσης ώστε το σύστημα να μπορέσει να λάβει από τις τιμές τους τη γνώση που χρειάζεται. Ανάλογα με το είδος της πληροφορίας που κωδικοποιεί ένα χαρακτηριστικό, αυτό μπορεί να είναι συνεχές ή ονομαστικό. Με αυτόν τον τρόπο, έχοντας επιλέξει λ χαρακτηριστικά, απεικονίζουμε τον χώρο του προβλήματος μας σε έναν λ -διάστατο χώρο, τον χώρο των στιγμιότυπων (instance space), αντιστοιχώντας κάθε στιγμιότυπο εκπαίδευσης σε ένα διάνυσμα λ διαστάσεων. Επόμενο στάδιο στη σχεδίαση του συστήματος αποτελεί ο τρόπος με τον οποίο θα διαχειριστεί το τελευταίο τη γνώση που αποκομίζει κατά την εκπαίδευση, με σκοπό την αποδοτικότερη λειτουργία του. Υιοθετώντας το μοντέλο της διανυσματικής αναπαράστασης, αναζητούμε ουσιαστικά μια συνάρτηση f' η οποία προσεγγίζει όσο το δυνατόν περισσότερο μια ιδανική συνάρτηση με την οποία είναι δυνατόν να μοντελοποιηθεί το πρόβλημα, τη συνάρτηση f . Οι δύο συναρτήσεις έχουν ως ελεύθερη μεταβλητή ένα τυχαίο διάνυσμα x , με πεδίο ορισμού τον χώρο των στιγμιότυπων και σύνολο τιμών το οποίο καθορίζεται από την εκάστοτε εφαρμογή. [3] [1]

Στην εικόνα 1 εμφανίζονται συνοπτικά τα βασικά στάδια της σχεδίασης ενός συστήματος μηχανικής μάθησης



Εικόνα 1: Βήματα διαδικασίας Σχεδιασμού ενός συστήματος Μηχανικής μάθησης

1.1.1. Ταξινόμηση (Classification)

Η ταξινόμηση είναι η διαδικασία κατά την οποία παίρνεται η απόφαση σε ποια κατηγορία ανήκει μια ομάδα δεδομένων, και κατά την οποία δημιουργούνται δείγματα ζευγαριών από μεταβλητές εισόδου με ετικέτες, έτσι ώστε να δημιουργηθούν κατηγορίες. Οι κατηγορίες που δημιουργούνται κατά την ταξινόμηση είναι διακριτές και αποτελούνται από ακέραιους αριθμούς ή ετικέτες χορδών. Έτσι δημιουργείται μια συνάρτηση μεταξύ συνεχών μεταβλητών εισόδου και διακριτών σημάτων. Οι διαδικασίες εκμάθησης βασίζονται σε γνωστές μεθόδους εκπαίδευσης και αποτελούν παράδειγμα επιτηρούμενης μάθησης. Οι αλγόριθμοι οι οποίοι χρησιμοποιούνται κατά τη διαδικασία της ταξινόμησης ονομάζονται *ταξινομητές*. (1)

1.1.2. Επιτηρούμενη μάθηση

Η επιτηρούμενη μάθηση παίρνει ένα γνωστό σετ από δεδομένα εισόδου και γνωστές αντιδράσεις στα δεδομένα αυτά, και προσπαθεί να κτίσει ένα μοντέλο πρόβλεψης που παράγει λογικές προβλέψεις για την διαχείριση νέων δεδομένων. Η επιτηρούμενη μάθηση είναι αρκετά κοινή σε προβλήματα ταξινόμησης καθώς ο στόχος είναι συχνά ο υπολογιστής να εξαναγκαστεί να μάθει ένα σύστημα ταξινόμησης που έχουμε δημιουργήσει. Είναι η πιο κοινή τεχνική για την εκπαίδευση νευρωνικών δικτύων (neural networks) και δέντρων απόφασης (decision trees) καθώς οι δυο αυτές οι τεχνικές εξαρτώνται σε μεγάλο βαθμό από τις πληροφορίες που δίνονται από τις προκαθορισμένες ταξινομήσεις. Στη περίπτωση των neural networks, η ταξινόμηση χρησιμοποιείται για τον προσδιορισμό του σφάλματος του δικτύου και στη συνέχεια ρυθμίζει το δίκτυο για να το ελαχιστοποιήσει. Στα decision trees, οι ταξινομήσεις χρησιμοποιούνται για να προσδιορίσουν ποια χαρακτηριστικά

παρέχουν τις περισσότερες πληροφορίες που μπορούν να χρησιμοποιήσουν ώστε να λυθεί το πάζλ της ταξινόμησης. Η διαδικασία εφαρμογής εποπτευόμενης μηχανικής μάθησης πραγματοποιείται σε δυο βήματα. Το πρώτο βήμα είναι η συλλογή των δεδομένων. Το δεύτερο βήμα είναι η προετοιμασία των δεδομένων και η προ επεξεργασία τους. Ανάλογα με τις περιστάσεις, οι ερευνητές έχουν μια σειρά από μεθόδους για να επιλέξουν ώστε να χειριστούν τα δεδομένα που λείπουν (Batista, 2003).

1.1.3. Μη - επιτηρούμενη μάθηση

Η μη- επιτηρούμενη μάθηση φαίνεται να είναι πιο δύσκολη. Ο στόχος είναι ο υπολογιστής να κάνει κάτι που δεν του έχει υποδειχτεί πώς να το κάνει. Υπάρχουν δυο προσεγγίσεις στη μη- επιτηρούμενη μάθηση. Η πρώτη προσέγγιση είναι να διδάξουμε τον παράγοντα με τη χρήση κάποιου είδους συστήματος ανταμοιβής που να δηλώνει πότε υπάρχει επιτυχία. Ένας δεύτερος τύπος μη- επιτηρούμενης μάθησης ονομάζεται clustering. Σε αυτό το είδος της μάθησης, ο στόχος δεν είναι να μεγιστοποιηθεί μια πρακτική λειτουργία, αλλά απλώς να βρει ομοιότητες στα δεδομένα εκπαίδευσης. Οι αλγόριθμοι της μη-επιτηρούμενης μάθησης συμφώνα με τον Ghahramani (Ghahramani, 2008) έχουν σχεδιαστεί έτσι ώστε να εξάγουν δομές από δείγματα δεδομένων.

1.2. Αλγόριθμοι

Οι διάφοροι επαγωγικοί αλγόριθμοι μάθησης (για συντομία EAM) που έχουν αναπτυχθεί έως σήμερα διαφέρουν ως προς την υποκείμενη αναπαράσταση του χώρου των δυνατών υποθέσεων, και κατά συνέπεια και του τρόπου που οργανώνουν την αναζήτηση σε αυτόν τον χώρο. Μερικά παραδείγματα αναπαραστάσεων είναι οι γραμμικοί συνδυασμοί, οι λογικές περιγραφές (λογικοί τύποι), τα δέντρα απόφασης (decision trees), τα τεχνητά νευρωνικά δίκτυα (artificial neural networks), κ.α. Διαφορετικές αναπαραστάσεις είναι κατάλληλες για τη μάθηση διαφορετικών ειδών συναρτήσεων-στόχων. Για κάθε μια από αυτές τις αναπαραστάσεις, ο αντίστοιχος EAM εκμεταλλεύεται τη διαφορετική υποκείμενη δομή για να οργανώσει την αναζήτηση στο χώρο των υποθέσεων. Μία θεμελιώδης ιδιότητα που χαρακτηρίζει κάθε EAM είναι η *επαγωγική προδιάθεση* ή *κλίση* του (*inductive bias*).[1] [3] [11]

1.2.1 Τύποι αλγορίθμων

Στους τομείς της εποπτευόμενης και μη εποπτευόμενης μάθησης μπορούμε να βρούμε τους παρακάτω τύπους αλγορίθμων : [52] [68] [8]

- Linear Classifiers
 - Logical Regression
 - Naïve Bayes Classifier
 - Perceptron
 - Support Vector Machine

- Quadratic Classifiers
- K-Means Clustering
- Boosting
- Decision Tree
- Neural networks

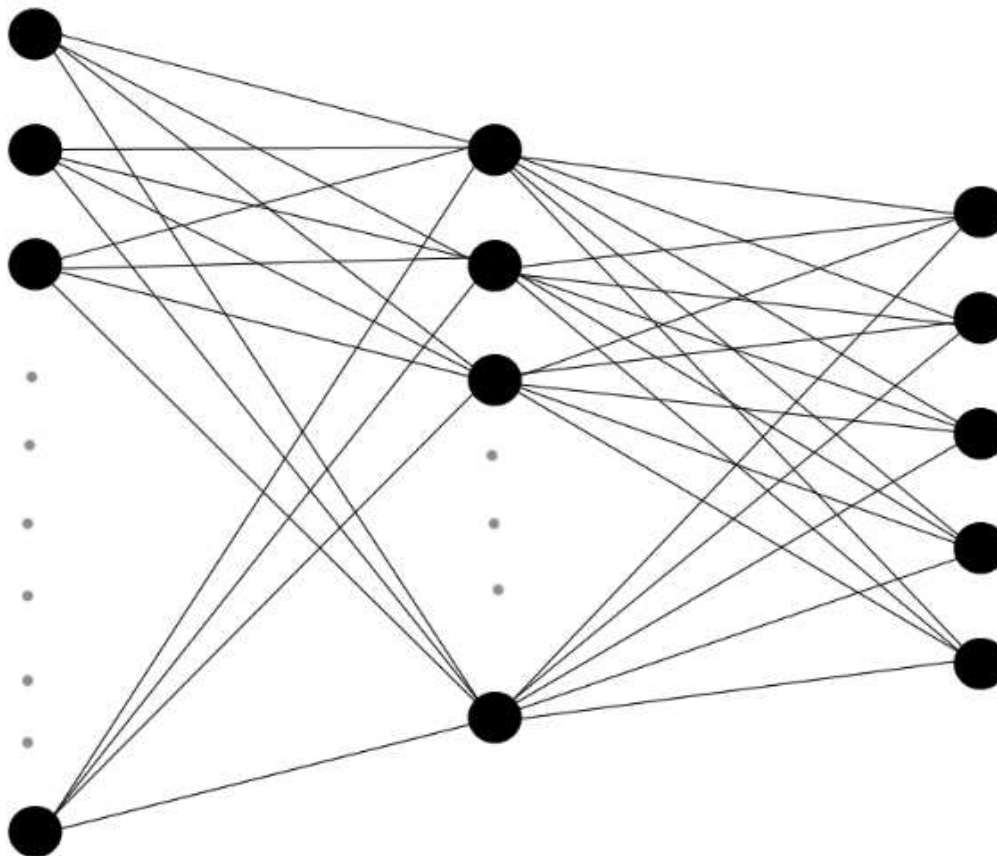
- Bayesian Networks
- Artificial neural network
- Data clustering
- Expectation-maximization algorithm
- Self-organizing map
- Radial basis function network
- Vector Quantization
- Generative topographic map
- Information bottleneck method
- IBSEAD

Τα Support Vector Machines, (SVMs, ή support vector networks,) είναι μοντέλα εποπτευόμενης μάθησης με σχετικούς αλγόριθμους εκμάθησης που αναλύουν δεδομένα και αναγνωρίζουν μοτίβα που χρησιμοποιούνται για ταξινόμηση και ανάλυση παλινδρόμησης. Το βασικό SVM παίρνει ένα σετ από δεδομένα εισόδου και προβλέπει, για κάθε δοσμένη είσοδο, ποια από τις δύο πιθανές κατηγορίες αποτελεί την έξοδο, κάνοντας τη έναν μη πιθανοτικό δυαδικό γραμμικό ταξινομητή. Δίνοντας του ένα σετ από αλγόριθμους εκπαίδευσης, καθένα προορισμένο να ανήκει σε μια από τις δυο κατηγορίες, ένας αλγόριθμος εκπαίδευσης SVM χτίζει ένα μοντέλο που αναθέτει νέα παραδείγματα στη μια κατηγορία ή την άλλη . Ένα μοντέλο SVM είναι μια αναπαράσταση των παραδειγμάτων ως σημεία στον χώρο, αντιστοιχισμένα έτσι ώστε τα παραδείγματα των επιμέρους κατηγοριών να διαχωρίζονται από ένα σαφές κενό το οποίο είναι όσο το δυνατόν πιο ευρύ. Τα νέα παραδείγματα στη συνέχεια αντιστοιχίζονται μέσα στο ίδιο κενό και η κατηγορία στην οποία θα ανήκουν προβλέπεται μέσα από το σε ποια μεριά του κενού θα πέσουν.

Ένα μπαεσιανό δίκτυο (Bayesian network), είναι ένα πιθανολογικό γραφικό μοντέλο (ένα είδους στατιστικού μοντέλου) το οποίο αναπαριστά ένα σετ από τυχαίες μεταβλητές και τις εξαρτήσεις τους μέσω ενός κατευθυνόμενου ακυκλικού συστήματος. Σαν παράδειγμα θα μπορούσαμε να δώσουμε τις πιθανολογικές σχέσεις μεταξύ των ασθενειών και των συμπτωμάτων που προκαλούν. Δίνοντας τα συμπτώματα, το δίκτυο μπορεί να χρησιμοποιηθεί για να υπολογίσει τις πιθανότητες της παρουσίας διαφόρων ασθενειών. Πιο επίσημα τα Bayesian networks είναι κατευθυνόμενα ακυκλικά γραφήματα των οποίων οι κόμβοι αποτελούν τυχαίες μεταβλητές στην Μπεϋζιανή λογική: μπορούν να είναι παρατηρήσιμες ποσότητες, λανθάνουσες μεταβλητές, άγνωστες παράμετροι ή υποθέσεις .

Ένα artificial neural network, ή πιο απλά neural network, είναι ένα μαθηματικό μοντέλο εμπνευσμένο από τα βιολογικά νευρωνικά δίκτυα. Ένα νευρωνικό δίκτυο αποτελείται από μια αλληλοσυνδεδεμένη ομάδα τεχνητών νευρώνων, και επεξεργάζεται τις πληροφορίες χρησιμοποιώντας μια συνδυαστική προσέγγιση για τον υπολογισμό. Στις περισσότερες περιπτώσεις, ένα νευρωνικό δίκτυο είναι ένα προσαρμοστικό σύστημα που αλλάζει τη δομή του κατά τη διάρκεια της εκμάθησης. Τα νευρωνικά δίκτυα χρησιμοποιούνται για τη μοντελοποίηση σύνθετων σχέσεων μεταξύ εισόδων και εξόδων ή για να βρουν μοτίβα στα δεδομένα. Με τον όρο δίκτυο μπορούμε να περιγράψουμε τις διαφορετικές διασυνδέσεις των νευρώνων στα

διάφορα στρώματα του συστήματος. Για παράδειγμα: ένα σύστημα έχει 3 στρώματα. Το πρώτο στρώμα αποτελείται από τους νευρώνες εισόδου, που στέλνουν δεδομένα μέσω συνδέσεων στο δεύτερο στρώμα των νευρώνων, και στη συνέχεια μέσω περισσότερων συνάψεων στο τρίτο στρώμα των νευρώνων εξόδου. Πιο πολύπλοκα συστήματα θα έχουν περισσότερα στρώματα των νευρώνων. Οι συνδέσεις αποθηκεύουν παραμέτρους που ονομάζονται άξεις (weights) οι οποίες χειρίζονται τα δεδομένα στους υπολογισμούς.



Εικόνα 2 : Συνδέσεις Νευρωνικού δικτύου [79]

Από μαθηματική άποψη, η συνάρτηση ενός νευρωνικού δικτύου $f(x)$ ορίζεται ως μια σύνθεση από άλλες συναρτήσεις $g_i(x)$, οι οποίες μπορούν περαιτέρω να οριστούν ως μια σύνθεση άλλων συναρτήσεων. Αυτό μπορεί να παρουσιαστεί ως μια δομή δικτύου, με βέλη να υποδεικνύουν τις εξαρτήσεις μεταξύ των μεταβλητών

Decision Trees. Η μέθοδος του δέντρου απόφασης, χρησιμοποιεί ένα μοντέλο πρόβλεψης το οποίο δημιουργεί έναν χάρτη μέσα από τις παρατηρήσεις πάνω σε ένα στοιχείο ώστε να βγάλει συμπεράσματα σχετικά με τον αρχικό στόχο της εκτίμησης του. Η δομή του δίνει την εικόνα ενός δέντρου όπου τα φύλλα αντιπροσωπεύουν ετικέτες μιας κατηγορίας και τα κλαδιά συζεύξεις διαφόρων χαρακτηριστικών που οδηγούν σε αυτές τις ετικέτες. Σαν σκοπό έχει τη δημιουργία ενός μοντέλου που προβλέπει την αξία της μεταβλητής ενός στόχου που βασίζεται

σε διάφορες μεταβλητές εισόδου. Ένας ταξινομητής τυχαίου δάσους όπως ονομάζεται, χρησιμοποιεί έναν αριθμό από decision trees, με σκοπό τη βελτίωση του βαθμού ταξινόμησης.

1.3. Μηχανική μάθηση, Βάσεις δεδομένων

Σύμφωνα με τη διαχείριση βάσης δεδομένων, μια βάση δεδομένων είναι μια λογικά ολοκληρωμένη συλλογή από δεδομένα τα οποία διατηρούνται σε ένα ή περισσότερα αρχεία και οργανώνονται για να διευκολυνθεί η αποτελεσματική αποθήκευση, τροποποίηση, και ανάκτηση των σχετικών πληροφοριών. Για παράδειγμα σε μια σχεσιακή βάση δεδομένων τα δεδομένα οργανώνονται σε φάκελους ή πίνακες σταθερού μήκους καταγραφών, όπου η κάθε καταγραφή είναι μια διατεταγμένη λίστα τιμών με μια τιμή για κάθε πεδίο. Οι πληροφορίες σχετικά με το όνομα του κάθε πεδίου και των πιθανών τιμών διατηρούνται σε ένα ξεχωριστό αρχείο που ονομάζεται λεξικό δεδομένων. Ένα σύστημα διαχείρισης βάσης δεδομένων είναι μια συλλογή από διαδικασίες σχετικά με ανάκτηση, αποθήκευση και εκμετάλλευση δεδομένων μέσα στις βάσεις. Στην μηχανική μάθηση, ο όρος βάση δεδομένων αναφέρεται σε μια συλλογή από υποδείξεις ή παραδείγματα που υπάρχουν μέσα σε ένα συγκεκριμένο αρχείο. Ένας αλγόριθμος μάθησης παίρνει τα δεδομένα και τις πληροφορίες που τα συνοδεύουν ως είσοδο και επιστρέφει μια εντολή που εκπροσωπεί τα αποτελέσματα της μάθησης ως έξοδο. [9]

Στο διαδίκτυο μπορούμε να βρούμε πληθώρα από ιστοσελίδες οι οποίες λειτουργούν σαν βάσεις δεδομένων- αποθήκες για τη μηχανική μάθηση. Κάποιες από αυτές είναι : το UC Irvine Machine Learning Repository το οποίο διατηρεί 264 σύνολα δεδομένων ελεύθερα προς τη κοινότητα, Delve datasets το οποίο διατηρεί συλλογές δεδομένων για την ανάπτυξη, την αξιολόγηση και σύγκριση των μεθόδων μάθησης, libml ή αλλιώς Machine Learning Library, η οποία είναι γραμμένη σε OCaml και ο σκοπός της είναι να αναδιοργανώσει τις εφαρμογές των διαφόρων αλγορίθμων για μηχανική μάθηση. Στη συνέχεια έχουμε το mldata όπου ο κάθε ενδιαφερόμενος μπορεί να ανεβάσει ή να βρει ελεύθερα δεδομένα και να κάνει σύγκριση στις διάφορες μεθόδους. Τέλος υπάρχει το openML το οποίο πρόκειται για μια ελεύθερη επιστημονική πλατφόρμα σχετικά με τη μηχανική μάθηση. Σε αυτή τη πλατφόρμα κάποιος μπορεί να αναζητήσει και να συγκρίνει παλαιότερες μελέτες, να μοιραστεί πειράματα αλγορίθμων και δεδομένα. Παρόλο που όπως βλέπουμε υπάρχει ένας μεγάλος αριθμός από βάσεις δεδομένων στο διαδίκτυο, το αρνητικό είναι ότι λίγες από αυτές προσφέρουν έναν επαρκή αριθμό αλγορίθμων για τον μέσο χρήστη καθώς και λίγες είναι αυτές που προσφέρουν εύρος στην αναζήτηση.

1.4. Ηχητικά χαρακτηριστικά

Η μελέτη της μηχανικής μάθησης στον τομέα του ήχου και της μουσικής καλύπτει ένα μεγάλο εύρος εφαρμογών όπως επεξεργασία σήματος (signal processing), μουσική δομή (musical structure), ανάκτηση μουσικής πληροφορίας και αναγνώριση (music information retrieval and recognition), διαδραστική μουσική σύνθεση (interactive music composition and interpretation), ταξινόμηση μουσικού είδους και αναγνώριση ύφους (Musical genre classification- Style Recognition), αναγνώριση

ομιλίας (Speech recognition). Σαν ένα παράδειγμα μηχανικής μάθησης πάνω στον μουσικό τομέα θα μπορούσαμε να πάρουμε αυτό της ταξινόμησης οργάνων μέσα από ηχητικά δείγματα. Θα μπορούσαμε να πούμε ότι ένα πρόγραμμα υπολογιστή το οποίο προσπαθεί να λύσει αυτό το πρόβλημα , βελτιώνει την απόδοση του, μέσα από την ικανότητά του να ταξινομήσει σωστά άγνωστα δείγματα ήχου, όπως π.χ. μπάσο τύμπανο, πιάνο, βιολί, και μέσα από την εμπειρία που κερδίζει εκπαιδευοντας επισημασμένα δείγματα ήχου. [10]

Στον τομέα της ταξινόμησης μουσικού είδους και αναγνώρισης ύφους, σκοπός είναι η αναγνώριση του είδους και του ύφους των μουσικών κομματιών ακόμα και όταν ένας μουσικός αυτοσχεδιάζει παίζοντας εναλλασσόμενα και διαφορετικά είδη μουσικής π.χ. blues, jazz, rock σε θλιμμένο ύφος, λυρικό, ξέφρενο κλπ. Η αυτόματη αναγνώριση ομιλίας (Automatic Speech Recognition ASR), έχει ως σκοπό την μετατροπή ακουστικών πληροφοριών σε μια αλληλουχία δεδομένων ομιλίας μέσα στην θεμελιώδη γλωσσική δομή του, συνήθως με τη μορφή λεκτικών χορδών, κάτι το οποίο είναι ένα ουσιαστικό πρόβλημα μηχανικής μάθησης. Πολλές διαφορετικές τεχνικές μπορούν να χρησιμοποιηθούν για την εκπαίδευση ενός συστήματος αναγνώρισης ομιλίας, ωστόσο το επίπεδο πολυπλοκότητας διαφέρει ανάλογα με τη φύση της εκπαίδευσης. Τα συστήματα αναγνώρισης ομιλίας χρησιμοποιούν την μηχανική μάθηση για να εξαγουν λέξεις και κείμενο μέσα από μια συνεχόμενη ηχητική ροή ομιλίας, καθώς επίσης και για τη σύνθεση ομιλίας. Τα τελευταία ονομάζονται παραγωγικά συστήματα αναγνώρισης ομιλίας. Τα πρώτα συστήματα εξαρτιόντουσαν από τον ομιλητή αλλά στη πορεία μέσα από αναβαθμίσεις στην ανάλυση του σήματος οδηγηθήκαμε σε εξαιρετικά εύρωστα συστήματα ανεξάρτητα του ομιλητή.

Στον τομέα της μουσικής σύνθεσης μπορούμε να βρούμε συστήματα τα οποία παράγουν συγχορδίες για blues ,jazz αυτοσχεδιασμούς, ακόμα και κλασικά κομμάτια. Αυτά τα συστήματα συχνά λειτουργούν εφαρμόζοντας ένα σετ από κανόνες οι οποίοι παρέχονται στο σύστημα αποκλειστικά για να καθορίσει την πιο κατάλληλη ακολουθία τιμών εξόδου. Μπορούμε ακόμα να τα δούμε να χρησιμοποιούν αναγνώριση μοτίβου (pattern recognition) και τεχνικές παράγωγης όπως τα Markov models. Τέλος στον τομέα της ανάκτησης μουσικής πληροφορίας χρησιμοποιούνται τεχνικές και αλγόριθμοι της μηχανικής μάθησης για εργασίες όπως κατηγοριοποίηση της μουσικής ανά είδος, συναίσθημα και ύφος, αναγνώριση του τραγουδιού που ακούμε και προτάσεις μουσικής βασισμένες στα χαρακτηριστικά των τραγουδιών που ακούμε. Για τον τομέα της ανάκτησης μουσικής πληροφορίας θα γίνει εκτενέστερη αναφορά στις επόμενες σελίδες.

1.5. Open source εφαρμογές

Στο διαδίκτυο κυκλοφορεί μια πληθώρα από open source εφαρμογές σχετικά με τη μηχανική μάθηση με το πιο γνωστό και διαδεδομένο να είναι το weka. Το weka (Waikato Environment for Knowledge Analysis) είναι μια πολύ διαδεδομένη συλλογή λογισμικών γραμμένα σε Java, και τα ανέπτυξε το πανεπιστήμιο του Waikato στη Νέα Ζηλανδία. Πρόκειται για ένα ελεuthερο πρόγραμμα το οποίο διατίθεται μέσα από το GNU General Public License. Τα εργαλεία του Weka περιλαμβάνουν μια συλλογή από οπτικοποιημένα εργαλεία και αλγόριθμους για ανάλυση δεδομένων και μοντέλα πρόβλεψης. Το Weka υποστηρίζει διάφορες data

mining εργασίες όπως, προεπεξεργασία δεδομένων, ομαδοποίηση, ταξινόμηση, οπτικοποίηση, επιλογή χαρακτηριστικών και μοντελοποίηση ακολουθιών.

Στη πορεία ακολουθούν και άλλα προγράμματα όπως :

PredictionIO : είναι ένας ελεύθερος server για προγραμματιστές για τη δημιουργία χαρακτηριστικών πρόβλεψης, όπως εξατομίκευση, τη σύσταση και ανακάλυψη περιεχομένου.(<https://prediction.io/>)

Mlpy: Το mlpy είναι μια υπομονάδα της rython για μηχανική μάθηση. Δίνει τη δυνατότητα μόνο με λίγες γραμμές κώδικα να φτιαχτούν περιβάλλοντα εργασίας για ταξινόμηση, παλινδρόμηση, ομαδοποίηση και επιλογή χαρακτηριστικών.(<http://mlpy.sourceforge.net/>)

Marsyas: (Music Analysis, Retrieval and Synthesis for Audio Signals) είναι ένα πλαίσιο εργασίας για την ανάπτυξη συστημάτων και επεξεργασίας του ήχου. Παρέχει μια γενική αρχιτεκτονική για τη σύνδεση του ήχου, αρχεία ήχου, επεξεργασίας σήματος και της μηχανικής μάθησης.(marsyas.info/)

BoosTexter: είναι ένα πρόγραμμα γενικής χρήσης βασισμένο στο boosting με σκοπό τη δημιουργία ενός ταξινομητή μέσα από δεδομένα κειμένου.(<https://www.cs.princeton.edu/~schapire/boostexter.html>)

Orange: ένα ελεύθερης χρήσης πρόγραμμα για απεικόνιση και ανάλυση δεδομένων μέσα από data mining και κώδικα python.(<http://orange.biolab.si/>)

Τέλος το, **OpenCV** (Open Source Computer Vision) περιέχει ελεύθερα εργαλεία μηχανικής μάθησης για χρήση κατά τη διάρκεια μιας μουσικής εκτέλεσης.(<http://opencv.org/>)

2. Ανάκτηση μουσικής πληροφορίας (Music Information Retrieval M.I.R.)

<<θέλω να βρω ένα τραγούδι που να μοιάζει με αυτό... >>

Θα μπορούσαμε να το πούμε ως ένα κλασικό παράδειγμα χρήσης της MIR τεχνολογίας, το οποίο πολύ συχνά χρησιμοποιείται για να παρουσιάσει τις δυνατότητες του MIR στον κόσμο έξω από την επιστημονική κοινότητα. Είναι πολύ συνηθισμένο σενάριο ένας χρήστης ο οποίος έχει στο μυαλό του κάποια στοιχεία ενός μουσικού κομματιού (συνήθως μελωδία και ρυθμό) να προσπαθεί να ανακαλύψει το ίδιο το τραγούδι ,άλλα παρόμοια ή του ίδιου καλλιτέχνη. Ο τομέας της ανάκτησης μουσικής πληροφορίας λοιπόν ασχολείται με την επιστήμη της ανάκτησης πληροφοριών μέσα από ένα τραγούδι. Πρόκειται για έναν πολύ ευρύ τομέα, ο οποίος καλύπτει ένα επίσης ευρύ φάσμα πιθανών χρήσεων. Ασχολείται κυρίως με την εξαγωγή πληροφοριών μέσα από ένα μουσικό περιεχόμενο, χωρίς ωστόσο να βασίζεται στην παροχή μεταδεδομένων από τον άνθρωπο. Το πεδίο της ανάκτησης μουσικής πληροφορίας περιλαμβάνει πολλούς διαφορετικούς τομείς, όπως : [12]

- Αναγνώριση καλλιτέχνη
- Ακουστικό αποτύπωμα
- Audio thumbnailing
- Αναγνώριση συγχορδιών
- Εντοπισμός επανεκτέλεσης τραγουδιού
- Εντοπισμός γυναικείων/ ανδρικών φωνητικών
- Κατάταξη ανά είδος
- Αναγνώριση οργάνου και ηχοχρώματος
- Κατάταξη κατά ύφος
- Μουσική σύσταση
- Ρυθμικές ομοιότητες
- Εκτίμηση ομοιότητας μεταξύ τραγουδιών
- Εκτίμηση Tempo
- Εγγραφή

Ανάλογα με τον τύπου της εισόδου που δέχεται ένα MIR σύστημα μπορεί να ταξινομηθεί σε : [12]

- Ερώτημα μέσω βουητού (query by Humming QBH).
- Ερώτημα μέσω παραδείγματος (query by Example QBE).
- Ερώτημα μέσω χτυπήματος (query by Tapping QBT).
- Ερώτημα μέσω Beat-boxing (query by Beat-boxing QBB).

2.1. Διεργασίες

Οι ενέργειες της ανάκτησης μουσικής πληροφορίας μπορούν να χωριστούν σε δυο κατηγορίες όπως δείχνει και ο πίνακας 1, συγκεκριμένα αντικειμενικά προβλήματα και πολιτισμικά προβλήματα. Τα αντικειμενικά προβλήματα σχετίζονται κυρίως με τις εγγενείς ιδιότητες της μουσικής, όπως ενορχήστρωση, μελωδία, αρμονία και ρυθμό, και μπορούμε χωρίς αμφιβολία να δηλώσουμε τότε ένας αλγόριθμος έχει επιτυχία και τότε όχι, ενώ για τα πολιτισμικά προβλήματα, το πολιτιστικό περιβάλλον παίζει μεγάλο ρόλο, και πρέπει να έχουμε πληροφορίες σχετικά με το υπόβαθρο να καθοριστεί αν ο αλγόριθμος επιστρέφει το σωστό αποτέλεσμα.

Αντικειμενικό	Πολιτιστικό
Αναγνώριση καλλιτέχνη	Κατάταξη ανά είδος
Αναγνώριση συγχορδιών	Κατάταξη ανά διάθεση
Αναγνώριση διασκευών	Μουσική σύσταση
Αναγνώριση γυναικείων/ ανδρικών φωνητικών	Ρυθμική ομοιότητα
Fingerprinting	Ομοιότητα τραγουδιών
Αναγνώριση μουσικού οργάνου / ηχοχρώμα	
Εκτίμηση Tempo	
Εγγραφή	

Πίνακας 1: Ενέργειες της ανάκτησης μουσικής πληροφορίας ανά κατηγορία[12]

Στα διάφορα ερευνητικά πεδία που καλύπτει περιλαμβάνονται τα : [14] [12] [16]

Συστήματα αναγνώρισης ομιλίας τα οποία αυτόματα εξάγουν τους στίχους από ένα τραγούδι. Τα προβλήματα που αντιμετωπίζει αυτή η μέθοδος είναι ο μεγάλος βαθμός θορύβου βάθους, οι παραλλαγές στην ομιλία και το εύρος των γλωσσών που χρησιμοποιούνται στη μουσική.

Αυτόματη εγγραφή η οποία παράγει μουσικές παρτιτούρες από κάθε είδος μουσικής οι οποίες είναι κατάλληλες να παιχτούν από έναν σολίστα ή ακόμα και από ολόκληρη ορχήστρα.

Ερώτημα μέσω βουητού (QBH).Είναι η αναζήτηση σε μια μουσική βάση δεδομένων μέσα μουρμουρητού ή σφυρίγματος στον υπολογιστή. Ο υπολογιστής στη συνέχεια αναγνωρίζει τις θεμελιώδεις συχνότητες στο μουρμουρητό, και τις αναζητεί στο ευρετήριο.

Ακουστικό αποτύπωμα . Εντοπίζει αυτόματα τα αναγνωριστικά χαρακτηριστικά ενός μουσικού κομματιού. Κάποια από αυτά τα συστήματα επιτρέπουν την αναζήτηση μέσω της τεχνικής του κοντινότερου γείτονα , άλλα επικεντρώνονται αποκλειστικά στον εντοπισμό μιας συγκεκριμένης ηχογράφησης ενός τραγουδιού. Μπορούν να χρησιμοποιηθούν σε περιπτώσεις όπως αναζήτηση ενός τραγουδιού το οποίο παίζεται από ένα ραδιόφωνο.

Εξαγωγή χαρακτηριστικών. Χρησιμοποιείται για την αναγνώριση των χαρακτηριστικών ενός μουσικού κομματιού όπως τα bps. Η Echo Nest έχει δημιουργήσει μια ελεύθερη εφαρμογή η οποία εξάγει τυχαία χαρακτηριστικά από τον ήχο και έχει ένα παράδειγμα που επιτρέπει στον χρήστη να εξάγει διάφορα στοιχεία από τον ήχο και του επιτρέπει να βάλει μαζί ένα ομοιογενές μείγμα από διάφορα τραγούδια.

Κατηγοριοποίηση ανά είδος. Η σημερινή μουσική βιομηχανία απαρτίζεται από έναν μεγάλο αριθμό μουσικών ειδών. Έτσι λοιπόν έχουμε ένα σύστημα που κατηγοριοποιεί την μουσική ανά είδος, και που μπορεί να συγκρίνει τις ομοιότητες μεταξύ διαφορετικών μουσικών ειδών.

Αναγνώριση καλλιτέχνη. Οι αλγόριθμοι αυτής της κατηγορίας συνήθως προσπαθούν να αναγνωρίσουν κομμάτια με παρόμοια ενορχήστρωση, ρυθμό ή και μελωδία. Καθότι είναι σύνηθες μερικές φορές τραγούδια του ίδιου καλλιτέχνη να περιέχουν τα ίδια χαρακτηριστικά. Όπως με την κατηγοριοποίηση ανά είδος, έτσι και η αναγνώριση καλλιτέχνη έχει πολλές φορές ως στόχο της να αποδείξει ότι οι αλγόριθμοι συμπεριφέρονται <<λογικά>>.

Audio tagging/ περιγραφή . Το Audio tagging επιχειρεί να συνδέσει της λέξεις (tags) με τα τραγούδια. Πρόκειται για έναν σχετικά καινούριο τομέα στον χώρο ο όποιος όμως εξελίσσεται συνεχώς.

Μουσική ομοιότητα. Ουσιαστικά η αξιολόγηση του πόσο όμοια ή διαφορετικά είναι δυο μουσικά κομμάτια, και θεωρείται συχνά ως το βασικό πρόβλημα που οι ερευνητές προσπαθούν να λύσουν κατά την αξιολόγηση των αλγορίθμων τεχνικές ταξινόμησης για είδος και καλλιτέχνη.

Τα περισσότερα συστήματα είτε αναλύουν ένα τραγούδι και ορίζουν μια συγκεκριμένη ετικέτα, όπως για παράδειγμα αναγνώριση είδους ή καλλιτέχνη, είτε επιστρέφουν ένα μέτρο απόστασης- ομοιότητας μεταξύ των δυο κομματιών, κάτι που γίνεται σε πολλά συστήματα ταυτοποίησης διασκευών. Όλος ο τομέας είναι οργανωμένος γύρω από περιπτώσεις που περιγράφουν μια αναζήτηση, στην συνέχεια την αντιστοίχιση, και τέλος την έξοδο που παίρνουμε ως αποτέλεσμα. Σαν αναζητήσεις και εξόδους μπορούμε να έχουμε πληροφορίες κειμένου (metadata), μουσικά αποσπάσματα, ηχογραφήσεις, ή μουσικά χαρακτηριστικά. Η αντιστοίχιση μπορεί να είναι ακριβής όπως κατά την ανάκτηση μουσικής με συγκεκριμένο περιεχόμενο, ή κατά προσέγγιση όπως κατά την ανάκτηση κοντινότερου γείτονα.

2.2. Σχεδιασμός και αλγόριθμοι

Η είσοδος που δέχεται το σύστημα έχει τη μορφή ενός σήματος αναζήτησης και δέχεται τέτοια επεξεργασία έτσι ώστε να εξαχθούν κάποια βασικά χαρακτηριστικά του ήχου, ίδια στο σύνολο με αυτά των ήχων στην βάση δεδομένων. Στην συνέχεια το τραγούδι της αναζήτησης συγκρίνεται με τα τραγούδια στη βάση δεδομένων μέσω αυτών των χαρακτηριστικών. Ανάλογα με το μέτρο απόστασης (distance measure) που χρησιμοποιείται, ανακτώνται τα τραγούδια που έχουν κάποια ομοιότητα. Παρακάτω παρουσιάζονται τα διαφορετικά στάδια ενός τέτοιου συστήματος και περιλαμβάνουν διαφορετικές τεχνικές επεξεργασίας σήματος και αναγνώρισης μοτίβου :[12][13] [20]

Ανάκτηση σήματος: συνήθως μέσω κάποιας ακουστικής αναζήτησης όπως σφύριγμα, μурμυρητό, τραγούδι, ρυθμικό χτύπημα ή ακόμα και ένα μουσικό κομμάτι σαν είσοδος .

Thumbnailing σήματος: Thumbnailing είναι η διαδικασία αναζήτησης του πιο αντιπροσωπευτικού σημείου ενός τραγουδιού όπως ρεφρέν ή γέφυρα. Συνήθως το σύνολο των χαρακτηριστικών εξάγεται από τα τραγούδια της βάσης δεδομένων χρησιμοποιώντας το thumbnail των τραγουδιών αντί για ολόκληρα τα κομμάτια.

Εξαγωγή χαρακτηριστικών: σε αυτό το στάδιο το σήμα εισόδου χωρίζεται σε μικρά πλαίσια (frames) και υποθέτοντας ότι το σήμα είναι στάσιμο μέσα σε αυτά, εξάγονται διαφορετικά χαρακτηριστικά από κάθε πλαίσιο. Συνήθως το μήκος των πλαισίων αυτών είναι της τάξεως των 20ms-30ms και τα χαρακτηριστικά που εξάγονται πιο συχνά είναι αυτά του χρόνου, της συχνότητας καθώς και του τονικού ύψους.

Σύγκριση και ανάκτηση χαρακτηριστικών : τα χαρακτηριστικά που εξάγονται από την αναζήτηση και τα χαρακτηριστικά στη βάση δεδομένων συγκρίνονται για οποιαδήποτε ομοιότητα μπορεί να έχουν μέσω ενός μέτρου απόστασης. Ομοιότητες μπορεί να βρεθούν σε χαρακτηριστικά όπως ρυθμός, μελώδια, διάθεση, είδος, καλλιτέχνης ή ακόμα και σε κάποιον συνδυασμό όλων αυτών. Στη συνέχεια ένας αριθμός όμοιων τραγουδιών ανακτώνται και παρουσιάζονται ως κοντινότερα στο τραγούδι της αναζήτησης. Οι πιο συνήθεις τεχνικές ανάκτησης είναι αυτές του κοντινότερου γείτονα και της ιεραρχικής ταξινόμησης.

Οι αλγόριθμοι της ανάκτησης μουσικής πληροφορίας έχουν την ικανότητα να αναγνωρίζουν και να εξάγουν αυτές τις πληροφορίες, επιτρέποντας στα συστήματα

να πραγματοποιούν εκτενή ταξινόμηση, αναζήτηση, μουσικές προτάσεις, παραγωγή μεταδεδομένων, περιγραφή ακόμα και να βοηθούν σε real-time εκτελέσεις. Όπως έχει αναφερθεί και παραπάνω ο τομέας της ανάκτησης μουσικής πληροφορίας είναι κομμάτι της μηχανικής μάθησης και έτσι δε θα μπορούσε παρά οι αλγόριθμοι που χρησιμοποιούνται να προέρχονται από εκεί. Αλγόριθμοι όπως :

K-Nearest Neighbor (KNN): ένας μη παραμετρικός ταξινομητής ο οποίος βασίζεται στην ιδέα ότι ένας μικρός αριθμός «γειτόνων» επηρεάζουν μια απόφαση σχετικά με ένα θέμα.

Support Vector Machines (SVM): Τα SVMs βασίζονται σε δύο ιδιότητες: μεγιστοποίηση του περιθωρίου και γραμμική μετατροπή του χώρου με τη χρήση πυρήνων. Έχουν χρησιμοποιηθεί ιδιαίτερος στον χώρο της αναγνώρισης είδους.

Hidden Markov Model (HMM): Τα Hidden Markov Models που συνήθως χρησιμοποιούνται σε περιπτώσεις ταξινόμησης, έχουν χρησιμοποιηθεί ευρέως στην αναγνώριση λόγου εξαιτίας της ικανότητας τους να χειρίζονται καλά τα δεδομένα χρόνου.

k-means clustering: Είναι μια μέθοδος κβαντισμού διανυσμάτων η οποία αρχικά χρησιμοποιήθηκε για ανάλυση σήματος. Στοχεύει στην ομαδοποίηση n παρατηρήσεων σε k συμπλέγματα στα οποία η κάθε παρατήρηση ανήκει στο σύμπλεγμα με το κοντινότερο mean, παίζοντας τον ρόλο του πρωτοτύπου για το σύμπλεγμα. Έχει χρησιμοποιηθεί από τους ερευνητές σε εφαρμογές σύστασης μουσικής.

2.3. Βάσεις δεδομένων ανάκτησης μουσικής πληροφορίας

Εδώ, το σύνολο των δεδομένων ορίζεται πιο ελεύθερα ως μια ομάδα αρχείων ήχου που συνοδεύονται από έναν αριθμό πληροφοριών οι οποίες μπορούν να φανούν χρήσιμες κατά τη διάρκεια της εκπαίδευσης και της αξιολόγησης αλγορίθμων που προορίζονται για την αναπαραγωγή αυτών των πληροφοριών. Το μέγεθος της βάσης δεδομένων παίζει μεγάλο ρόλο στην ανάκτηση καθώς και στην ποιότητα αυτής. Σε γενικές γραμμές, όσο μεγαλύτερο είναι το μέγεθος της βάσης δεδομένων, συνήθως τόσο καλύτερη είναι η ποιότητα της ανάκτησης. Μεγάλο ρόλο στην ποιότητα της ανάκτησης παίζει ακόμα και το είδος του περιεχομένου της βάσης δεδομένων. Οι βάσεις δεδομένων μπορούν να χωριστούν ανάλογα το περιεχόμενό τους, το οποίο ποικίλει από ρυθμό, είδος μουσικής, αναζήτηση διάθεσης, όργανα και ηχητικά δείγματα μέχρι αναζήτηση μέσω βουητού και beat.

Παρακάτω συναντάμε Κάποιες από τις πολλές βάσεις δεδομένων που υπάρχουν στο διαδίκτυο:

Echonest : Η βάση δεδομένων του Echonest περιέχει περίπου 30 εκατομμύρια τραγούδια τα οποία συγκεντρώνονται χρησιμοποιώντας web crawling, εξόρυξη δεδομένων και τεχνικές επεξεργασίας ψηφιακών σημάτων. Τα δεδομένα τροφοδοτούν τομείς όπως μουσική σύσταση, ακουστική ανάλυση, και αναγνώριση μουσικών κομματιών. (<http://the.echonest.com/>)

ENST Drum Dataset: Πρόκειται για μια μεγάλη ερευνητική βάση δεδομένων για αυτόματη εγγραφή και επεξεργασία τυμπάνων. (<http://perso.telecom-paristech.fr/~grichard/ENST-drums/>)

MTG- QHB: Περιέχει 118 αρχεία τραγουδισμένων μελωδιών για αναζήτηση μέσω βουητού. (<http://mtg.upf.edu/download/datasets/mtg-qbh>)

MAPS: Είναι μια μεγάλη βάση δεδομένων για ήχους πιάνο, νότες, συγχορδίες και αποσπάσματα τα οποία έχουν ηχογραφηθεί από κάποιον πιανίστα. (<http://www.tsi.telecom-paristech.fr/aa0/en/category/database/>)

RWC Music Database: Είναι μιας μεγάλης κλίμακας βάση δεδομένων που περιέχει 6 συλλογές: the Popular Music Database (100 τραγούδια), Royalty-Free Music Database (15 τραγούδια), Classical Music Database (50 τραγούδια), Jazz Music Database (50 τραγούδια), Music Genre Database (100 τραγούδια), και Musical Instrument Sound Database (50 όργανα). (<https://staff.aist.go.jp/m.goto/RWC-MDB/>)

MusicBrainz: Μια βάση μεταδεδομένων που προσπαθεί να δημιουργήσει μια ιστοσελίδα μουσικών πληροφοριών η οποία ως χαρακτηριστικό της θα έχει τη νοημοσύνη. Προσφέρει ετικέτες σχετικά με καλλιτέχνες ή άλμπουμ. (<http://musicbrainz.org/>)

DJ bpm studio: Μια βάση δεδομένων μεγάλων προδιαγραφών που περιέχει τιμές που αφορούν το τέμπο από περίπου 80,000 γνωστά τραγούδια. (<http://djbpmstudio.com/>)

Gracenote: Πρόκειται για μια εταιρία που περιέχει και διαχειρίζεται μια διαδικτυακή βάση δεδομένων που περιέχει όλες τις πληροφορίες σχετικά με τα περιεχόμενα ψηφιακών δίσκων και βινυλίων. Δουλεία της είναι να προμηθεύει με λογισμικά και μεταδεδομένα άλλες εταιρίες οι οποίες έχουν σαν αντικείμενο τους την διαχείριση και την αναζήτηση ψηφιακών μουσικών δεδομένων. (<http://www.gracenote.com/>)

2.4. Συστήματα

Καθώς ο τομέας του MIR αναπτύσσεται ραγδαία εμφανίζεται και η ανάγκη από τους μελετητές να δημιουργήσουν συστήματα τα οποία θα έχουν ως σκοπό τους να λύνουν τις οποιεσδήποτε επιπλοκές που μπορεί να δημιουργηθούν. Έτσι έχουμε μια πληθώρα συστημάτων με σκοπό τους να λύσουν αυτές τις επιπλοκές και να κάνουν τον τομέα της μουσικής ανάκτησης προσβάσιμο ακόμα και στους απλούς χρήστες. Παρακάτω παρουσιάζονται κάποια παραδείγματα τέτοιων συστημάτων και τεχνολογιών.

Musart: Πρόκειται για ένα σύστημα φωνητικής αναζήτησης, και «πρόκειται για ένα καλό παράδειγμα του πως ένα θεματικό υλικό μπορεί να εξαχθεί και να γίνει απόλυτη σύγκριση του με χορδές αναζήτησης» (Birmingham et al., 2003). Το Musart το κάνει αυτόματα μέσω ενός ειδικά σχεδιασμένου αλγορίθμου και μπορεί να μεταφράσει κάθε μορφοποίηση σε κάποια άλλη δίνοντας τη δυνατότητα στους χρήστες να δουν τις πληροφορίες που χρειάζονται με διάφορους τρόπους.

Foote: Επιχειρεί να βρει και να ανακτήσει ρυθμικές ομοιότητες ανάμεσα σε τραγούδια τα οποία τα αναλύει και τα συγκρίνει φασματικά. Χρησιμοποιεί μια

συνάρτηση αναζήτησης βασισμένη σε 2D matrix ώστε να αναδείξει της ομοιότητες των τραγουδιών που συγκρίνει.

C-Brahms: Χρησιμοποιεί περίπου 9 διαφορετικούς αλγορίθμους που ονομάζονται P1, P2, P3, MonoPoly, IntervalMatching, ShiftOrAnd, Poly-Check, Splitting, και LCTS. Αυτό που προσφέρει είναι διαφορετικούς συνδυασμούς μονοφωνίας, πολυφωνίας, σταθερού ρυθμού, μερικής ή ακριβής αντιστοίχισης .

Notify: Αυτό που κάνει το Notify είναι να ταιριάζει μονοφωνικές αναζητήσεις από σφύριγμα με πολυφωνικά σετ από νότες, ενώ ακόμα και αν υπάρχουν διαφορές ή διακυμάνσεις στον ρυθμό ένας ανιχνευτής ρυθμού επιτρέπει το ταίριασμα μεταξύ του σφυρίγματος και των νοτών. Έχει ακόμα και τη δυνατότητα τα αποτελέσματα της αναζήτησης να τα αποτυπώσει σε μορφή piano roll.

Themefinder: Είναι μια web-based εφαρμογή η οποία επιτρέπει στον χρήστη να αναζητήσει μουσικά θέματα δίνοντας του τη δυνατότητα να χρησιμοποιήσει μια ποικιλία από λέξεις κλειδιά περιλαμβανόμενου του τονικού ύψους, κλίμακα, ημέρα δημιουργίας και πολλά άλλα. Η αναζήτηση στις βάσεις δεδομένων που περιέχουν αυτά τα μουσικά θέματα γίνεται μέσα από αλγορίθμους αντιστοίχισης χορδών .

Last. Fm: Χρησιμοποιεί ένα σύστημα ονόματι "Audioscrobbler", το οποίο έχει τη δυνατότητα να δημιουργήσει ένα λεπτομερές προφίλ των χρηστών με τις μουσικές τους προτιμήσεις, χρησιμοποιώντας τις πληροφορίες από τα τραγούδια που ακούει ο χρήστης είτε μέσα από το διαδίκτυο είτε μέσα από άλλες συσκευές. Αυτές οι πληροφορίες μεταφέρονται στην πορεία στη βάση δεδομένων του Last.FM είτε μέσα από τον ίδιο τον player είτε μέσα από plugins εγκατεστημένα στον player. Τα δεδομένα τέλος εμφανίζονται στη σελίδα του χρήστη και μπορούν ακόμα να χρησιμοποιηθούν ώστε να δημιουργήσουν σελίδες αναφοράς για μεμονωμένους καλλιτέχνες.

2.5. ISMIR και MIREX

Η διεθνής κοινότητα σχετικά με την ανάκτηση μουσικής πληροφορίας (*International Society for Music Information Retrieval*) είναι ένα διεθνές forum που σχετίζεται με την έρευνα και την οργάνωση των μουσικών δεδομένων. Σκοπός του είναι να παρέχει έναν χώρο για ανταλλαγή ιδεών και νέων μέσα από τις παρουσιάσεις των έργων που έχουν γίνει σε θεωρητικό και πρακτικό επίπεδο, άλλα και να δώσει σε βάθος πληροφορίες σε συγκεκριμένους τομείς ενώ ταυτόχρονα παρουσιάζει νέα προϊόντα. [18]

Το Music Information Retrieval Evaluation eXchange (MIREX) είναι μια ετήσια αξιολόγηση των αλγορίθμων της ανάκτησης μουσικής πληροφορίας που λαμβάνει χώρα σε συνδυασμό με το ISMIR. Ο σκοπός του η σύγκριση και αξιολόγηση των πιο γνωστών αλγορίθμων και συστημάτων που αφορούν την ανάκτηση μουσικής πληροφορίας.[18]

ΚΕΦΑΛΑΙΟ 2

1. Ακουστικό αποτύπωμα (Audio Fingerprinting AF)

1.1. Περιγραφή

Όπως ο κάθε άνθρωπος έχει ένα διαφορετικό και μοναδικό δακτυλικό αποτύπωμα από το οποίο μπορεί να ταυτοποιηθεί, έτσι ακριβώς συμβαίνει και με το ακουστικό αποτύπωμα και την αναγνώριση των μουσικών κομματιών. Τα ακουστικά αποτυπώματα σε συνδυασμό με τις μουσικές βάσεις δεδομένων, μπορούν να χρησιμοποιηθούν ώστε να αντλήσουν πληροφορίες για άγνωστα ηχητικά αποσπάσματα όπως το όνομα του τραγουδιού, του δίσκου ή του καλλιτέχνη. Το soundhound και το Shazam είναι δυο πολύ καλά παραδείγματα τέτοιων εφαρμογών και υπηρεσιών, όπου μέσα από μόλις λίγα δευτερόλεπτα ήχου που έχουν παρθεί από ένα κινητό τηλέφωνο, επιστρέφουν στον χρήστη όλες τις πληροφορίες που χρειάζεται για το συγκεκριμένο τραγούδι. Τα τελευταία χρόνια, το ακουστικό αποτύπωμα χρησιμοποιείται ακόμα και για θέματα πνευματικής ιδιοκτησίας της μουσικής, όπως για παράδειγμα η ανίχνευση τραγουδιών στο διαδίκτυο που η διανομή τους παραβιάζει τα πνευματικά δικαιώματα του δημιουργού τους. Τι είναι όμως ακριβώς το ακουστικό αποτύπωμα και πως βοηθά στην εξέλιξη της σημερινής μουσικής βιομηχανίας;

1.2. Ορισμός του ακουστικού αποτυπώματος

Σύμφωνα με τον Pedro Cano (*Audio Fingerprinting: Concepts And Applications*) [30] ένα ακουστικό αποτύπωμα είναι μια συμπαγής υπογραφή βασισμένη στο περιεχόμενο η όποια περιγράφει μια ηχητική ηχογράφηση. Τα συστήματα που λειτουργούν με το ακουστικό αποτύπωμα εξάγουν ακουστικά χαρακτηριστικά από ένα ηχητικό απόσπασμα και τα αποθηκεύουν σε μια βάση δεδομένων. Όταν αυτά τα συστήματα τροφοδοτηθούν με ένα άγνωστο μουσικό απόσπασμα τότε αναλύουν και συγκρίνουν τα χαρακτηριστικά αυτού του αποσπάσματος με τα χαρακτηριστικά που είναι αποθηκευμένα στην βάση δεδομένων. Μέσα από την χρήση αποτυπωμάτων και των κατάλληλων αλγορίθμων είναι δυνατό να αναγνωριστούν ακόμα και ηχογραφήσεις οι οποίες για οποιονδήποτε λόγο έχουν δεχτεί παραμόρφωση. Για να επιτευχτεί αυτό αρχικά τα αρχεία ήχου πρέπει να διαβαστούν έχοντας πρώτα μετατραπεί σε μια κυματομορφή βασισμένη στον χρόνο, χωρίζοντας την στην πορεία σε μια σειρά από κομμάτια που περιέχουν τα δεδομένα για την συχνότητα και αναζητούνται καθοριστικά χαρακτηριστικά που θα δημιουργήσουν το αποτύπωμα βασισμένο στη συχνότητα τον χρόνο και στο πως σχετίζονται μεταξύ τους. Υπάρχουν ήδη αρκετοί γνωστοί και αξιόπιστοι αλγόριθμοι που σχετίζονται με το ακουστικό αποτύπωμα και έχουν δημιουργηθεί από εταιρίες για διάφορους σκοπούς. Οι πιο σύγχρονοι από αυτούς έχουν δημιουργηθεί για ταχύτητα και έχουν ως σκοπό να χρησιμοποιούν χαρακτηριστικά που βρίσκουν μέσα από φασματική ανάλυση του ήχου ώστε να παράγουν ακουστικά αποτυπώματα.

Υπάρχουν δυο βασικά είδη αποτυπωμάτων : MusicID-File και MusicID-Stream. Το MusicID-File αναγνωρίζει αρχεία ήχου όπως για παράδειγμα αυτά που δημιουργούνται κατά την αντιγραφή ενός CD και χρησιμοποιεί το πρώτο τμήμα του αρχείου για την αναγνώριση. Το MusicID-Stream μπορεί να αναγνωρίσει ένα

απόσπασμα ήχου το οποίο μπορεί να παρθεί από οποιοδήποτε σημείο του track και να το κάνει κατάλληλο για αναγνώριση ήχου συνεχούς ροής (stream) ή ήχο περιβάλλοντος. Τα MusicID-File αποτυπώματα χρησιμοποιούν τα πρώτα 15 δευτερόλεπτα του ήχου από μια συνεχόμενη ροή, ενώ αγνοούν την οποιαδήποτε ησυχία στην αρχή του track. Το MusicID-Stream αποτύπωμα χρησιμοποιεί περίπου 6.5 δευτερόλεπτα από την ροή του ήχου για τις περισσότερες περιπτώσεις, ωστόσο ανάλογα την περίπτωση μπορεί να χρειαστεί και περισσότερος χρόνος. Σε γενικές γραμμές η αναγνώριση δεν χρειάζεται να ξεκινήσει από τη αρχή ενός ηχητικού αποσπάσματος αλλά και από άλλα σημεία του. Θα μπορούσαμε να θεωρήσουμε κάθε σημείο σαν σημείο εκκίνησης της αναγνώρισης αλλά αυτό χρειάζεται πολύ χρόνο για να συμβεί. Έτσι για να αντιμετωπιστεί αυτό το πρόβλημα έχουν δημιουργηθεί διάφορες σύγχρονες και ακριβείς μέθοδοι ανάκτησης. Ένα «ερώτημα» (ερώτημα) είναι ένα απόσπασμα από ένα τραγούδι, το οποίο όμως μπορεί να φέρει αλλοιώσεις από την ανάμιξη με τον περιβαλλοντικό θόρυβο, ή μπορεί να έχει υποστεί μετατροπές από ένα low-pass φίλτρο. Οι μέθοδοι ανάκτησης θα πρέπει να είναι σε θέση να χειρίζονται ερωτήματα που μπορεί να είναι παρόμοια αλλά όχι ολόιδια με ένα τραγούδι στην βάση δεδομένων. Η μέθοδος του Locality-Sensitive Hashing (LSH) είναι ένα τέτοιο παράδειγμα και ασχολείται με την επίλυση μεγάλης κλίμακας ανακτήσεων ομοιότητας.

Ένα ιδανικό σύστημα ακουστικών αποτυπωμάτων θα πρέπει να πληροί κάποιες προϋποθέσεις όπως: να μπορεί να αναγνωρίζει με ακρίβεια ένα αντικείμενο ανεξαρτήτου συμπίεσης ή παραμόρφωσης που μπορεί να έχει υποστεί. Ανάλογα την εφαρμογή, θα πρέπει να είναι σε θέση να προσδιορίσει ολόκληρους τίτλους μέσα από αποσπάσματα μόλις λίγων δευτερολέπτων, το οποίο χρειάζεται μεθόδους που να αντιμετωπίζουν ένα πρόβλημα που ονομάζεται shifting, δηλαδή την έλλειψη συγχρονισμού μεταξύ του αποτυπώματος που έχει εξαχθεί και εκείνων που βρίσκονται στη βάση δεδομένων. Θα πρέπει να είναι σε θέση να αντιμετωπίσει άλλες αιτίες υποβάθμισης, όπως το pitching, ισοστάθμιση, θόρυβος περιβάλλοντος, D/A-A/D conversion, κωδικοποιήσεις ήχου (όπως GSM ή MP3), κλπ. Θα πρέπει επίσης να είναι και υπολογιστικά αποδοτικό, κάτι που έχει να κάνει με το μέγεθος των αποτυπωμάτων, την πολυπλοκότητα των αλγορίθμων αναζήτησης και την πολυπλοκότητα της διαδικασίας εξαγωγής του αποτυπώματος.

1.3. Παράμετροι των ακουστικών αποτυπωμάτων

Οι απαιτήσεις διαφέρουν σημαντικά ανάλογα την εφαρμογή αλλά είναι χρήσιμες ώστε να μπορούμε να συγκρίνουμε και να αξιολογήσουμε τις διάφορες audio fingerprinting τεχνολογίες. Το IFPI (International Federation of the Phonographic Industry) και το RIAA (Recording Industry Association of America) έχουν προσπαθήσει να αξιολογήσουν διάφορα συστήματα αναγνώρισης περιεχομένου. Τέτοια συστήματα πρέπει να είναι εύρωστα καθώς και αξιόπιστα στους υπολογισμούς τους. Παρακάτω βλέπουμε τις παραμέτρους που πρέπει να πληρούνται για αυτά τα συστήματα:[24]

Ευρωστία: Το ακουστικό αποτύπωμα πρέπει να είναι αρκετά εύρωστο ώστε ένα ηχητικό απόσπασμα να αναγνωριστεί ακόμη και αν το σήμα έχει υποστεί σοβαρές παραμορφώσεις. Για να επιτευχθεί ικανοποιητική ευρωστία, το ακουστικό αποτύπωμα πρέπει να βασίζεται σε χαρακτηριστικά του σήματος που μένουν ανεπηρέαστα στον θόρυβο. Έτσι, ακόμη και ακουστικά αντικείμενα που έχουν

παραμορφωθεί σοβαρά, να οδηγούν σε πολύ όμοια αποτυπώματα. Το false negative χρησιμοποιείται για να εκφράσει την ευρωστία. Ένα false negative συμβαίνει όταν τα αποτυπώματα ίσων αντικειμένων είναι πολύ διαφορετικά για να οδηγήσουν σε θετική αναγνώριση. Δηλαδή όταν δεν γίνεται αναγνώριση, ενώ θα έπρεπε να είχε γίνει.

Αξιοπιστία: Πόσο αξιόπιστη είναι η αναγνώριση, ή με άλλα λόγια, πόσο συχνά ένα τραγούδι αναγνωρίζεται λάθος. Για παράδειγμα, το τραγούδι “Scorpions - Wind of change” να αναγνωριστεί ως “John Lennon -Imagine”. Το false positive χρησιμοποιείται για να εκφράσει την αξιοπιστία. Έτσι, στην προηγούμενη λάθος αναγνώριση έχουμε ένα false positive. Δηλαδή όταν γίνεται αναγνώριση, ενώ δεν θα έπρεπε να είχε γίνει.

Μέγεθος: Το μέγεθος που έχει ένα ακουστικό αποτύπωμα. Για τη γρήγορη ανάκτηση τους, τα αποτυπώματα συνήθως αποθηκεύονται στη μνήμη RAM. Επομένως, το μέγεθος των αποτυπωμάτων, που συνήθως εκφράζεται σε bits ανά δευτερόλεπτο καθορίζει σε μεγάλο βαθμό τις απαιτήσεις σε μνήμη του server της βάσης δεδομένων.

Granularity: Πόσα δευτερόλεπτα ενός ακουστικού αντικειμένου χρειάζονται για να εξαχθεί ένα ακουστικό αποτύπωμα και να αναγνωριστεί το τραγούδι. Η παράμετρος αυτή εξαρτάται από την συνάρτηση που χρησιμοποιείται για την εξαγωγή του ακουστικού αποτυπώματος και αποτελεί ένα ελάχιστο κάτω όριο στη χρονική διάρκεια του τραγουδιού η οποία απαιτείται ώστε να έχουμε ικανοποιητική ποιότητα αναγνώρισης.

Ταχύτητα Αναζήτησης: Ποιος είναι ο χρόνος που χρειάζεται για να βρεθεί ένα ακουστικό αποτύπωμα στην βάση δεδομένων και πώς αυτός ο χρόνος επηρεάζεται όσο μεγαλώνει ο αριθμός των τραγουδιών της βάσης δεδομένων.

Ασφάλεια: Το πόσο ευπαθής σε αλλοιώσεις μπορεί να είναι ο αλγόριθμος. Δηλαδή σε καταστάσεις που μπορούν να ξεγελάσουν τον αλγόριθμο αναγνώρισης του αποτυπώματος.

Σταθερότητα: Η ικανότητα να αναγνωρίζει τον ήχο ανεξάρτητα από την μορφοποίηση του. Καθώς και η ικανότητα να χρησιμοποιεί την ίδια βάση δεδομένων για διαφορετικές εργασίες.

Κλιμάκωση : όταν υπάρχει επίδοση με πολύ μεγάλες βάσεις δεδομένων ή τίτλους ή και με μεγάλο αριθμό ταυτόχρονων ταυτοποιήσεων. Κάτι που επηρεάζει την ακρίβεια και την πολυπλοκότητα του συστήματος.

Πολυπλοκότητα: Αναφέρεται στο υπολογιστικό κόστος της εξαγωγής αποτυπωμάτων, το μέγεθος του αποτυπώματος, την πολυπλοκότητα της αναζήτησης, τη πολυπλοκότητα της σύγκρισης των αποτυπωμάτων, το κόστος του να προσθέτει νέα αντικείμενα στην βάση δεδομένων κλπ.

Ευθραυστότητα: Όταν τα αποτυπώματα θα πρέπει να είναι ισχυρά απέναντι σε μετασχηματισμούς του περιεχομένου τους αλλά όχι απέναντι σε άλλες τροποποιήσεις, καθώς η τελειοποίηση κάποιας συγκεκριμένης απαίτησης συχνά συνεπάγεται απώλεια επιδόσεων σε κάποια άλλη.

Σε γενικές γραμμές ένα ακουστικό αποτύπωμα πρέπει να είναι: [27][30]

- Μια αντιληπτή σύνοψη της ηχογράφησης όπου πρέπει να διατηρεί το μέγιστο από τις ακουστικές πληροφορίες της. Αυτή η σύνοψη θα πρέπει να επιτρέπει τη διάκριση ανάμεσα από έναν μεγάλο αριθμό αποτυπωμάτων κάτι που έρχεται όμως σε αντίθεση με κάποιες άλλες απαιτήσεις όπως η πολυπλοκότητα και η ευρωστία.
- Αμετάβλητο σε παραμορφώσεις. Αυτό προκύπτει μέσα από τις απαιτήσεις της ευρωστίας. Κάποιες εφαρμογές ωστόσο χαλαρώνουν τους περιορισμούς των παραμορφώσεων του περιεχομένου ώστε να μπορούν να ανιχνεύσουν σκόπιμες παραποιήσεις.
- Συμπαγές. Μια μικρού μεγέθους παρουσίαση είναι ενδιαφέρουσα σχετικά με την πολυπλοκότητα από τη στιγμή που ένας μεγάλος αριθμός αποτυπωμάτων πρέπει να αποθηκευτεί και να συγκριθεί. Μια υπερβολικά μικρή εκπροσώπηση, ωστόσο, ενδέχεται να μην είναι επαρκής για των διαχωρισμό μεταξύ ηχογραφήσεων που επηρεάζουν την ακρίβεια, την αξιοπιστία και την ευρωστία.
- Εύκολα υπολογίσιμο. Η εξαγωγή του αποτυπώματος θα πρέπει να μην καταναλώνει πολύ χρόνο.

Όπως αναφέρθηκε και παραπάνω ένα σύστημα ακουστικών αποτυπωμάτων θα πρέπει να μπορεί να αναγνωρίζει με ακρίβεια ένα αντικείμενο, ανεξαρτήτου συμπίεσης ή παραμόρφωσης που μπορεί να έχει υποστεί και να αναγνωρίζει τα τραγούδια μέσα σε λίγο μόνο δευτερόλεπτα. Πλέον είναι προφανές ότι θα πρέπει να είναι και υπολογιστικά αποδοτικό. Κάτι που είναι κρίσιμο σε πραγματικές εφαρμογές τόσο στον υπολογισμό του αποτυπώματος του αγνώστου ήχου, όσο και κατά τη διάρκεια της αναζήτησης για ταίριασμα μέσα από τη βάση δεδομένων. Αυτό το υπολογιστικό κόστος σχετίζεται με το μέγεθος των αποτυπωμάτων, την πολυπλοκότητα του αλγορίθμου αναζήτησης και τη πολυπλοκότητα της εξαγωγής του αποτυπώματος.

1.4. Αλγόριθμοι, Δομή και σχεδιασμός

Κατά τον Haitsma et al. (2001) ένα αποτύπωμα μπορεί να αποκαλεστεί και « ισχυρός κατακερματισμός», καθώς δηλώνει ότι ένας αλγόριθμος πρέπει να παράγει κατακερματισμούς βασισμένους σε μια είσοδο που είναι δυνατή απέναντι σε τροποποιήσεις του ήχου που δεν αλλάζουν δραστικά τον ήχο. Προτείνεται ότι πρέπει να σχεδιαστεί ένας αλγόριθμος ο οποίος να είναι πολύ κοντά στο ανθρώπινο ακουστικό σύστημα: [24]

«Ένας ισχυρός ηχητικός κατακερματισμός είναι μια λειτουργία που σε κάθε βασική μονάδα χρόνου του ηχητικού περιεχόμενου συνδέει μια σύντομη ημι-μοναδική αλληλουχία από bit που είναι σε συνεχή σχέση με την ομοιότητα περιεχομένου όπως την αντιλαμβάνεται το ανθρώπινο ακουστικό σύστημα». (Haitsma et al. 2001, p. 2)

Για να μπορέσει ένα ερώτημα (query) να αντιστοιχηθεί με ένα αποτέλεσμα από την βάση αναφοράς, ένας αλγόριθμος πρέπει να παράγει πανομοιότυπους κατακερματισμούς μεταξύ της ηχογράφησης και του ερωτήματος. Πρέπει να είναι πανομοιότυποι ακόμα και όταν το ερώτημα (query) είναι πολύ μικρό, ή ακόμα και όταν οι δοσμένες ηχογραφήσεις ακούγονται παρόμοιες στον άνθρωπο αλλά στην

πραγματικότητα έχουν διαφορετικά ηχητικά σήματα. Ο αλγόριθμος θα πρέπει να είναι σε θέση να ξεχωρίσει τον καθαρό ήχο από την ηχογράφηση από τον οποιοδήποτε θόρυβο έχει προστεθεί σε αυτόν κατά τη διάρκεια της. Επειδή οι αλγόριθμοι πρέπει να παράγουν πανομοιότυπους κατακερματισμούς σε ήχο που ακούγεται ίδιος στο ανθρώπινο αυτί αλλά μπορεί να μην είναι αποθηκευμένος στον ίδιο δίσκο, κάποιες τεχνικές κατακερματισμού όπως ο κρυπτογραφικός είναι ακατάλληλες για τις συγκεκριμένες διαδικασίες. Η συγκεκριμένη τεχνική δίνει διαφορετικές εξόδους για παρόμοιες αλλά όχι ακριβώς ίδιες εισόδους. Δυο ηχητικά αποσπάσματα που ακούγονται το ίδιο στο ανθρώπινο αυτί μπορούν να έχουν μια εντελώς διαφορετική μορφή όταν είναι αποθηκευμένα στον σκληρό δίσκο ενός υπολογιστή. Σχεδόν κάθε αλγόριθμος βασίζεται στα δεδομένα του πεδίου των συχνοτήτων που προκύπτουν μέσα από τον κατακερματισμό. Για να μεταφερθούν τα δεδομένα από το πεδίο του χρόνου στο πεδίο συχνοτήτων χρησιμοποιείται ο διακριτός μετασχηματισμός Fourier (DFT). Ο πιο γνωστός αλγόριθμος για αυτή τη διαδικασία ονομάζεται Fast Fourier Transform (FFT) και πρόκειται για έναν τρόπο να υπολογιστούν τα ίδια αποτελέσματα με το DFT αλλά σε μικρότερο χρονικό διάστημα. Μέσα από αυτή τη διαδικασία μπορούν να προκύψουν προσεγγιστικά λάθη τα οποία ωστόσο μπορούν να μειωθούν με την αύξηση των υπολογισμών [24][22]. Ο FFT δεν είναι ο μοναδικός τρόπος όμως για την μετατροπή το τομέα του χρόνου στον τομέα των συχνοτήτων. Μια άλλη διαδεδομένη μέθοδος ονομάζεται Wavelets, και πρόκειται «για μαθηματικές συναρτήσεις που διαχωρίζουν τα δεδομένα σε συνιστώσες διαφορετικών συχνοτήτων και στην πορεία μελετάνε την κάθε συνιστώσα αναλύοντας την ανάλογα την κλίμακα της» (IEEE, 1995) [34]. Ο μετασχηματισμός μέσω του wavelet προσφέρει καλύτερα αποτελέσματα όσον αφορά την αναζήτηση χαρακτηριστικών μέσα από το φάσμα, παρόλα αυτά χρησιμοποιείται περισσότερο στον τομέα της οπτικής και δεν έχουν υπάρξει ως τώρα σημαντικές δημοσιεύσεις για τον τομέα του audio fingerprinting.

Ένα σύστημα αναγνώρισης ήχου με βάση το περιεχόμενό του αποτελείται από δύο βασικά μέρη: τη μέθοδο δημιουργίας /εξαγωγής του ακουστικού αποτυπώματος και την αναζήτησή και αναγνώριση του στη βάση δεδομένων.

Δημιουργία: Μέσα από μουσικές δουλειές καλλιτεχνών εξάγονται τα ακουστικά αποτυπώματα αναφοράς των τραγουδιών. Αυτά τα αποτυπώματα αποτελούν μια συνεχόμενη ροή από ακουστικά αποτυπώματα που εξάγονται απ'όλη τη διάρκεια του τραγουδιού. Αυτά καταχωρούνται στη βάση δεδομένων και κάθε ακουστικό αποτύπωμα αναφοράς συνδέεται με τα μεταδεδομένα που αντιστοιχούν στο τραγούδι από το οποίο προήλθε.

Αναγνώριση: Αφού εξαχθεί το ακουστικό αποτύπωμα από το άγνωστο μουσικό κομμάτι, αναζητείται στη βάση δεδομένων. Αν γίνει ταυτοποίηση, τότε ανακτώνται τα μεταδεδομένα που αντιστοιχούν σε αυτό κι έτσι γίνεται η αναγνώριση.

Κατά την εξαγωγή των ακουστικών αποτυπωμάτων η διαδικασία που ακολουθείται είναι ότι βασισμένο σε εύρωστα χαρακτηριστικά του σήματος εξάγεται ένα ακουστικό αποτύπωμα που οι διαστάσεις του είναι κατά πολύ μικρότερες από αυτές του ίδιου του ακουστικού αντικειμένου. Τα βήματα της διαδικασίας είναι τα παρακάτω:[24] [30] [27]

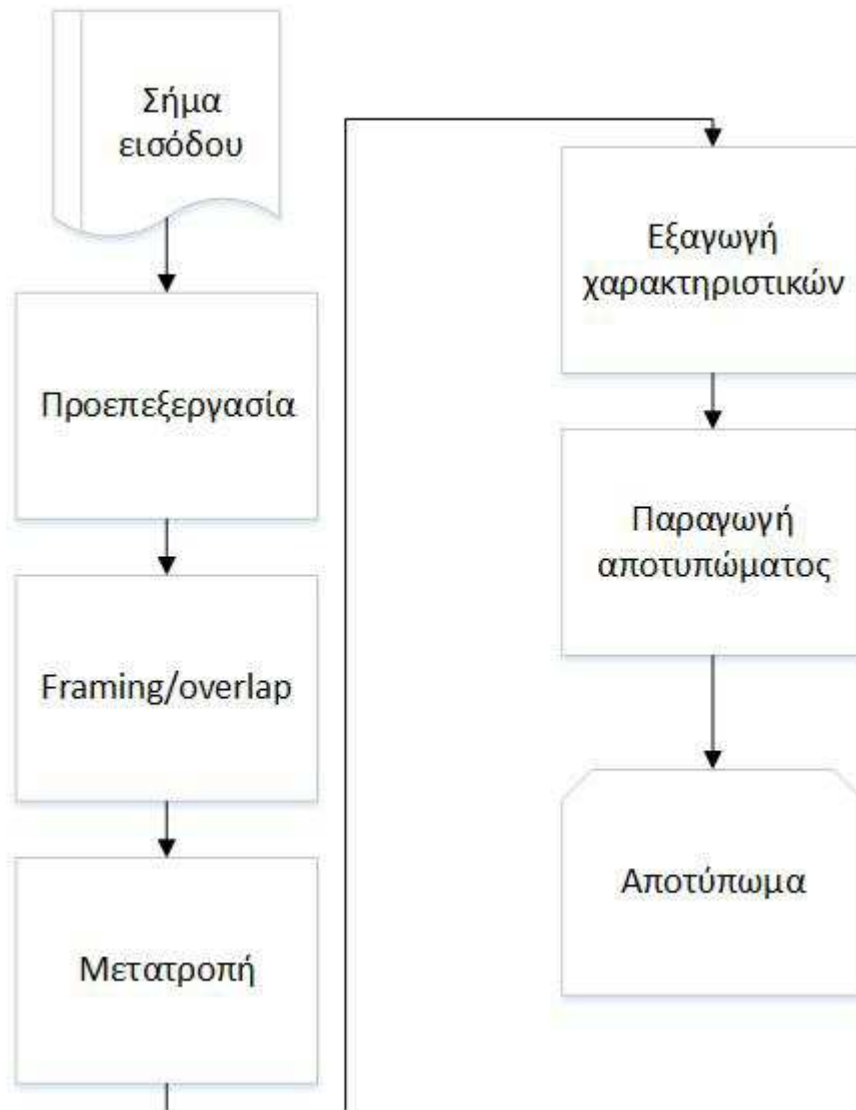
Preprocessing : Αρχικά το σήμα ψηφιοποιείται, αν χρειάζεται, και στη συνέχεια μετατρέπεται σε μια πρότυπη μορφή, π.χ. μονοφωνικό, 16 bits PCM, συχνότητας 5 kHz. Κάποιες φορές το σήμα υφίσταται επεξεργασία ώστε να προσομοιώνει το κανάλι μεταφοράς του, π.χ. υπόκειται σε ζωνοπερατό φίλτράρισμα, για αναγνώριση μέσω τηλεφώνου ή υπόκειται σε GSM κωδικοποίηση/αποκωδικοποίηση σε ένα σύστημα αναγνώρισης μέσω κινητού τηλεφώνου.

Framing & Overlap: Εδώ το σήμα θεωρείται στάσιμο για ένα διάστημα μερικών milliseconds. Στη συνέχεια, το σήμα χωρίζεται σε frames σε κάθε ένα από τα οποία εφαρμόζεται κάποιο πολλαπλασιαστικό παράθυρο, π.χ. Hanning, για να ελαχιστοποιηθούν οι ασυνέχειες στην αρχή και το τέλος κάθε frame. Επίσης, πρέπει να υπάρχει και μεγάλη επικάλυψη μεταξύ των frames ώστε να υπάρχει ευρωστία σε χρονικές ολισθήσεις (time-shifting).

Transformation: Στη συνέχεια κάθε frame μετασχηματίζεται. Οι περισσότερες μέθοδοι χρησιμοποιούν μετασχηματισμούς από το πεδίο του χρόνου στο πεδίο της συχνότητας. Σκοπός είναι να επιτευχθεί κυρίως συμπίεση και αφαίρεση θορύβου. Οι μετασχηματισμοί που χρησιμοποιούνται συνήθως είναι οι: Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT) , Haar Transform, Walsh-Hadamard Transform, Modulated Complex Lapped Transform και άλλοι.

Εξαγωγή χαρακτηριστικών: Σε αυτό το βήμα εξάγονται τα χαρακτηριστικά στα οποία θα βασιστεί ο υπολογισμός του ακουστικού αποτυπώματος. Αναλόγως με την τεχνική, σε αυτό το βήμα μπορούμε να δούμε μία πλειάδα χαρακτηριστικών. Τα χαρακτηριστικά που χρησιμοποιούνται κυρίως είναι οι συντελεστές Fourier, οι Mel-Frequency Cepstrum Coefficients (MFCC) , το Spectral Flatness Measure (SFM) , οι ενέργειες συγκεκριμένων μπαντών του φάσματος , οι Linear Predictive Coding (LPC) coefficients και άλλα.

Post-processing: Σε κάποιες τεχνικές τα χαρακτηριστικά που έχουν εξαχθεί από το προηγούμενο βήμα χρησιμοποιούνται κατευθείαν για να δημιουργήσουν το ακουστικό αποτύπωμα. Σε κάποιες άλλες μεθόδους όμως, χρειάζεται και μια περαιτέρω επεξεργασία που γίνεται σε αυτό το βήμα. Έτσι, τα χαρακτηριστικά μπορεί να κανονικοποιηθούν , να παραγωγιστούν, να υποστούν περαιτέρω μετασχηματισμούς, όπως Oriented Principal Component Analysis (OPCA), να κβαντιστούν κτλ. Επίσης, εδώ από κάποιες τεχνικές γίνεται και μοντελοποίηση των ακουστικών αποτυπωμάτων με διάφορες μεθόδους, όπως π.χ. με Hidden Markov Models (HMM) .



Εικόνα 3: βήματα εξαγωγής ακουστικού αποτυπώματος [24]

Περίληπτικά λοιπόν η διαδικασία ξεκινά με το πρώτο βήμα το οποίο είναι η μετατροπή της εισόδου του σήματος σε mono signal και χαμηλώνοντας το sampling rate από αυτό του CD που βρίσκεται στα 44,100Hz. Στη συνέχεια ένας αλγόριθμος παίρνει το ηχητικό σήμα του τομέα του χρόνου και το μετατρέπει σε σήμα του τομέα της συχνότητας από όπου θα εξαχθούν οι περισσότερες πληροφορίες. Το Framing και overlap καθορίζει πόσα δείγματα θα εξεταστούν κατά τον υπολογισμό ενός μετασχηματισμού σήματος στον τομέα του χρόνου. Κατά τη διαδικασία εξαγωγής των χαρακτηριστικών επιλέγονται βασικά χαρακτηριστικά από το σήμα που έχει υποστεί την μετατροπή, τα οποία χαρακτηρίζουν τον ήχο. Στο τέλος αφού τα χαρακτηριστικά έχουν επιλεγεί και εξαχτεί από το σήμα πρέπει να μετατραπούν σε μια αναπαράσταση του αποτυπώματος που μπορεί να αποθηκευτεί στη βάση δεδομένων και να συγκριθεί με άγνωστα σήματα. Μόλις το αποτύπωμα ενεργοποιηθεί, πρέπει να αποθηκευτεί σε μια ενδεικτική βάση δεδομένων. Η αριθμητική εκπροσώπηση των αποτυπωμάτων είναι συνήθως πολύ βαριά ώστε να χρησιμοποιηθεί για αναγνώριση, οπότε χρειάζεται κάτι διαφορετικό και μικρότερο για

να επιτευχθεί αυτό. Κάτι όπως το όνομα του καλλιτέχνη ή ο τίτλος του άλμπουμ. Το σύστημα θα ξεκινήσει τη διαδικασία της αναζήτησης η οποία θα πρέπει να είναι σε θέση να λάβει ένα άγνωστο σήμα ως είσοδο και στη πορεία θα πρέπει να το συγκρίνει και να ταυτοποιήσει το αποτύπωμα του με ένα από τα αποτυπώματα που βρίσκονται στην αναφορική βάση δεδομένων. Στην πορεία επιστρέφει τα καλύτερα αποτελέσματα της αναζήτησης, και την τοποθεσία του τραγουδιού. Αν το ζητούμενο τραγούδι δεν βρίσκεται στη βάση δεδομένων, το σύστημα θα αναφέρει ότι δεν βρίσκεται εκεί αντί να δώσει κάποια λανθασμένη απάντηση.

Η αναζήτηση ενός ακουστικού αποτυπώματος στη βάση δεδομένων είναι μια ασαφής αναζήτηση (fuzzy search), για το ποιο ακουστικό αποτύπωμα της βάσης δεδομένων μοιάζει πιο πολύ με το ακουστικό αποτύπωμα που έχει εξαχθεί από το άγνωστο σήμα [30]. Έτσι, αναλόγως την τεχνική εξαγωγής του ακουστικού αποτυπώματος υπάρχει και μια διαφορετική μέθοδος αναζήτησης. Οι πιο πολλές όμως ακολουθούν μια κοινή ιδέα: τη δημιουργία μιας δομής δεδομένων με ένα ευρετήριο (index), ώστε όταν εμφανίζεται ένα ακουστικό αποτύπωμα προς αναζήτηση να μη γίνονται όλες οι δυνατές συγκρίσεις μεταξύ του αποτυπώματος αναζήτησης και αυτών της βάσης δεδομένων, κάτι που θα ήταν πολύ χρονοβόρο. Επιπλέον, κάτι που είναι βασικό στην αναζήτηση των ακουστικών αποτυπωμάτων είναι το είδος της απόστασης που θα χρησιμοποιηθεί κατά τη σύγκριση. Κατά τη διάρκεια της αναζήτησης χρησιμοποιούνται διάφορων ειδών αποστάσεις (distance metrics) από τις διάφορες τεχνικές audio fingerprinting. Κάποιες από τις αποστάσεις που χρησιμοποιούνται στη σύγκριση των ακουστικών αποτυπωμάτων είναι η Ευκλείδεια απόσταση ή κάποιες διαφοροποιημένες εκδόσεις της, η απόσταση Manhattan, η απόσταση Hamming και άλλες. Υπάρχουν δυο βασικές ιδέες για την αύξηση της ακριβείας των ακουστικών αποτυπωμάτων. Η πρώτη ιδέα ορίζει ότι θα πρέπει να παρθεί ένα μεγαλύτερο δείγμα ή να εξαχθούν περισσότερα «μοναδικά» δεδομένα των χαρακτηριστικών του ήχου ή, η εξαγωγή περισσότερων πληροφοριών από ένα δείγμα κανονικού μεγέθους από μεγαλύτερα δείγματα. Το αρνητικό είναι ότι δημιουργείται η πιθανότητα να προστεθεί παραμόρφωση η οποία μειώνει την ακρίβεια και σαν αποτέλεσμα έχει να φέρει αντίθετα αποτελέσματα. Το να ληφθούν περισσότερα δεδομένα από ένα δείγμα αυξάνει επίσης την πιθανότητα να έχουμε false positive κατά το ταίριασμα ενώ ταυτόχρονα κάθε μεγάλο δείγμα θα έχει μεγαλύτερη δυνατότητα για μεταβολές και ως αποτέλεσμα μοναδικότητα. Θα έχει επίσης και πιο πολλές πιθανότητες να ταιριάζει με κάποιο άλλο δείγμα. Με τα δείγματα μεγαλύτερου μεγέθους αυξάνεται το μήκος του ήχου που επιχειρείται να αναγνωρισθεί, αλλά παράγει μόνο έναν μικρό κατακερματισμό (hash) ανά δείγμα και έχει σαν αποτέλεσμα ότι κάθε μεταβολή μειώνεται σε τέτοιο βαθμό ώστε οι γρήγορες μεταβολές χάνονται ενώ προηγουμένως θα έπρεπε να έχουν κατακερματιστεί και αντιστοιχηθεί. Η δεύτερη ιδέα περιλαμβάνει το να ληφθούν υπόψη περισσότερα από απλά την τονικότητα και τον χρόνο όταν γίνεται ο υπολογισμός του κατακερματισμού και η αναζήτηση για όμοια αποτυπώματα. Μπορούμε να προσθέσουμε περισσότερους μοναδικούς παράγοντες στην αναζήτηση, κάτι που πιθανότατα να μειώσει τον χρόνο αναζήτησης και την πιθανότητα για false positives. Παρόλο που δυο μουσικά κομμάτια μπορεί να έχουν παρόμοια αποτυπώματα λόγω παρεμβολών από θόρυβο, μπορούν να διαχωριστούν κατά την ανάλυση τους μέσα από τη κλίμακα που είναι γραμμένα ή το τέμπο τους. Περιορίζοντας την αναζήτηση σε

συγκεκριμένα στοιχεία όπως το τέμπο μειώνεται η λίστα με τα αντικείμενα που αναζητούνται και αντίστοιχα μειώνεται και ο χρόνος αναζήτησης.

Παρακάτω παρουσιάζονται κάποιες προτάσεις αλγορίθμων [23][26][28][30]

Ο αλγόριθμος των Haitsma και Kalker προτείνει ότι το πλάνο της εξαγωγής αποτυπωμάτων θα πρέπει να βασίζεται σε μια γενική προσέγγιση της ροής. Δηλαδή, παίρνοντας ένα σήμα ήχου και πλαισιώνοντας το σε παράθυρα μήκους 370ms για κάθε 11.6ms και δίνοντας του έναν παράγοντα της τάξεως του 31/32 ώστε ένα υπό-αποτύπωμα να εξάγεται κάθε 11.6ms. Στην πορεία υπολογίζεται ο FFT κάθε πλαισίου. Ένα άγνωστο δείγμα ηχητικού σήματος προς αναγνώριση θα πρέπει να είναι μεγέθους 3-30 δευτερόλεπτων ώστε ο αλγόριθμος να μπορεί να κάνει την αντιστοίχιση. Οι Haitsma και Kalker υπέδειξαν ότι το πρόσημο των εναλλαγών ενέργειας ανάμεσα σε διαφορετικές ήταν αποτελεσματικό για μουσική αναγνώριση και επίσης ισχυρό απέναντι σε εισόδους που έχουν υποστεί αλλοιώσεις όπως συμπίεση.

Ο Avery Wang δίνει έναν χρόνο αναζήτησης της τάξεως των 5 ως 500ms για μια βάση δεδομένων που περιέχει 20 χιλιάδες tracks αποθηκευμένα σε έναν ηλεκτρονικό υπολογιστή. Ο Dan Ellis δημιούργησε έναν γενικό κώδικα για το MATLAB και στην πορεία ο Robert Macrae τον τροποποίησε ώστε να είναι κατάλληλος για χρήση σε περιβάλλον windows. Ο αλγόριθμος που προτείνεται έχει ως σκοπό τη χρήση του φασματογράφου. Συνήθως ο φασματογράφος χωρίζεται σε μικρά κομμάτια που ονομάζονται παράθυρα ή frames. Οι διαφορές με τους αλγορίθμους αποτυπωμάτων περιλαμβάνουν το τι ποσοστό του πλαισίου επικαλύπτεται, και πως ορίζεται το αποτύπωμα στο πλαίσιο την αποθήκευση και την αναζήτηση των αποτυπωμάτων. Ο αλγόριθμος του Shazam για παράδειγμα χρησιμοποιεί τις ενεργειακές κορυφές που προκύπτουν στο πλαίσιο και σχηματίζουν ορόσημα φασματικών ζευγών. Τα τοπικά μέγιστα μέσα σε ένα καθορισμένο τμήμα ομαδοποιούνται σε ζεύγη, οι τιμές του κατακερματισμού υπολογίζονται και συγκρίνονται και στην πορεία η είσοδος με τα περισσότερα «χτυπήματα» επιστρέφεται σαν ταυτοποίηση. [23]

Ο αλγόριθμος PRH δημιουργήθηκε από τους Haitsma *et al* κάτω από την επίβλεψη της Philips και πουλήθηκε στην Grace note. Αρχικά το σήμα μετατρέπεται σε mono και μειώνεται η δειγματοληψία του στα 5,512.5 Hz. Στην πορεία χωρίζεται σε πλαίσια των 371ms (2048 δείγματα) τα οποία έχουν μια υπέρθεση της τάξεως του 96% (31/32). Αυτή η ισχυρή υπέρθεση χρησιμοποιείται ώστε να αποτρέψει οποιαδήποτε χρονική ευθυγράμμιση των πλαισίων. Τα πλαίσια που χρησιμοποιούνται για το αποτύπωμα μεταβάλλονται κατά 11.6ms, διαφορετικά έχουμε έναν μέγιστο βαθμό απόκλισης μεταξύ τους στα 5.8ms. Το κάθε πλαίσιο παραθυρώνεται και υπολογίζεται το περιοδόγραμμα του. Το φάσμα του χωρίζεται λογαριθμικά σε 33 ζώνες συχνοτήτων σε ένα εύρος των 300-2000 Hz. Με αυτό τον τρόπο κάθε νότα έχει τη δίκια της ζώνη συχνοτήτων. Το αποτύπωμα είναι βασισμένο στα κατώτερα σημεία του φάσματος αφού περιέχουν την περισσότερη ενέργεια και η οποία διατηρείται σε περιπτώσεις παραμορφώσεων.[25]

1.4.1. Αποτελέσματα

Κάθε αρχείο στο σύστημα τροποποιείται με τον ίδιο τρόπο ώστε να παράγει ένα σήμα πληροφοριών όπου κάποιες από αυτές ίσως υπάρχουν μέσα στη βάση δεδομένων αλλά κάποιες ίσως και να μη υπάρχουν. Όταν δίνεται ένα τέτοιο σήμα λοιπόν το σύστημα μπορεί να δώσει πίσω δυο απαντήσεις : η μια είναι ότι δεν υπάρχει κάποια γνωστή ηχογράφηση στη βάση δεδομένων που να ταιριάζει με τις πληροφορίες που δώσαμε και η άλλη είναι η επιστροφή ενός αποτελέσματος που σύμφωνα με το σύστημα ταιριάζει καλύτερα με τις πληροφορίες που δώσαμε προς αναζήτηση.

Βασισμένο σε αυτόν τον παράγοντα λοιπόν το σύστημα αξιολογεί και κατηγοριοποιεί τις απαντήσεις στις αντίστοιχες κατηγορίες: [24]

1. *True positive*. όταν τα 2 αποτυπώματα προέρχονται από τα ίδια tracks και η σύγκριση δείχνει ότι τα 2 tracks είναι ισοδύναμα.
2. *True negative*. Τα δυο αποτυπώματα είναι από διαφορετικά tracks και η σύγκριση δείχνει ότι δεν είναι ισοδύναμα.
3. *False positive*. Τα δυο αποτυπώματα είναι από διαφορετικά tracks, αλλά η σύγκριση δείχνει ότι είναι ισοδύναμα.
4. *False negative* . Τα δυο αποτυπώματα προέρχονται από το ίδιο track , αλλά η σύγκριση δείχνει ότι δεν είναι ισοδύναμα .
5. *False accept*: το σύστημα αναφέρει ότι έχει βρει μια ηχογράφηση όταν αυτή δεν υπάρχει μέσα στη βάση δεδομένων.

1.4.2. Αλγόριθμος εξαγωγής

Οι περισσότεροι αλγόριθμοι εξαγωγής αποτυπωμάτων χρησιμοποιούν μια κοινή μεθοδολογία. Το ηχητικό σήμα χωρίζεται σε πλαίσια και για κάθε πλαίσιο υπολογίζεται ένα σετ από χαρακτηριστικά. Τα χαρακτηριστικά αυτά επιλέγονται με την προϋπόθεση να είναι όσο το δυνατόν αμετάβλητα στις μεταβολές του σήματος. Τέτοια χαρακτηριστικά είναι καλά γνωστά χαρακτηριστικά του ήχου, όπως συντελεστές Fourier, Mel Frequency Cepstral Coefficients (MFCC) , φασματική επιπεδότητα , ευκρίνεια , Linear Predictive Coding (LPC) συντελεστές και άλλα. Χρησιμοποιούνται επίσης ποσοτικοί παράγοντες όπως τα παράγωγα, τα μέσα και οι διακυμάνσεις των χαρακτηριστικών του ήχου. Τα χαρακτηριστικά που εξάγονται χαρτογραφούνται σε πιο συμπαγής αναπαραστάσεις που ονομάζονται υπό- αποτυπώματα. Η γενική διαδικασία μετατρέπει μια ροή ήχου σε μια ροή υπό- αποτυπωμάτων αλλά αυτά τα υπό- αποτυπώματα δεν περιέχουν αρκετά δεδομένα ώστε να γίνει ταυτοποίηση με κάποιο ηχητικό απόσπασμα. Η βασική μονάδα που περιέχει αρκετά στοιχεία ώστε να προσδιορίσει ένα ηχητικό απόσπασμα ονομάζεται *fingerpint block*. Για την εξαγωγή των χαρακτηριστικών λοιπόν εξάγονται αποτυπώματα των 32-bit για κάθε παύση των 11.6ms. Ένα *fingerpint block* αποτελείται από 256 υπό- αποτυπώματα, που αντιστοιχούν σε μια ανάλυση των 3 δευτερόλεπτων. Το σήμα χωρίζεται σε επικαλυπτόμενα πλαίσια τα όποια έχουν μέγεθος 0.37 δευτερόλεπτων και σταθμίζονται από ένα παράθυρο Hanning το οποίο

έχει παράγοντα υπέρθεσης 31/32. Αυτό έχει σαν αποτέλεσμα την εξαγωγή ενός υπό- αποτυπώματος κάθε 11.6 ms. Η μεγάλη υπέρθεση σιγουρεύει ότι σε κάθε περίπτωση τα υπό- αποτυπώματα του ηχητικού αποσπάσματος θα είναι όμοια με αυτά του ίδιου αποσπάσματος στην βάση δεδομένων. Στην πορεία έρχεται ο αλγόριθμος αναζήτησης ο οποίος βρίσκει τα αποτυπώματα που ταιριάζουν καλύτερα σε μια βάση δεδομένων μέσα από κάποιο μέτρο σύγκρισης. Προκειμένου να επιταχυνθεί η διαδικασία αναζήτησης και να αποφευχθεί μια διαδοχική σάρωση της βάσης δεδομένων, χρησιμοποιούνται διαφορές στρατηγικές ώστε να απορριφτούν όσο το δυνατόν πιο γρήγορα αποτυπώματα τα οποία δεν έχουν κάποια ομοιότητα μεταξύ τους. [29]

1.4.3. Προϋποθέσεις

Για να λειτουργεί ικανοποιητικά και αξιόπιστα ένα σύστημα ακουστικών αποτυπωμάτων θα πρέπει να πληροί κάποιους περιορισμούς και προϋποθέσεις : [24][30][25]

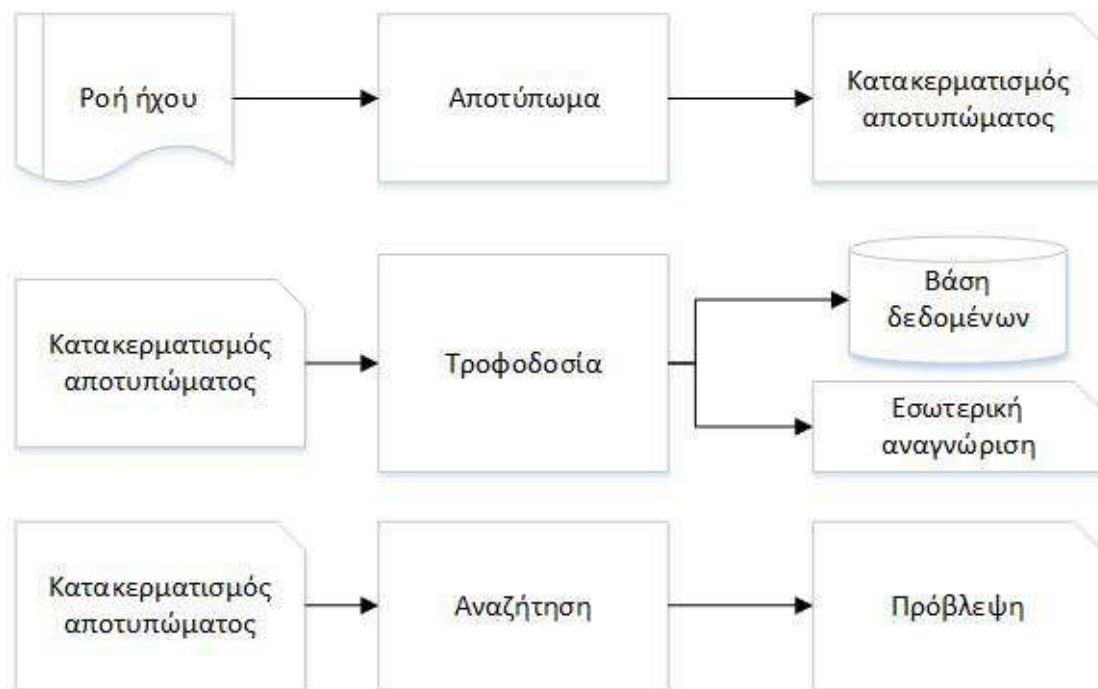
1. Η εξαγωγή του ακουστικού αποτυπώματος να μην είναι υπολογιστικά πολύπλοκη, να είναι γρήγορη. Αυτό κάνει δυνατή τη χρήση τους σε εφαρμογές με περιορισμένες υπολογιστικές δυνατότητες.
2. Τα αποτυπώματα πρέπει να είναι όσο το δυνατό μικρότερα ώστε να μην υπάρχουν μεγάλες απαιτήσεις σε μνήμη και χωρητικότητα.
3. Το σύστημα Θα πρέπει να μπορεί να κάνει αξιόπιστη αναγνώριση ανεξάρτητα από τη συμπίεση, τη παραμόρφωση ή τις παρεμβολές από το κανάλι μετάδοσης. Δηλαδή, το ακουστικό αποτύπωμα που εξάγεται πρέπει να είναι εύρωστο. Πρέπει να μπορεί να αναγνωρίσει ακουστικά αντικείμενα με τη χρήση ενός αποσπάσματος λίγων δευτερολέπτων, αντιμετωπίζοντας οποιασδήποτε μορφής παραμόρφωση του σήματος όπως χρονική ολίσθηση, ισοστάθμιση, θόρυβο, D-A/A-D μετατροπές, κωδικοποιήσεις φωνής και ήχου, όπως GSM και MP3, και άλλες.
4. Η αναζήτηση στη βάση δεδομένων πρέπει να είναι γρήγορη και υπολογιστικά απλή ώστε η ταυτοποίηση να γίνεται γρήγορα ακόμη και σε πολύ μεγάλες βάσεις δεδομένων και χρησιμοποιώντας περιορισμένους υπολογιστικούς πόρους.
5. Η μέθοδος αναζήτησης πρέπει να είναι αξιόπιστη και να δίνει σωστά αποτελέσματα έχοντας τη δυνατότητα να κάνει την αναγνώριση ανάμεσα από έναν μεγάλο αριθμό τραγουδιών της βάσης δεδομένων. Έτσι, τα αποτελέσματα της αναζήτησης πρέπει να είναι ακριβή, με όσο το δυνατόν λιγότερα false positives και false negatives.
6. Οι αλγόριθμοι θα πρέπει να είναι αξιόπιστοι, δηλαδή να ελαχιστοποιούν τον αριθμό των false positives που επιστρέφονται. Να μπορούν να εντοπίσουν τον ήχο ακόμα και αν το ερώτημα έχει παραμορφωθεί κατά τη διάρκεια της ηχογράφησης
7. Να μπορούν να αναγνωρίζουν σωστά ένα οποιοδήποτε τραγούδι μέσω ενός αποσπάσματος μόλις λίγων δευτερόλεπτων που θα έχει παρθεί από οποιοδήποτε σημείο του και όχι αποκλειστικά την αρχή του τραγουδιού.

8. Πρέπει να είναι ευέλικτοι ώστε να μπορούν να παράγουν αποτυπώματα από οποιαδήποτε μορφοποίηση ενός μουσικού αρχείου σε έναν υπολογιστή.
9. Τα αποτυπώματα θα πρέπει να είναι υπολογιστικά εύκολα να δημιουργηθούν και η εισαγωγή νέων τραγουδιών στη βάση δεδομένων δεν θα πρέπει να επηρεάζει την ταχύτητα κατά την αναζήτηση.

1.4.4. Σχεδιασμός

Ένα σύστημα ακουστικών αποτυπωμάτων αναλύεται σε δυο βήματα: εισαγωγή του ήχου σε μια βάση δεδομένων και αναζήτηση μέσα σε αυτή για μεταδεδομένα αγνώστων ηχητικών κομματιών. Και τα 2 αυτά μέρη περιλαμβάνουν μια διαδικασία που παράγει αποτυπώματα μέσα από ένα ηχητικό απόσπασμα. Το μέρος που αφορά το αποτύπωμα παίρνει ένα ηχητικό απόσπασμα και υπολογίζει τα αριθμητικά αποτυπώματα που αντιπροσωπεύουν τον ήχο.

Το μέρος της απορρόφησης παίρνει το αποτύπωμα ενός αρχείου ήχου και το εισάγει στην βάση αναφοράς του αλγορίθμου. Η μέθοδος της απορρόφησης επιστρέφει ένα μικρό και μοναδικό αναγνωριστικό το οποίο βοηθά στο να χαρτογραφηθούν οι κώδικες των αποτυπωμάτων. Το πλαίσιο αξιολόγησης διατηρεί έναν χάρτη μεταξύ των αρχείων και των ταυτοτήτων των αποτυπωμάτων τους. Συγκρίνοντας αυτές τις ταυτότητες μπορούμε να έχουμε ένα μέτρο σύγκρισης και αξιολόγησης των αλγόριθμων. Το μέρος της αναζήτησης παίρνει ένα αποτύπωμα που έχει παραχθεί από ένα μικρό μέρος του ήχου και ψάχνει για αυτό στην βάση δεδομένων του αλγορίθμου. Στην πορεία επιστρέφει την εσωτερική ταυτότητα που πιστεύει ότι ανήκει στο τραγούδι. Το πλαίσιο αξιολόγησης αποθηκεύει τις εσωτερικές ταυτότητες και τα αναγνωριστικά και τα συγκρίνει με τις πραγματικές ταυτότητες των αρχικών αρχείων που χρησιμοποιήθηκαν για την αναζήτηση



Εικόνα 4 :Οι ενότητες του ακουστικού αποτυπώματος. [24]

1.5. State of the Art

Ο τομέας της ανάκτησης μουσικής πληροφορίας αναπτύσσεται όλο και πιο γρήγορα τα τελευταία χρόνια. Αυτό δίνει την ευκαιρία σε απλούς χρήστες αλλά και μεγάλες εταιρίες να χρησιμοποιήσουν εφαρμογές όπως το ακουστικό αποτύπωμα ο καθένας για δικούς του σκοπούς. Οι απλοί χρήστες για παράδειγμα μπορούν να το χρησιμοποιήσουν ώστε να αναγνωρίσουν ένα άγνωστο τραγούδι με τη βοήθεια του κινητού τηλεφώνου τους, να ελέγξουν αν υπάρχει στη συλλογή τους, να ανακαλύψουν τραγούδια παρόμοια με αυτά που ήδη ακούνε ακόμα και να τα αγοράσουν. Οι εταιρίες από την άλλη ενδιαφέρονται περισσότερο για θέματα πνευματικών δικαιωμάτων, όπως για παράδειγμα της μουσικής που μοιράζεται μέσα από file-sharing δίκτυα και μέσω ακουστικών αποτυπωμάτων μπορούν να αναλύσουν ολόκληρες τις λίστες τους. Όσο ο τομέας της ανάκτησης πληροφοριών αναπτύσσεται τόσο εμφανίζονται νέες λύσεις και προτάσεις για την επίλυση των προβλημάτων και των αναγκών που παρουσιάζονται. Πολλές από αυτές βασίζονται πάνω στην δουλεία των Ke, Hoiem και Sukthankar, η οποία βασίζεται με τη σειρά της στην εφαρμογή τεχνικών computer vision στο φάσμα των αρχείων ήχου. Με τη χρήση του STFT (Short Time Fourier Transform), ο φασματογράφος περιέχει την ίδια ενέργεια όπως οι λογαριθμικά χωρισμένες ζώνες συχνοτήτων. Μέσα από τεχνικές μηχανικής μάθησης εντοπίζουν έναν αριθμό φίλτρων με την καλύτερη απόδοση. Αυτή η διεργασία έχει βρει εφαρμογή σε διάφορες εφαρμογές. Το libFingerprint του last.FM είναι μια διασκευή της δουλειάς των Ke et. al, με βασικές βελτιώσεις όσων αφορά την ταχύτητα. Η Google παρουσίασε μια σειρά από έγγραφα που περιγράφουν το Waverprint, μια τεχνική η οποία δεν βασίζεται στην μηχανική μάθηση αλλά στο wavelets. Ένας άλλος τύπος μουσικού αποτυπώματος μοντελοποιεί το ηχητικό σήμα χρησιμοποιώντας ένα ημίτονο το οποίο έχει διαμορφωθεί μέσα από παραμέτρους όπως το πλάτος, η φάση και η συχνότητα, και το οποίο λαμβάνει υπόψη τον θόρυβο που έχει παραμείνει. Για να εξαχθούν αυτά τα ημιτονοειδή μοντέλα πρέπει να χρησιμοποιηθεί ανάλυση Fourier κατά την οποία διαλέγονται οι ημιτονοειδής κορυφές με το μεγαλύτερο πλάτος. Το σύστημα αυτό μπορεί να αναλύσει αποσπάσματα ακόμα και του μεγέθους του ενός δευτερόλεπτου. Έχουν υπάρξει πολλές δημοσιεύσεις που αφορούν τον στόχο αναζήτησης ηχογραφήσεων από το ίδιο μουσικό έργο, όπως των Miotto και Larsen. Ο Larsen τονίζει ότι συστήματα όπως τα fdmf και libFoolD δεν είναι έμπιστα στο να αναγνωρίζουν τυχαίες ηχογραφήσεις. Ο Miotto έχρισε ένα στατιστικό μοντέλο που προβλέπει πιθανές διασκευές από ένα μουσικό έργο. Η αρχική ηχογράφηση κατακερματίζεται και εξάγονται διάφορα χαρακτηριστικά του ήχου. Στην πορεία χρησιμοποιούνται τεχνικές όπως τα Hidden Markov models ώστε να επιτευχθεί η αναγνώριση.[27]

1.5.1. Χρήσεις και εφαρμογές

Ο τομέας του ακουστικού αποτυπώματος μπορεί να βρει εφαρμογή σε μια πληθώρα ζητημάτων όπως παρακολούθηση μετάδοσης, μέτρηση ακροαματικότητας, εφαρμογές ομιλίας συλλογή μεταδεδομένων, αναζήτηση διπλοτύπων, πνευματικά δικαιώματα, ονομασία τραγουδιού, μουσικές προτάσεις, ακόμα και σε θέματα της αστυνομίας . Μπορεί δηλαδή να βρει εφαρμογή σε δημοσίους αλλά και ιδιωτικούς τομείς. [25] [27]

- Παρακολούθηση μετάδοσης

Για τις διαφημιστικές εταιρίες είναι πολύ σημαντικό οι διαφημίσεις τους να προβάλλονται συμφώνα με το πρόγραμμα που έχει συμφωνηθεί. Για αυτόν τον λόγο εταιρίες όπως η Clvolution Και η Nielsen Broadcast Data Systems προσφέρουν μια υπηρεσία η οποία ονομάζεται παρακολούθηση μετάδοσης (broadcast Monitoring). Η δουλειά αυτής της υπηρεσίας είναι να παρακολουθεί αυτόματα έναν αριθμό από ραδιοφωνικά και τηλεοπτικά κανάλια και να ψάχνει για ένα συγκεκριμένο περιεχόμενο, όπως για παράδειγμα διαφημίσεις και να καταγράφει πότε, που και για πόση ώρα μεταδόθηκε.

- Μέτρηση ακροαματικότητας

Δηλαδή να αναγνωρίσει ποιο πρόγραμμα προτιμάται περισσότερο από μια συγκεκριμένη ομάδα ανθρώπων. Μπορεί να δημιουργήσει στατιστικές ως προ το τι περιεχόμενο είναι διαθέσιμο στο internet ή να δημιουργήσει στατιστικές ανάμεσα στα μεταδεδομένα που έχουν συλλεχτεί μέσα από τα αποτυπώματα.

- Αστυνομικές εφαρμογές

Σε περιπτώσεις που έχει συμβεί κάποια εγκληματική πράξη όπως για παράδειγμα παιδική κακοποίηση η αστυνομία μαζεύει οπτικοακουστικό υλικό από τον τόπο του εγκλήματος. Έτσι μέσα σε αυτό το υλικό θα γίνει αναζήτηση για οποιοδήποτε περιεχόμενο που αφορά το θέμα που διερευνάται όπως συγκεκριμένες εικόνες ή εικόνες που εμφανίζονται για πρώτη φορά. Η δουλειά του αποτυπώματος είναι να αναζητήσει υλικό το οποίο έχει αναλυθεί σε παρελθοντικό χρόνο.

- Εντοπισμός μη εξουσιοδοτημένου υλικού

Οι κάτοχοι πνευματικών δικαιωμάτων θέλουν να ξέρουν που και πως χρησιμοποιούνται οι δημιουργίες τους. Ένας συνδυασμός από web crawlers και αποτυπώματα μπορεί να χρησιμοποιηθεί και να εντοπίσει το συγκεκριμένο υλικό και την πλατφόρμα στην οποία χρησιμοποιείται. Εάν ο κάτοχος των δικαιωμάτων το επιθυμεί μπορεί να επιβάλει να αποκλειστεί το υλικό του από οποιαδήποτε χρήση.

- Ονομασία τραγουδιού

Πολλές φορές κάποιος ακροατής ακούει ένα τραγούδι στο ραδιόφωνο του οποίου το όνομα δεν γνωρίζει. Με τη βοήθεια του κινητού τηλέφωνου του αρκεί μόνο να συλλέξει ήχο λίγων δευτερολέπτων μόνο και στην πορεία η συγκεκριμένη εφαρμογή υπολογίζει και ταιριάζει το αποτύπωμα . Τέλος επιστρέφει ένα μήνυμα που περιέχει όλες τις πληροφορίες όπως πχ : τίτλο, όνομα καλλιτέχνη, όνομα δίσκου κλπ.

- Συλλογή μεταδεδομένων και αναζήτηση διπλότυπων

Οι χρηστές ηλεκτρονικών υπολογιστών μέσα από το download έχουν τη δυνατότητα να συλλέξουν έναν τεράστιο αριθμό από μουσικά κομμάτια, τα οποία αποθηκεύονται στον σκληρό δίσκο του υπολογιστή τους. Ένας τέτοιο μεγάλος αριθμός όμως καθιστά εξαιρετικά δύσκολη την αναζήτηση και την οργάνωση των μουσικών συλλόγων τους. Για αυτόν τον λόγο υπάρχουν εφαρμογές που όχι μόνο οργανώνουν το μουσικό περιεχόμενο των χρηστών αλλά βοηθούν στην ανακάλυψη

τραγουδιών που υπάρχουν πάνω από μια φορές στον υπολογιστή και βοηθάμε ώστε να αυξηθεί ο αποθηκευτικός χώρος.

- Added value υπηρεσίες

Υπάρχουν πολλές υπηρεσίες που μπορούν να χρησιμοποιήσουν τα δεδομένα ενός τραγουδιού από τη στιγμή που η ταυτοποίηση του έχει πραγματοποιηθεί. Για παράδειγμα, προσφορές για αγορά του συγκεκριμένου τραγουδιού, μουσικές διαφημίσεις βασισμένες στα μουσικά ενδιαφέροντα του χρήστη, αναζήτηση στίχων και γενικές πληροφορίες όπως η βιογραφία του καλλιτέχνη.

- Audio tag

Το audio tagging μπορεί να αναλυθεί μέσα από δυο τομείς του: τον σχολιασμό και την ανάκτηση. Ο ηχητικός σχολιασμός παρουσιάζεται σαν θέμα ταξινόμησης της κάθε ετικέτας, εφόσον μιλάμε για έναν μεγάλο αριθμό ετικετών. Ο κάθε ταξινομητής ετικετών αποφασίζει εάν το ηχητικό απόσπασμα που δίνεται σαν είσοδος θα πρέπει να έχει μια συγκεκριμένη ετικέτα δίνοντας μια βαθμολογία σαν έξοδο. Η απόδοση μπορεί να αξιολογηθεί ανάλογα με το ποσοστό των ετικετών που έχουν επαληθευτεί σωστά, αφού ανάλογα το ποσοστό των σωστών επαληθεύσεων μεγαλώνει και η βαθμολογία. Στον τομέα της ηχητικής ανάκτησης, και αφού μια συγκεκριμένη ετικέτα έχει δοθεί ως ερώτημα, ξεκινά η διαδικασία του να ανακτηθεί το ηχητικό απόσπασμα που αντιστοιχεί σε αυτήν την ετικέτα. Αυτό μπορεί να γίνει μέσω ενός ταξινομητή ετικετών που αποφασίζει ανάλογα με τη βαθμολογία ποιο ηχητικό απόσπασμα είναι σχετικό με την ετικέτα. Τα αποσπάσματα τότε κατατάσσονται ανάλογα με τη σχετική βαθμολογία και αυτά με την υψηλότερη επιστρέφονται στον χρήστη.

- Πρόταση μουσικής

Η μουσική πρόταση αποτελεί ένα σημαντικό εργαλείο για πολλές σύγχρονες εφαρμογές, όπως μουσικές κοινότητες, online music stores, και συσκευές αναπαραγωγής όπως είναι τα MP3 players όπου χρησιμοποιείται για δημιουργία λιστών αναπαραγωγής. Η πιο κοινή μέθοδος είναι η άμεση ανάλυση του ηχητικού σήματος, κάτι που ονομάζεται *πρόταση μουσικής με βάση την ακουστική*. Όταν οι άνθρωποι μιλούν για «μουσική σύσταση» ή «μουσική ανακάλυψη» μιλούν συνήθως ένα από τα παρακάτω πράγματα:

- Ομοιότητα καλλιτέχνη ή τραγουδιού

Μια πολύ κοινή εφαρμογή στις μουσικές υπηρεσίες, πρόκειται ουσιαστικά για πρόταση καλλιτεχνών και τραγουδιών παρόμοιων με τα όσα έχει αναζητήσει ο χρήστης.

- Εξατομικευμένη πρόταση

Μοντελοποιώντας τις κινήσεις των χρηστών σε μια μουσική υπηρεσία(αναπαραγωγή, skips, βαθμολογίες, αγορές) η υπηρεσία ταιριάζει στο προφίλ του χρήστη μια λίστα από τραγούδια τα οποία πιστεύει ότι ο ίδιος δεν γνωρίζει.

- Δημιουργία Playlist

Κάτι που το χρησιμοποιούν οι περισσότεροι χρήστες. Η λίστα αναπαραγωγής μπορεί να προσωποποιηθεί ή όχι, και μπορεί να βρίσκεται μέσα σε κάποιον κατάλογο ή σε κάποια online μουσική υπηρεσία (Pandora, Spotify, iHeartRadio). Συνήθως ποικίλουν ως προς τους καλλιτέχνες και το είδος των τραγουδιών όσο εξελίσσονται και πολλές βασίζονται σε κάποια ειδή καθοδήγησης και αξιολόγησης όπως για παράδειγμα : thumbs up, skips, αγαπημένα κλπ.

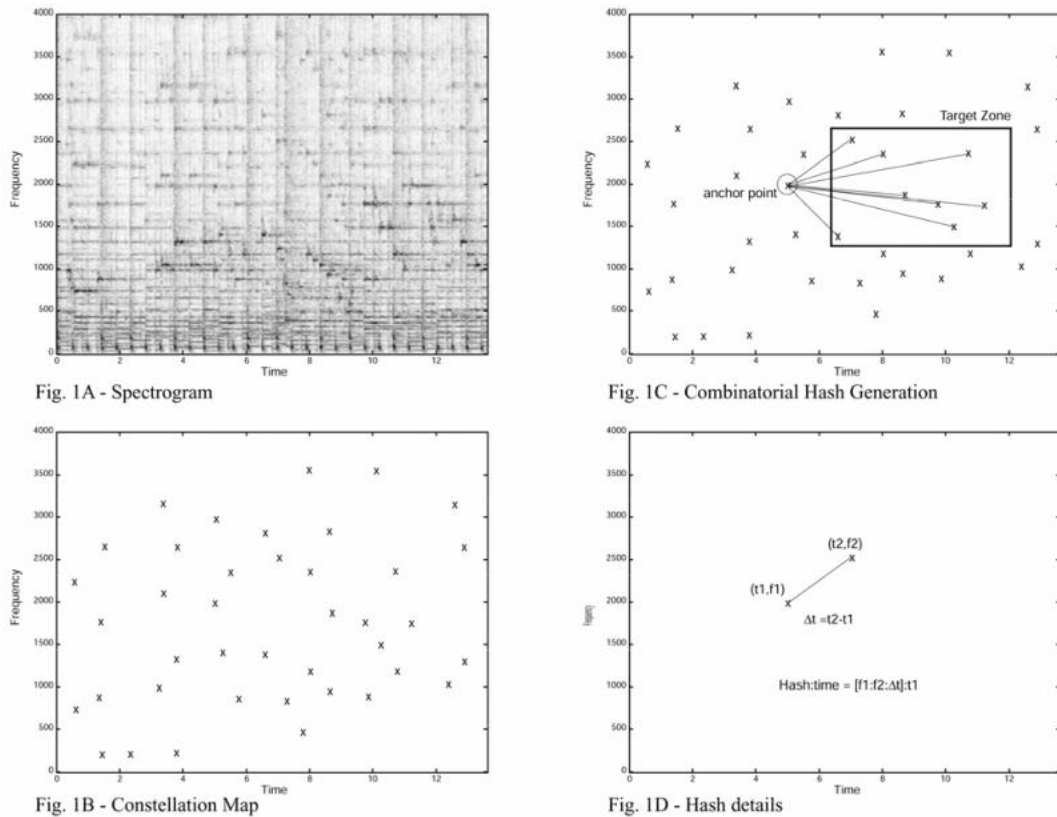
Οι περισσότεροι άνθρωποι ακόμα και αν δε το ξέρουν χρησιμοποιούν το ακουστικό αποτύπωμα μέσα από εφαρμογές όπως το Shazam, και το SoundHound, όπου και τα δυο χρησιμοποιούνται μέσω κινητού τηλεφώνου για αναγνώριση μουσικής μέσω του αέρα σε οποιοδήποτε μέρος, είτε αυτό είναι αυτοκίνητο είτε καφετερία. Η Gracenote, είναι μια ακόμα εταιρία που προσφέρει λογισμικό για κινητά τηλέφωνα, αλλά διαθέτει ακόμα λογισμικά τα οποία προορίζονται για εφαρμογές ψηφιακής αναπαραγωγής όπως, τα iTunes και Winamp έτσι ώστε να είναι πιο εύκολο να αναζητήσουμε μουσική, να ονομάσουμε, να κατηγοριοποιήσουμε και να ταυτοποιήσουμε. Το Gracenote χρησιμοποιείται ακόμα και για φωνητική αναγνώριση και βρίσκει εφαρμογή μαζί με το SYNC της Ford στον τομέα των συστημάτων επικοινωνίας των αυτοκινήτων. Οι χρήσεις του ακουστικού αποτυπώματος, όπως έχει αναφερθεί και πιο πάνω, δεν περιορίζονται μόνο στη δημιουργία λιστών αναπαραγωγής και στην ταυτοποίηση τραγουδιών. Συνθέτες και καλλιτέχνες το χρησιμοποιούν ώστε να αναγνωρίσουν μέρη τραγουδιών που ίσως κάποιος άλλος τα έχει χρησιμοποιήσει και αυτό αντιτίθεται στα πνευματικά τους δικαιώματα. Η αστυνομία το χρησιμοποιεί για την ανακάλυψη παράνομης μουσικής στους υπολογιστές διαφόρων χρηστών. Αν κάποιο μουσικό κομμάτι που δεν έχει κυκλοφορήσει ακόμα διαρρεύσει στο κοινό, οι εταιρίες παραγωγής μπορούν να το εντοπίσουν καθώς και να εντοπίσουν το ποιος το διέρρευσε. Όπως η αστυνομία έτσι και το YouTube, <http://www.youtube.com/>, χρησιμοποιεί ένα σύστημα ονόματι Content ID (YOUTUBE, 2008) το οποίο δημιουργήθηκε από την εταιρία Audible Magic, <http://www.audiblemagic.com/>. Το Content ID βοηθά τους ιδιοκτήτες των πνευματικών δικαιωμάτων στον εντοπισμό βίντεο στο YouTube τα οποία μπορούν να χρησιμοποιούν ήχο ή εικόνα που δεν τους ανήκει. Ο κάθε χρήστης μπορεί δημιουργήσει αποτυπώματα από διαφόρους ήχους ή προσωπικές του μουσικές δημιουργίες, ή ακόμα και να δημιουργήσει συστήματα φωνητικής αναγνώρισης. [27]

1.5.2. Open source εφαρμογές

Shazam

Οι πιο κοινοί και αξιόπιστοι αλγόριθμοι για αναγνώριση ήχου βασίζονται στην ιδέα των landmarks. Η λειτουργία του Shazam, το οποίο είναι και το πιο κοινό πρόγραμμα αναγνώρισης για κινητά, περιγράφεται από τον Avery Li-Chun Wang στην δημοσίευση με τίτλο “An Industrial Strength Audio Search Algorithm” (WANG, Avery Li-Chun, 2003). Περιγράφει το πώς μετατρέπουν τα δεδομένα από το πεδίο

του χρόνου στο πεδίο των συχνοτήτων όπου η ροή διαχωρίζεται σε μεγάλα τεμάχια δεδομένων και στην πορεία μετατρέπονται ώστε οι επιμέρους συχνότητες που βρίσκονται σε αυτά τα τεμάχια να αναγνωριστούν και στην πορεία να χρησιμοποιηθούν μαζί με τα δεδομένα του χρόνου για να υπολογιστεί ο κατακερματισμός όπως φαίνεται και στην εικόνα 5.



Εικόνα 5 : Μετατροπή δεδομένων/ υπολογισμός κατακερματισμού[23]

Στο 1A τα δεδομένα έχουν μετατραπεί στο πεδίο των συχνοτήτων όπου οι συχνότητες με τη μεγαλύτερη σπουδαιότητα ξεχωρίζουν στα πιο σκούρα σημεία του γραφήματος. Στο 1B ο αλγόριθμος έχει επιλέξει τα πιο σημαντικά χαρακτηριστικά του γραφήματος όπου η σημασία ορίζεται ως ένα σημείο χρόνου-συχνότητας το οποίο έχει μεγαλύτερη σπουδαιότητα από όλα τα άλλα σημεία στην γύρω περιοχή. Αν αποθηκεύονται δεδομένα από αυτό το σημείο στη βάση δεδομένων η αναζήτηση δεν θα μπορεί να δώσει αποτελέσματα που να είναι τόσο ακριβή όσο θα θέλαμε. Για αυτό το Shazam διαλέγει κάποια σημεία όπως αυτά στο 1C, και στην συνέχεια δημιουργεί τον κατακερματισμό ο οποίος βασίζεται στη σχέση μεταξύ του anchor point και των άλλων σημείων γύρω του όπως φαίνεται και στο 1D. Μόλις το Shazam έχει δημιουργήσει τη βάση δεδομένων με τους κατακερματισμούς, ξεκινά μια αναζήτηση για την εύρεση συσχετίσεων μεταξύ ενός δείγματος και των τιμών της βάσης δεδομένων. Μέσα από αυτό το Shazam έχει δημιουργήσει έναν αλγόριθμο ο οποίος είναι αρκετά γρήγορος ώστε να χρησιμοποιηθεί από κινητό τηλέφωνο το οποίο απλά στέλνει τα δεδομένα στους servers του Shazam για αναγνώριση. Τέλος η μέθοδος τους είναι πολύ αποτελεσματική ακόμα και σε περιπτώσεις υψηλού θορύβου ή κακών ηχογραφήσεων. [23]

MusicBrainz Picard

Είναι μια cross-platform εφαρμογή (Linux/Mac OS X/Windows) γραμμένη σε Python και είναι το επίσημο πρόγραμμα επισήμανσης του MusicBrainz . Υποστηρίζει την πλειοψηφία των μουσικών μορφοποιήσεων και έχει τη δυνατότητα να χρησιμοποιεί αποτυπώματα ώστε να πραγματοποιεί αναζητήσεις από CD. (<https://picard.musicbrainz.org/>)

AcoustID

Σκοπός του είναι η δημιουργία μιας βάσης δεδομένων ελεύθερης προς το ευρύ κοινό χρησιμοποιώντας τα μεταδεδομένα από τη βάση δεδομένων του Musicbrainz και να δημιουργήσει μια διαδικτυακή υπηρεσία για αναγνώριση αρχείων ήχου μέσω αυτής. Το περιεχόμενο αυτής της βάσης δεδομένων προσφέρεται από τους χρήστες της. (<https://acoustid.org/>)

Fingerprinter

Το fingerprinter του AcoustID είναι μια cross-platform GUI εφαρμογή που χρησιμοποιεί το Chromaprint για να υποβάλλει ακουστικά αποτυπώματα από τη μουσική συλλογή του χρήστη στην βάση δεδομένων του AcoustID . Δέχεται μόνο αρχεία που έχουν ετικέτα με προτεραιότητα σε αυτά που προέρχονται από το Picard ή το jaikoz της MusicBrainz. (<https://acoustid.org/fingerprinter>)

Chromaprint

Είναι ο βασικός ανταγωνιστής του AcoustID. Είναι μια client-side βιβλιοθήκη η οποία χρησιμοποιεί έναν δικό της αλγόριθμο για να εξάγει αποτυπώματα από οποιαδήποτε πηγή ήχου. Βασικός σκοπός της είναι να παρέχει ακριβείς ταυτοποιήσεις από ηχογραφήσεις. (<https://acoustid.org/chromaprint>)

Musicg

Είναι μια βιβλιοθήκη γραμμένη σε Java για ανάλυση ήχου, με σκοπό την εξαγωγή τόσο υψηλού αλλά και χαμηλού επιπέδου ηχητικών χαρακτηριστικών. Αυτό το API επιτρέπει στους προγραμματιστές να εξάγουν ηχητικά χαρακτηριστικά και να πραγματοποιούν λειτουργίες με τα δεδομένα του ήχου όπως διάβασμα κατευθείαν μέσα από μια ροή εισόδου. Παρέχει τα εργαλεία για ψηφιακή επεξεργασία και διαχειρίζεται την κυματομορφή και το φάσμα για σκοπούς έρευνας και ανάπτυξης. (<https://code.google.com/p/musicg/>)

The fdmf

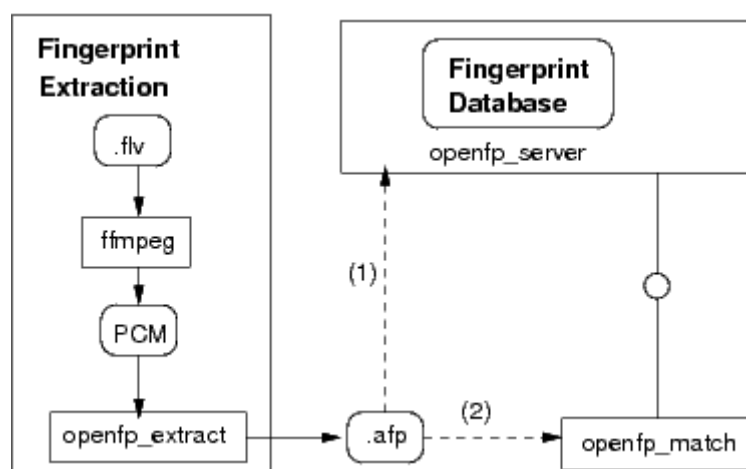
Πρόκειται για ένα πρόγραμμα το οποίο έχει ως σκοπό την εξεύρεση διπλότυπων μουσικών αρχείων σε μια συλλογή. Γραμμένο σε GNU C, και κάποια σημεία σε Perl το fdmf δημιουργεί αποτυπώματα από τα αρχεία ήχου και τα αποθηκεύει σε μια εσωτερική βάση δεδομένων. Ένα δεύτερο πρόγραμμα αντιστοιχεί τα αποτυπώματα με άλλα μέσα σε αυτή τη βάση και τυπώνει τα αποτελέσματα. (<http://freecode.com/projects/fdmf>)

Audfprint

Γραμμένο σε Matlab παίρνει τα ηχητικά αρχεία και δημιουργεί μια βάση δεδομένων από landmarks. Στη συνέχεια παίρνει τα queries από ένα ή περισσότερα αρχεία ήχου και τα αντιστοιχεί με αυτά που βρίσκονται στη βάση δεδομένων που δημιούργησε αρχικά. Αυτό βρίσκει εφαρμογή σε περιπτώσεις που θέλουμε να αφαιρέσουμε διπλότυπα κομμάτια από κάποια συλλογή. Το αποτύπωμα είναι ισχυρό απέναντι σε περιπτώσεις θορύβου ή διαφορετικής κωδικοποίησης και μπορεί να αντιστοιχήσει ακόμα και πολύ μικρά κομμάτια ήχου των 10 δευτερόλεπτων και κάτω. (<http://labrosa.ee.columbia.edu/matlab/audfprint/>)

OpenFP

Επιτρέπει στον χρήστη να δημιουργήσει αποτυπώματα από οποιοδήποτε μουσικό track χρησιμοποιώντας το βασικό πρόγραμμα το οποίο τα αντιστοιχεί με αυτά που προσφέρονται από τον server του OpenFP. Παρακάτω βλέπουμε τις διαδικασίες και τα βήματα που ακολουθούνται:



Εικόνα 6: Διαδικασία αναγνώρισης του openFP [78]

- i. Τα δεδομένα εξάγονται με τη χρήση ενός εργαλείου που υποστηρίζει την εκάστοτε κωδικοποίηση (π.χ. ffmpeg)
- ii. Μια συνάρτηση παράθυρου εφαρμόζεται στο ηχητικό σήμα.
- iii. Ακολουθεί μετασχηματισμός Fourier των δεδομένων για την δημιουργία ενός φάσματος ισχύος.
- iv. Η ισχύς του φάσματος μειώνεται στις σχετικές ζώνες ισχύος (Bark bands)
- v. Ο θόρυβος μειώνεται με τη χρήση ενός lowpass φίλτρου.
- vi. Οι τιμές κβαντίζονται σε δεκαδικές σημαίες (energy band active/inactive)
- vii. Μειώνονται τα δεδομένα εξόδου
(<http://open-fp.sourceforge.net/>)

SoundHound

Είναι μια μηχανή μουσικής αναζήτησης που επιτρέπει στους χρήστες να αναγνωρίζουν μουσική παίζοντας την, τραγουδώντας ή μουρμουρίζοντας μια μελωδία, ενώ ταυτόχρονα τους δίνει τη δυνατότητα να κάνουν αναζήτηση γράφοντας το όνομα του καλλιτέχνη, του τραγουδιού ή του σύνθετη. Το ταίριασμα επιτυγχάνεται μέσα από την τεχνολογία της εταιρίας ονόματι 'Sound2Sound' η οποία μπορεί να ταιριάξει οποιαδήποτε απόδοση του τραγουδιού όσο κακή και αν είναι, με το κανονικό τραγούδι. Μόλις συμβεί αυτό επιστρέφει τους στίχους, σύνδεσμο για το αντίστοιχο βίντεο στο YouTube, συνδέσμους για το iTunes και άλλα. Μια από τις λειτουργίες του η οποία ονομάζεται LiveLyrics μπορεί να προβάλει τους στίχους του τραγουδιού ταυτόχρονα με την αναπαραγωγή του. δίνει επίσης τη δυνατότητα στους χρήστες να αναπαράγουν τα μουσικά κομμάτια μέσα από τα κινητά τους τηλέφωνα ή οποιαδήποτε άλλη συσκευή χρησιμοποιούν. (<http://www.soundhound.com/>)

Άλλα συστήματα

MusicURI, μέρος του προγράμματος Mpeg-7 Audio DB .

Το jHears είναι ένα πλαίσιο ακουστικών αποτυπωμάτων σχεδιασμένο με βάση το FutureProof FingerPrint του Geoff Schmid .

Το AudioScout είναι βασισμένο στην βιβλιοθήκη ακουστικών αποτυπωμάτων rHash και χρησιμοποιεί τον αλγόριθμο "Philips Robust Hashing".

Philips. Ένας από τους πρώτους επεκτάσιμους αλγορίθμους για το ακουστικό αποτύπωμα.

Audible Magic. Λειτουργεί σε συνεργασία με την Philips.

AudioID. Σχεδιάστηκε για Digital Media Technology και παρέχεται από το muffin GmbH.

Rovi Media Recognition Service

USC, σχεδιασμένο για ψηφιακή επεξεργασία σήματος.

Κεφάλαιο 3

1. Echo nest - Echoprint

1.1.1. Echo Nest

Είναι μια εταιρία με τη μορφή μουσικής πλατφόρμας η οποία παρέχει μουσικές υπηρεσίες σε προγραμματιστές και άλλες εταιρίες που επεξεργάζονται δεδομένα. Με απώτερο σκοπό τη δημιουργία μιας βάσης δεδομένων αποτελούμενης από τραγούδια και καλλιτέχνες η Echo Nest συνδυάζει τεχνικές μηχανικής μάθησης και γλωσσικής επεξεργασίας στην τεχνολογία της. Διαθέτει έναν μηχανισμό μουσικής σύστασης που αναγνωρίζει τονικότητα, τέμπο και άλλες ομοιότητες μεταξύ τραγουδιών και συνδυάζει αυτά τα δεδομένα με μια πολιτιστική κατανόηση των τραγουδιών και των καλλιτεχνών που έχει προκύψει από διάφορα blogs, μουσικές αξιολογήσεις και άλλες εξωτερικές πηγές. Η Echo Nest διαθέτει μια βάση δεδομένων που περιέχει περίπου 30 εκατομμύρια τραγούδια και πάνω σε αυτή πατάνε όλες οι εφαρμογές που δημιουργούνται από τους προγραμματιστές της. Η βάση δεδομένων συγκεντρώνει τα δεδομένα της με τη χρήση web crawling, data mining και τεχνικές ψηφιακής επεξεργασίας σήματος και στη συνέχεια τα δεδομένα τροφοδοτούν υπηρεσίες μουσικής σύστασης, δημιουργίας λιστών αναπαραγωγής, ακουστικής ανάλυσης, μουσικής αναγνώρισης και τροφοδοσίας δεδομένων. Η εταιρεία έχει δημιουργήσει ένα API για ελεύθερη χρήση από τους προγραμματιστές και τους παροτρύνει να δημιουργήσουν μέσα από αυτό τις δίκες τους ανεξάρτητες μουσικές εφαρμογές, ενώ ταυτόχρονα προσφέρει περίπου ένα εκατομμύριο τραγούδια από τη βάση δεδομένων της για ερευνητικούς σκοπούς. Τέλος η Echo Nest συνεργάζεται και βοηθά στη δημιουργία υπηρεσιών με εταιρίες και τηλεοπτικά κανάλια όπως είναι τα MTV, island def jam, BBC, MOG, Warner Music Video, eMusic, Rdio, Clear Channel, VEVO, Nokia και Thumbplay ενώ πρόσφατα εξαγοράστηκε από την Spotify.

1.2. Εμπορική και ακαδημαϊκή χρήση

Η εμπορική πλευρά της Echo Nest έχει να κάνει με την δημιουργία εφαρμογών μέσα από τις συνεργασίες της με άλλες εταιρίες και απλούς χρήστες και προγραμματιστές. Μέχρι στιγμής κυκλοφορούν περίπου 150 εφαρμογές δημιουργημένες με τα εργαλεία που έχει προσφέρει η εταιρία και μπορεί ο οποιοσδήποτε να τις χρησιμοποιήσει ελεύθερα από αυτό το site <http://the.echonest.com/showcase/>. Το MTV σε συνεργασία με την εταιρία δημιούργησε το Music Meter, το οποίο ενημερώνει κάθε 15 λεπτά σχετικά με το ποια συγκροτήματα είναι πιο πολυσυζητημένα στο Διαδίκτυο.

Στον ακαδημαϊκό τομέα όπως έχει αναφερθεί ήδη, η Echo Nest έχει προσφέρει πρόσβαση σε περίπου ένα εκατομμύριο τραγούδια από τη βάση δεδομένων της για ερευνητικούς σκοπούς και ουσιαστικά βοηθά τους ερευνητές στον τομέα του MIR να προχωρήσουν στην έρευνα τους και να επεκτείνουν τον συγκεκριμένο επιστημονικό τομέα. Βοηθά δηλαδή σε περιπτώσεις ανάπτυξης εφαρμογών που μπορούν να ξεχωρίσουν τον ήχο της κιθάρας από αυτόν των ντραμς ή της ανθρώπινης φωνής, εφαρμογών που ακούνε ένα τραγούδι και περιγράφουν τον ρόλο του κάθε οργάνου ξεχωριστά ακόμα και εφαρμογών που μπορούν να πουν σε ποια δεκαετία

δημιουργήθηκε ένα συγκεκριμένο τραγούδι μέσα από στοιχεία όπως η παραγωγή του, αν είναι ηχογραφημένο σε mono ή stereo κλπ.

1.3. Υπηρεσίες

Μέσα από την πλατφόρμα της Echo Nest ο χρήστης έχει πρόσβαση σε υπηρεσίες όπως :

Μουσική αναζήτηση και εξατομίκευση όπου μέσα από playlists μπορεί να ανακαλύψει νέα είδη μουσικής και καλλιτέχνες. Η Echo Nest δημιούργησε το data-driven playlisting για να τροφοδοτήσει λίστες που είναι ικανές να κάνουν προβλέψεις και γίνονται «εξυπνότερες» με τον χρόνο καθώς συνδυάζουν την μηχανική μάθηση με τις ανθρώπινες ικανότητες. Οι λίστες αυτές έχουν τη δυνατότητα να γίνουν πιο προσωπικές συμφώνα με τις ενέργειες του χρήστη, δηλαδή ποια κομμάτια προσθέτει στα αγαπημένα, ποια προτείνει στους άλλους και ποια προσπέρνα. Αυτό βοηθά τον ίδιο τον χρήστη στην αναζήτηση του καλύτερου αποτελέσματος για εκείνον και στη δημιουργία ενός δικού του μουσικού προφίλ.

Dynamic Music Data πρόκειται για μια ατελείωτη ροή από δεδομένα σχετικά με νέα, εικόνες, posts, συζητήσεις για μουσική και καλλιτέχνες. Αυτό βοηθά τους συνεργάτες της εταιρίας να δείχνουν στο κοινό τι συμβαίνει στον κόσμο της μουσικής την κάθε στιγμή. Επίσης βοηθά τους προγραμματιστές να δημιουργήσουν μουσικά sites και εφαρμογές που να καταλαβαίνουν τα τραγούδια, τους καλλιτέχνες, τα είδη και τα άλμπουμ στην χώρα τους.

Music Audience Understanding, χωρίζει τους φίλους της μουσικής σε τμήματα βασισμένα στα προσωπικά τους προφίλ και στις αγαπημένες τους προτιμήσεις. Η Echo Nest έχει καταφέρει να δημιουργήσει ένα μοντέλο πρόβλεψης που καταλαβαίνει τα χαρακτηριστικά του κοινού ενώ ταυτόχρονα σέβεται την ιδιωτικότητα τους. Τα τμήματα τα οποία στοχεύει αφορούν την ηλικία, το φύλο καθώς και το lifestyle του κάθε χρήστη ξεχωριστά. Μπορεί να δημιουργήσει τμήματα κατά παραγγελία για μουσικές υπηρεσίες, διαφημιστικά δίκτυα και διαφημιστές απλά αναγνωρίζοντας τα μουσικά χαρακτηριστικά του κοινού.

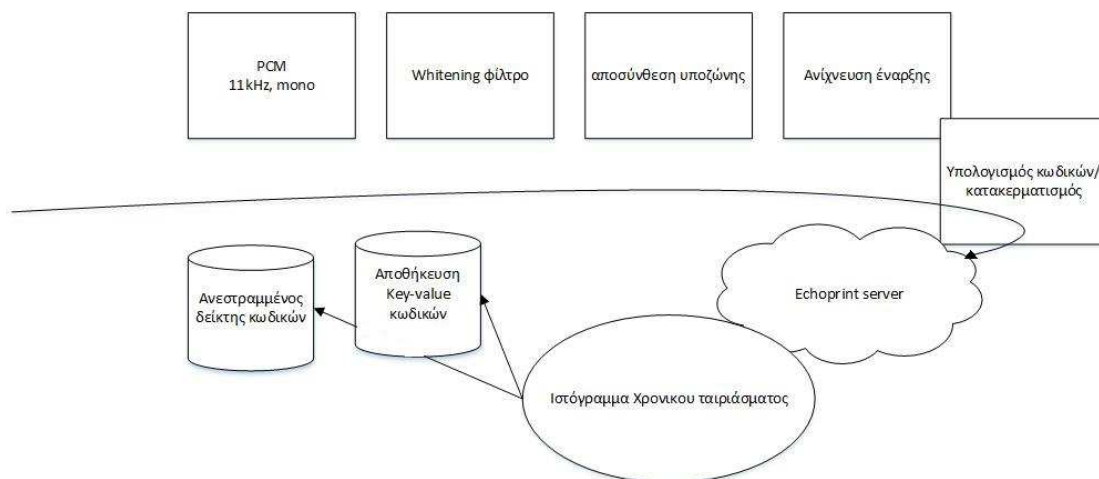
Ακουστικό αποτύπωμα η Echo Nest διαθέτει την τεχνολογία της για αναγνώριση ακουστικών αποτυπωμάτων δωρεάν και ελεύθερα προς το ευρύ κοινό καθώς επίσης και για εμπορικούς σκοπούς μέσα από ένα μεγάλο εύρος εφαρμογών όπως σκανάρισμα και ταίριασμα για διάφορες συσκευές, καθάρισμα μεταδεδομένων, βασική ταυτοποίηση τραγουδιών και άλλα.

Interactive Music & Remix Applications δίνει τη δυνατότητα στους μουσικούς και τους προγραμματιστές να πειράξουν και να φτιάξουν από την αρχή με διαφορετικό τρόπο οποιοδήποτε μουσικό κομμάτι ή video αυτόματα, καθώς και να δημιουργήσουν ανάλογες εφαρμογές μέσα από την ανάλυση δεδομένων, και την τεχνολογία για μετατροπή μουσικής της Echo Nest.

1.4. Τι είναι το echoprint

Η Echo Nest σε συνεργασία με την Musicbrainz δημιούργησε το Echoprint το οποίο πρόκειται για μια ελεύθερη εφαρμογή ταυτοποίησης που δίνει την δυνατότητα στον καθένα να κτίσει ένα μουσικό αποτύπωμα στις εφαρμογές του.

Το Echoprint έχει τη δυνατότητα να ακούσει έναν ήχο από το κινητό τηλέφωνο ή τον υπολογιστή και να αναγνωρίσει ποιο τραγούδι είναι, με τέτοια ταχύτητα και ακρίβεια που μπορεί να αναγνωρίσει ακόμα και τις πιο παραμορφωμένες από θόρυβο ηχογραφήσεις όπως για παράδειγμα από το μικρόφωνο ενός κινητού τηλεφώνου. Αυτό συμβαίνει γιατί ο αλγόριθμος του είναι έτσι σχεδιασμένος ώστε να είναι ισχυρός απέναντι σε θορυβώδη ερωτήματα και ο server του μπορεί να υποστηρίξει πάνω από 50 παράλληλα ερωτήματα το δευτερόλεπτο. Το Echoprint αποτελείται από 3 μέρη: την γεννήτρια, η οποία κωδικοποιεί τον ήχο, τον server, ο οποίος αποθηκεύει και βρίσκει κώδικες, και τα δεδομένα τα οποία έρχονται από συνεργάτες και απλούς χρήστες του Echoprint. Η γεννήτρια υπολογίζει ζευγάρια χρόνου και κατακερματισμού από ένα ηχητικό σήμα για να ανακαλύψει τυχόν θόρυβο και τροποποιήσεις. Αρχίζοντας με ένα mono σήμα των 11kHz δημιουργείται ένα whitening filter, και προχωρεί σε έναν διαχωρισμό 8 ζωνών σε μικρότερες. Αυτός ο διαχωρισμός αναζητεί ενάρξεις (onsets) και αφού τα ανακαλύψει τα κατακερματίζει σε διαστήματα των 20 bits και τα αποθηκεύει μαζί με τον χρόνο που δημιουργήθηκαν.



Εικόνα 7: Στάδια λειτουργίας του echoprint [41]

Ο κώδικας του server τοποθετεί κάθε έναρξη σε ένα ανεστραμμένο ευρετήριο, αποθηκεύοντας το τότε το onset του κάθε κομματιού δημιουργήθηκε σε μια μεγάλη λίστα για γρήγορη αναζήτηση. Επίσης αποθηκεύει το κωδικοποιημένο υλικό κάθε τραγουδιού κατά τη διάρκεια της αντιστοίχισης. Το Ερώτημαing είναι μια αναζήτηση για όλους τους κώδικες των ερωτημάτων στο ανεστραμμένο ευρετήριο και το αποτέλεσμα που επιστρέφεται είναι ο αριθμός των υπερθέσεων των ενάρξεων των ερωτημάτων ανάμεσα σε ένα ερώτημα και κάθε τραγούδι.

1.4.1. Περιγραφή λειτουργίας

Ο αλγόριθμος του Echoprint δέχεται τα ηχητικά σήματα, τα μετατρέπει σε mono και μειώνει τη συχνότητα δειγματοληψίας τους στα 11025 Hz. Προκειμένου να αποφύγει αναπάντεχους θορύβους που μπορεί να θεωρηθούν λανθασμένα ως μουσικές ενάρξεις, εφαρμόζει ένα whitened filter στο σήμα εισόδου. Αυτό το φίλτρο μετατρέπει το κάθε δείγμα που είναι έτοιμο να αξιολογηθεί σε μια ομαλοποιημένη τιμή από 40 προηγούμενα δείγματα. Αυτή η διαδικασία μειώνει το πλάτος των ξαφνικών κορυφών στο σήμα. Μόλις ολοκληρωθούν αυτές οι δυο διαδικασίες ακολουθεί η μετατροπή του ήχου στο πεδίο των συχνοτήτων. Το Echoprint

χρησιμοποιεί μια τράπεζα 128 συνιμιτονοειδων ζωνοπερατών φίλτρων για τον συγκεκριμένο μετασχηματισμό, η οποία, κινείται πάνω από το σήμα με αναπηδήματα της τάξεως των 32 δειγμάτων. Οι συχνοτικές ζώνες που δίνονται ως αποτέλεσμα ομαδοποιούνται σε 8 ισαπέχοντα κουτιά μέσα από το άθροισμα των απολύτων διαφορών των παρακείμενων ζωνών. Αυτά τα 8 κουτιά κατανέμονται στην πορεία από τα 0 Hz έως τα 5512.5 Hz. Οι κατακερματισμοί του Echoprint υπολογίζονται μέσα από τη διαφορά στον χρόνο μεταξύ των μουσικών onsets σε κάθε ζώνη. Το πρώτο βήμα της διαδικασίας είναι η εύρεση αυτών των onsets μέσα στο ηχητικό σήμα από κάθε συχνοτική ζώνη ξεχωριστά. Το πλάτος της κάθε ζώνης μετριέται μέσα από ένα envelope follower και μόλις το πλάτος φτάσει στο κατώφλι καταχωρείται και μια έναρξη. Μόλις εντοπιστεί μια έναρξη πρέπει να περάσουν 128 δείγματα μέχρι την επόμενη. Το πλάτος της κάθε έναρξης πολλαπλασιάζεται με μια εκθετικά φθίνουσα καμπύλη ώστε να υπολογιστεί νέα τιμή κατωφλίου, η οποία είναι η τιμή που πρέπει να υπερβούν οι μελλοντικές ενάρξεις προκειμένου να υπολογιστούν. Ο στόχος του Echoprint είναι να παράγει ένα onset για κάθε συχνοτική ζώνη ανά ένα δευτερόλεπτο. Στην περίπτωση που παράγονται συχνότερα από αυτόν τον ρυθμό ο πολλαπλασιαστής αυξάνεται με αποτέλεσμα να έχουμε ένα μεγαλύτερο κατώφλι. Αν ο ρυθμός της παραγωγής τους μειωθεί δραστικά ο πολλαπλασιαστής πρέπει να μειωθεί ως αντιστάθμιση. Για να έχουμε υπολογισμό των ενάρξεων σε αριθμητικές αξίες ο αλγόριθμος λαμβάνει υπόψη του τον χρόνο του κάθε onset και τον χρόνο όσων ακολουθούν. Στη συνέχεια μια αξία κατακερματισμού δημιουργείται για τον χρόνο μεταξύ των ζευγαριών των ενάρξεων και τις ζώνες στις οποίες βρίσκονται. Οι δυο τιμές κατακερματισμού και ο δείκτης της ζώνης αποθηκεύονται σε έναν αριθμό των 40 bit ο οποίος στη συνέχεια μειώνεται σε έναν ακέραιο των 32 bit. Η κάθε έναρξη μαζί με τα σετ των επόμενων παράγει έξι κατακερματισμούς. Ουσιαστικά με το να βάζει σε ζεύγη τις ενάρξεις και τους διάδοχους ο αλγόριθμος γίνεται πιο ισχυρός απέναντι σε περιπτώσεις που αποτύχει να τα αναγνωρίσει. Οι κατακερματισμοί συνδυάζονται με τον χρόνο στον οποίο εμφανιστήκαν οι ενάρξεις ενώ αντίστοιχα τα ζευγάρια του κατακερματισμού που αφορούν την λήξη (offset) μιας ηχογράφησης χωρίζονται σε έναν αριθμό από υπό- ηχογραφήσεις των 60 δευτερόλεπτων το κάθε ένα, που υπερκαλύπτουν προηγούμενες υπό- ηχογραφήσεις κατά 30 δευτερόλεπτα. Αυτό συμβαίνει επειδή η συνδυαστική συνιστώσα του συστήματος βαθμολογεί τις ηχογραφήσεις ανάλογα το πόσες φορές ένας κατακερματισμός στην ηχογράφηση αντιστοιχείται με έναν κατά το ερώτημα. Αν οι κατακερματισμοί δεν διαχωρίζονταν τότε οι ηχογραφήσεις με μεγάλη διάρκεια θα είχαν ένα αθέμιτο πλεονέκτημα κατά την αναζήτηση, αφού οι ηχογραφήσεις με επαναλαμβανόμενο περιεχόμενο παράγουν τις ίδιες τιμές κατακερματισμού σε πολλά σημεία της ηχογράφησης. [32]

Οι τιμές του κατακερματισμού αποθηκεύονται σε ένα ανεστραμμένο ευρετήριο, όπου η κάθε τιμή περιέχει αναφορές σχετικά με της υπό- ηχογραφήσεις και την διάρκεια της ηχογράφησης που συμβαίνει ο κατακερματισμός. Ο server του Echoprint χρησιμοποιεί τον Apache Solr9, μια μηχανή η οποία κάνει ταχύτατες αναζητήσεις κείμενων και με τη βοήθεια της αποθηκεύει τον δείκτη του κατακερματισμού. Τα ολοκληρωμένα πακέτα των τιμών του κατακερματισμού που αφορούν τα offset της κάθε ηχογράφησης αποθηκεύονται σε διαφορετικές βάσεις δεδομένων για την διαδικασία της αναζήτησης.[41]

Η διαδικασία της αναζήτησης χωρίζεται σε 2 βήματα. Κατά το πρώτο βήμα ξεχωρίζονται οι τιμές του χρόνου και γίνεται αναζήτηση στον ανεστραμμένο δείκτη για τις υπό- ηχογραφήσεις των 60 δευτερόλεπτων που περιέχουν τις τιμές του κατακερματισμού που εμφανιστήκαν κατά το ερώτημα. Αυτές οι υπό- ηχογραφήσεις στη συνέχεια διευθετούνται ανάλογα το πόσες φορές ο κατακερματισμός μιας ηχογράφησης ταιριάζει με τον κατακερματισμό ενός ερωτήματος. Οι 15 πιο συχνά συνδυασμένες είναι και αυτές που θα επιστραφούν. Στην περίπτωση που επιστραφεί παραπάνω από μια υπό- ηχογράφηση της ίδιας ηχογράφησης τότε μόνο μια από αυτές θα παραμείνει και οι υπόλοιπες θα απορριφθούν. Το τελικό αποτέλεσμα υπολογίζεται με το να ταιριάξουν οι κατακερματισμοί του με αυτούς της ηχογράφησης. Για να συμβεί αυτό πρέπει να υπολογιστεί η διαφορά του χρόνου ανάμεσα στον χρόνο της έναρξης κατά το ερώτημα και τον χρόνο της έναρξης στην ηχογράφηση για κάθε κατακερματισμό μέσα σε αυτή. Το άθροισμα των φορών που εμφανίζεται η διαφορά του χρόνου είναι και αυτό που κρατείται. Αν η ηχογράφηση με τα περισσότερα ταιριάσματα κατακερματισμών έχει δυο φορές περισσότερα ταιριάσματα από την επομένη ηχογράφηση τότε επιστρέφεται ως αποτέλεσμα. Αλλιώς δεν επιστρέφεται τίποτα. [32] [41]

1.4.2. Λειτουργίες του echoprint

Ένα σύστημα ακουστικών αποτυπωμάτων όπως το Echoprint αναγνωρίζει τη μουσική με βάση το πώς αυτή ακούγεται και μόνο, χωρίς να το απασχολούν το bit rate, η κωδικοποίηση ή η συμπίεση που έχει υποστεί για να δημιουργηθεί το αρχείο. Ακόμα και αν έχει να αντιμετωπίσει πρόχειρα μεταδεδομένα σε ένα μουσικό αρχείο αν η μουσική ακούγεται ίδια τότε το σύστημα θα την αναγνωρίσει.

Πρακτικά, η εφαρμογή μπορεί να συλλέξει ήχο από ένα αρχείο, να χρησιμοποιήσει τον δοσμένο κώδικα ώστε να δημιουργήσει ένα αποτύπωμα, και να το αναζητήσει μέσα από τον server του echoprint. Ο server θα επιστρέψει τα αντιστοιχισμένα αποτελέσματα σε μορφοποίηση JSON, αλλιώς αν δεν υπάρχει αντιστοίχιση η εφαρμογή υποβάλει της πληροφορίες στη βάση δεδομένων του server μαζί με τα όποια μεταδεδομένα του τραγουδιού έχουν αποκτηθεί και από άλλα μέσα.

Η πιο βασική χρήση του echoprint είναι ότι μπορεί να χρησιμοποιηθεί από τον οποιονδήποτε προγραμματιστή για να κτίσει τις δικές του εφαρμογές μουσικής αναγνώρισης χωρίς κανένα κόστος και χωρίς να χρειάζεται να έχουν κάποιον δικό τους server ή να συγκεντρώσουν δικά τους δεδομένα. Οι εφαρμογές που θα μπορούσαν να δημιουργηθούν θα μπορούσαν να είναι μια μουσική εφαρμογή αναγνώρισης τραγουδιών, ένα μουσικό παιχνίδι που θα αναγνωρίζει το τραγούδι που ακούγεται σε έναν χώρο και θα φορτώνει τα δεδομένα μέσα από κάποια εφαρμογή ή βιβλιοθήκη της Echo nest, μια εφαρμογή όπου μέσω τηλεφώνου θα αναγνωρίζεται μια κινηματογραφική ταινία απλά ακούγοντας το soundtrack.

Άλλες βασικές χρήσεις είναι:

1. η αναγνώριση τραγουδιών μέσα από τον σκληρό δίσκο του υπολογιστή ή μέσα από κάποιον server. Για παράδειγμα υπάρχει ένα αρχείο του οποίου δεν είναι γνωστό το όνομα ή το όνομα του καλλιτέχνη. Αφού έχει εγκατασταθεί το codegen στο σύστημα απλά χρειάζεται η εντολή «./echoprint-codegen ~/όνομα τραγουδιου.mp3 10 30» και άμεσα θα δοθούν τα στοιχεία για το συγκεκριμένο αρχείο.

2. **Ξεκαθάρισμα διπλών αρχείων και αντιγράφων.** Για να γίνει αυτό αρκεί να φορτωθεί ο κατάλογος με τα αρχεία στον server του echoprint και στην πορεία πραγματοποιείται η αναζήτηση για το ξεκαθάρισμα των αρχείων.

3. **Αναγνώριση τραγουδιών μέσω μιας συσκευής που χρησιμοποιεί λογισμικό iOS.** Κάτι το οποίο βρίσκεται ακόμα σε δοκιμαστικό στάδιο παρόλα αυτά το echoprint δίνει τις βάσεις για την δημιουργία κώδικα σε μια συσκευή που θα δέχεται είσοδο από ένα μικρόφωνο ή αρχεία από τη προσωπική συλλογή στο iPod ενός χρήστη.

4. **Monetizing User-Generated Content** όταν κάποιος ανεβάσει ένα video, ένα slideshow, ή οτιδήποτε άλλο που περιέχει ένα γνωστό τραγούδι ως soundtrack, το Echoprint συνεργάζεται με εταιρίες online video οι οποίες ερευνούν το αν το συγκεκριμένο τραγούδι πληροί τα πνευματικά δικαιώματα του δημιουργού κατά τη συγκεκριμένη χρήση.

5. **Ακουστική αναγνώριση.** Οι δυνατότητες του Echoprint για αναγνώριση μέσω του αέρα σε πραγματικό χρόνο είναι πολύ περιορισμένες τη δεδομένη στιγμή. Ωστόσο το Echoprint έχει τη δυνατότητα για την συγκεκριμένη εφαρμογή και προσπαθεί να την αναπτύξει μέσω των προγραμματιστών του αλλά και των προγραμματιστών που χρησιμοποιούν τα εργαλεία του.

Το βασικό εργαλείο για να επιτευχθέν όλα αυτά είναι το echoprint codegen. Προσφέρεται ελεύθερα μέσα από το Github και έχει δυο τρόπους λειτουργίας :

1. Η βιβλιοθήκη κωδίκων (libcodegen) η οποία είναι προορισμένη να συνδέεται με έναν κώδικα που παίρνει ένα buffer PCM δεδομένων και δίνει σαν έξοδο μια σειρά από κώδικες.
2. Το αυτόνομο διάδικο codegen, το οποίο δέχεται ονόματα αρχείων σαν είσοδο και τα τρέχει σε ένα multithread περιβάλλον εργασίας.

Το Codegen σκανάρει τον ήχο περίπου στα 250x σε πραγματικό χρόνο ανάλογα τον επεξεργαστή, αφού πραγματοποιήσει αποκωδικοποίηση και αναδειγματοληψία στα 11025 Hz. Αυτό σημαίνει ότι ένα τραγούδι μπορεί να σκαναριστεί ολόκληρο σε λιγότερο από 0.5s σε έναν απλό υπολογιστή, και ότι το δείγμα που λαμβάνει για τη διαδικασία του ερωτήματος (30s) μπορεί να σκαναριστεί σε λιγότερο από 0.04s. Ανάλογα το λογισμικό του κάθε χρήστη το codegen εγκαθίσταται μέσα από το τερματικό για χρήστες mac και Linux και μέσα από το command line για χρήστες windows.

Το Echoprint μπορεί να χρησιμοποιηθεί ταυτόχρονα με άλλες εφαρμογές της Echo Nest όπως το **Rosetta Stone** και το **Taste Profiles** και επιτρέπει στους προγραμματιστές να κτίσουν εφαρμογές που μπορούν να μετατρέψουν προσωπικούς μουσικούς καταλόγους σε κάθε μουσική υπηρεσία. Για παράδειγμα το Echoprint μπορεί να πάρει μια οποιαδήποτε πηγή ήχου και να αναγνωρίσει τη σελίδα του καλλιτέχνη στο Facebook ή να φτιάξει μια playlist στο 7digital.

2. Online Music Services and APIs

Η επανάσταση που έχει φέρει στη μουσική το διαδίκτυο, δεν θα μπορούσε να μην επηρεάσει και τον τρόπο με τον οποίο το κοινό πλέον επιλεγεί να ακούσει μουσική. Ο πιο κοινός τρόπος ακρόασης στη σημερινή εποχή είναι μέσα από online streaming εφαρμογές οι οποίες μπορούν να προσφέρουν στο κοινό τις υπηρεσίες τους χωρίς να χρειάζεται να κατεβάσουν τραγούδια ή να οργανώσουν ολόκληρες λίστες με αυτά. Αυτές οι υπηρεσίες λειτουργούν σαν online βάσεις δεδομένων όπου μέσα από μια αναζήτηση επιστρέφουν στον χρήστη το αντικείμενο της αναζήτησης του ενώ ταυτόχρονα του προτείνουν νέα ακούσματα βασισμένα στις προτιμήσεις του. Θα λέγαμε δηλαδή ότι χρησιμοποιούν τεχνικές MIR και audio fingerprinting στην λειτουργία τους και επιτρέπουν στους χρήστες να αναζητούν τραγούδια, καλλιτέχνες ή και ολόκληρες playlists βασισμένες στο είδος και τη διάθεση. Κάποιες από αυτές όπως το youtube χρησιμοποιούν εξελιγμένους audio fingerprinting αλγορίθμους ώστε να προστατέψουν τα πνευματικά δικαιώματα των καλλιτεχνών και των δημιουργών.

2.1. Κατηγορίες

Οι ψηφιακές μουσικές υπηρεσίες χωρίζονται σε κατηγορίες ανάλογα με τις ανάγκες του κοινού και του κάθε χρήστη ξεχωριστά. Μερικές από αυτές είναι :

Download – Οι υπηρεσίες για Download επιτρέπουν την αγορά δίσκων, βίντεο και οτιδήποτε άλλου περιέχει μουσικό περιεχόμενο μέσω υπολογιστή ή κινητού αρκεί να είναι συνδεδεμένα στο ίντερνετ. Μερικές από αυτές τις υπηρεσίες είναι οι Apple iTunes, 7Digital, Amazon, HMV, Napster, Fairshare Music, Play και Tesco.

Κινητής τηλεφωνίας – Αυτές οι υπηρεσίες επιτρέπουν την αποθήκευση και την αναπαραγωγή μουσικών κομματιών, albums, videos, και ήχων κλήσης στο κινητό τηλέφωνο. Περιλαμβάνει ακόμα και μουσικές εφαρμογές, οι οποίες λειτουργούν σε συνδυασμό με συμπληρωματικές υπηρεσίες στον υπολογιστή. Οι υπηρεσίες κινητής συνεργάζονται με τους περισσότερους φορείς εκμετάλλευσης κινητών υπηρεσιών όπως Orange, T-Mobile, cosmote στην Ελλάδα και Vodafone, καθώς και με κάποιες πιο εξειδικευμένες εταιρίες όπως οι Jamba και Shazam.

Συνδρομής – Επιτρέπουν την παρακολούθηση ή το κατέβασμα ενός συγκεκριμένου αριθμού τραγουδιών μέσω πληρωμής ενός συγκεκριμένου πόσου κάθε μήνα. Τέτοιου είδους υπηρεσίες είναι οι Spotify, eMusic, Deezer και Rara.

Ad-Supported – Παρέχουν δωρεάν μουσική με αντάλλαγμα την προβολή διαφημίσεων. Παράδειγμα τέτοιων υπηρεσιών είναι η ελεύθερη υπηρεσία του Spotify, We7, last.FM και το YouTube.

Cloud – Οι συγκεκριμένες υπηρεσίες επιτρέπουν την αποθήκευση μιας μουσικής συλλογής και πρόσβαση σε αυτή από οποιαδήποτε συσκευή, είτε είναι υπολογιστής, κινητό, tablet ή MP3 player. Τέτοιες υπηρεσίες παρέχονται από το iTunes, Google και Amazon οι οποίες προσφέρουν τον δικό τους αποθηκευτικό χώρο.

Social Media – Επιτρέπουν την πρόταση μουσικής και playlists σε άλλους ανθρώπους, μερικές φορές προσφέροντας και οικονομικό κέρδος. Κάποιες από αυτές είναι οι Blackberry Music, Rara, Spotify και Last FM.

2.2. Λειτουργία και παραδείγματα

Η χρήση αυτών των υπηρεσιών βασίζεται στην δημιουργία ενός profile αρχικά και στη συνέχεια μιας μουσικής βιβλιοθήκης βασισμένης στις προτιμήσεις του κάθε χρήστη. Η υπηρεσία στη συνέχεια προωθεί τη μουσική μέσω του διαδικτύου στον ίδιο τον χρήστη. Κάποιες υπηρεσίες όπως το Pandora Radio, επιτρέπουν την δημιουργία σταθμών βασισμένων σε κάποιον συγκεκριμένο καλλιτέχνη ή τραγούδι. Ωστόσο δεν είναι σίγουρο ότι ο δημιουργός τους θα ακούσει αυτόν τον σταθμό κάθε φορά που θα χρησιμοποιεί αυτή την υπηρεσία καθώς το Pandora Radio θα επιλέγει τραγούδια ανάλογα με τις προτιμήσεις του. Όσο κάποιος ακούει μουσική μέσα από το Pandora radio και εγκρίνει ή απορρίπτει τραγούδια που παίζουν στους διάφορους σταθμούς του, αυτό προσαρμόζει τις playlist ανάλογα με το προσωπικό γούστο του ακροατή. Το Pandora radio δεν είναι το μοναδικό που χρησιμοποιεί μια σειρά από αλγόριθμους ώστε να αντιστοιχήσει τις προτιμήσεις του κάθε χρήστη με τις διάφορες υπηρεσίες που προσφέρει. Υπηρεσίες όπως το **Spotify** επιτρέπουν την αναζήτηση συγκεκριμένων τραγουδιών και τα προσθέτουν σε playlists. Οποιαδήποτε στιγμή ο χρήστης συνδεθεί με τον λογαριασμό του μπορεί να ακούσει τις playlists του, να αλλάξει τη σειρά αναπαραγωγής και να προσθέσει ή να διαγράψει μουσική. Όπως αναφέρθηκε και πιο πάνω η Spotify πλέον έχει ξεκινήσει άμεση συνεργασία με την **Echo Nest**. Το API του Spotify προμηθεύει τους προγραμματιστές με οτιδήποτε χρειάζονται για την αναπαραγωγή μουσικής και την διευκόλυνση της συνεργασίας με τον ακροατή, ενώ το Echo Nest προμηθεύει με όλα τα απαραίτητα δεδομένα για την μουσική. Το **WhoSampled** (<http://www.whosampled.com/>) είναι μια ιστοσελίδα και βάση δεδομένων με πληροφορίες για μουσική βασισμένη σε samples. Συγκρίνει τα αυθεντικά τραγούδια με τις διασκευές του ή με εκείνα που έχουν δανειστεί samples, μέσω συλλογής πληροφοριών από τους χρήστες της ιστοσελίδας. Η βάση δεδομένων είναι οργανωμένη ανά καλλιτέχνη και καταλόγους δειγμάτων μέσω ενσωμάτωσης τρίτων κατόχων των κομματιών. Το **Last.FM** (<http://www.last.fm/>) είναι μια μουσική ιστοσελίδα, όπου ο χρήστης μπορεί να κτίσει μουσικά προφίλ χρησιμοποιώντας μια σειρά από μεθόδους: ακούγοντας την προσωπική μουσική συλλογή του σε μια εφαρμογή αναπαραγωγής σε έναν υπολογιστή ή iPod που χρησιμοποιεί το Audioscrobbler plugin, ή ακούγοντας το internet radio του Last.FM. Όλα τα τραγούδια που έπαιξαν προστίθεται σε ένα αρχείο καταγραφής από το οποίο υπολογίζονται τα ιστογράμματα των μουσικών προτάσεων για τους κορυφαίους καλλιτέχνες, τραγούδια κλπ. Αυτή η τεχνική ονομάζεται **scrobbling**. Οι προτάσεις υπολογίζονται μέσα από έναν collaborative filtering αλγόριθμο που όταν ο χρήστης σαρώνει και ακούει προεπισκοπήσεις από μια λίστα με καλλιτέχνες που δεν περιλαμβάνονται στο δικό του προφίλ, αυτές εμφανίζονται στα προφίλ άλλων με παρόμοια μουσικά γούστα. Περιέχει ακόμα μια λίστα με μουσική που έχει προταθεί απευθείας στο χρήστη και σε ομάδες που ανήκει ο χρήστης, περιοδικά γραμμένα από χρήστες σχετικά με καλλιτέχνες που ο συγκεκριμένος χρήστης ακούει και άλλους χρήστες που έχουν ακούσει την ίδια μουσική το ίδιο χρονικό διάστημα. Υπάρχει ακόμα ένα ραδιόφωνο μουσικής σύστασης το οποίο παίζει μουσική φιλτραρισμένη συγκεκριμένα από το τι έχει ακούσει ο χρήστης την τελευταία εβδομάδα. Η μουσική βιβλιοθήκη του Last.FM περιέχει πάνω από 12 εκατομμύρια ξεχωριστά τραγούδια. Το **BMAT** (<http://www.bmat.com/>) είναι μια υπηρεσία μουσικής σύστασης και ανάλυσης το οποίο προσφέρει 3 εφαρμογές : το *Vericast* που είναι μια παγκόσμια υπηρεσία αναγνώρισης μουσικής που παρακολουθεί εκατομμύρια τραγούδια και πάνω από 3000 ραδιοφωνικές και τηλεοπτικές εκπομπές σε περισσότερες από 60

χώρες παγκοσμίως. Παρέχει αναγνώριση σε πραγματικό χρόνο βασισμένο σε ένα ακουστικό αποτύπωμα που είναι ανθεκτικό απέναντι σε εναλλαγές του σήματος. Το *Ella* που είναι μια μηχανή ηχητικής ανάλυσης, αναζήτησης και σύστασης για media services ώστε να κατανοήσουν και να προσωποποιήσουν μουσική. Το Ella που παρέχει αντιληπτική συνεπή μουσική περιήγηση και ανακάλυψη μέσα από τα διάφορα πεδία γνώσης: πλαίσιο (τίτλος, καλλιτέχνης, ετικέτες, ημερομηνία κυκλοφορίας, τη χώρα, τη γλώσσα, την ημερομηνία κυκλοφορίας), περιεχόμενο (διάθεση, φωνή, τονικότητα, κλίμακα, beats ανά λεπτό) και δεδομένα χρηστών (ιστορικό αγοράς, συμπεριφορά ακροατή, συνήθειες λίστες αναπαραγωγής). Τέλος το *Score* που είναι ένα λογισμικό αξιολόγησης τραγουδιού που αξιολογεί a cappella εκτελέσεις βασισμένο στην μουσικότητα της φωνής και την έκφραση. Είναι το μοναδικό που υπάρχει στον τομέα της αυτόματης αξιολόγησης των φωνητικών ικανοτήτων ανεξαρτήτου είδους τραγουδιού ή γλώσσας. Το **Decibel** (<http://www.decibel.net/>) είναι ένα κατάστημα αγοράς μουσικών μεταδεδομένων. Περιέχει δίσκους, τραγούδια και καλλιτέχνες και προμηθεύει μια μηχανή αναζήτησης η οποία συνδέει όλα τα δεδομένα μαζί. χρησιμοποιεί τεχνολογία που προέρχεται από τις υπηρεσίες συλλογής πληροφοριών ώστε να καταστεί δυνατή η διασύνδεση των δεδομένων, αποθήκευση πληροφοριών για καλλιτέχνες και τη δουλειά τους αλλά και τις σχέσεις μεταξύ τους. Το **seevl** (<https://developer.seevl.fm/>) είναι μια υπηρεσία η οποία έχει ως σκοπό την ανακάλυψη νέων καλλιτεχνών και μουσικών μέσα από έναν αριθμό που ξεπερνά τους 40000 καλλιτέχνες. Αυτό όμως που το κάνει να διαφέρει από της άλλες υπηρεσίες είναι ότι όχι μόνο προτείνει καλλιτέχνες βασισμένους στις ήδη προτιμήσεις του χρήστη αλλά αιτιολογεί το γιατί τους προτείνει και χρησιμοποιεί μια επιλογή η οποία λέγεται artist connection. Δηλαδή για παράδειγμα μπορεί να προτείνει νέες μπάντες μέσα από τις προσωπικές συνδέσεις των καλλιτεχνών.

3. Musicbrainz και audio fingerprinting

3.1. Musicbrainz

Το MusicBrainz είναι μια σχεσιακή βάση δεδομένων που αφορά μουσικά μεταδεδομένα. Πέραν της συλλογής μεγάλων ποσοτήτων πληροφοριών πέρα από το παραδοσιακό μοντέλο του καλλιτέχνη / άλμπουμ / κομμάτι, το MusicBrainz έχει σκοπό να προμηθεύει με μουσικά μεταδεδομένα τις μουσικές «αποθήκες». Το MusicBrainz είναι μια αξιόπιστη πηγή για δωρεάν μουσικές πληροφορίες και μεταδεδομένα για ηχητικά αρχεία χωρίς να είναι πλήρως εμπορικό και απαιτητικό. Έχει τη μορφή του site στο οποίο οι χρήστες μπορούν να συγκρίνουν και να βελτιώσουν τα ηχητικά μεταδεδομένα ταιριάζοντας και συμβάλλοντας με πληροφορίες σε μια κοινόχρηστη βάση δεδομένων. Το MusicBrainz είναι δομημένο ως μια σχεσιακή βάση δεδομένων όπου το περιεχόμενο της καθώς και όλες οι πληροφορίες υποστήριξης διατίθενται δωρεάν μέσα από την ιστοσελίδα, όπως και οι εφαρμογές του. Η πρόσβαση και η αναζήτηση μέσα στη βάση δεδομένων μπορεί να γίνει με διάφορους τρόπους. Ένα κουτί αναζήτησης είναι διαθέσιμο σε όλες τις σελίδες του κεντρικού site, και επιτρέπει την αναζήτηση μέσα από λέξεις κλειδιά σε τομείς όπως καλλιτέχνες, κυκλοφορίες, τραγούδια και σχόλια. Μια εξειδικευμένη αναζήτηση σαρώνει όλους τους τομείς των μεταδεδομένων και επίσης δίνονται ψηφιακά έγγραφα που περιγράφουν λεπτομερώς τις δυνατότητες της μηχανής αναζήτησης.

3.2. Βάση δεδομένων

Η σχεσιακή βάση δεδομένων του MusicBrainz λειτουργεί με βάση τα albums και περιέχει λεπτομερή μεταδεδομένα σχετικά με καλλιτέχνες, κυκλοφορίες, τραγούδια, σχέσεις μεταξύ τους, τις αλλαγές που μπορεί να κάνουν οι χρήστες του MusicBrainz στα περιεχόμενα της βάσης δεδομένων. Το MusicBrainz είναι η μόνη μουσική βάση δεδομένων που μπορεί να αποθηκεύει όχι μόνο περιγραφικά, αλλά και διαρθρωτικά και διαχειριστικά μεταδεδομένα. Πολύ σημαντικό γεγονός καθότι βοηθά στην έκφραση των πολλών σχέσεων μεταξύ των μουσικών πόρων, καθώς και την παροχή αποτελεσματικών τρόπων διαχείρισης και την μεταξύ τους διασύνδεση. Το MusicBrainz αποθηκεύει επίσης ακουστικά αποτυπώματα (PUIDs) που μπορούν να χρησιμοποιηθούν για αναζήτηση πληροφοριών ακόμα και αν τα περιγραφικά μεταδεδομένα δεν είναι διαθέσιμα. Τέλος το MusicBrainz χρησιμοποιεί 30 διαφορετικές γλώσσες για να περιγράψει τον τρόπο με τον οποίο είναι δομημένη η βάση δεδομένων του.

Η βάση δεδομένων είναι οργανωμένη με τον εξής τρόπο :

Βασικά δεδομένα

Καλλιτέχνες

Όνομα, επίθετο, ψευδώνυμο, τύπος type, ημερομηνία έναρξης και τέλους, σχόλια, MBID

Release Groups

Τίτλος, καλλιτέχνης, είδος, σχόλια, MBID

Κυκλοφορίες

Τίτλος, καλλιτέχνης, είδος, κατάσταση, γλώσσα, ημερομηνία , χώρα ,ετικέτα, αριθμός τραγουδιών, barcode,φορέας, disc ID(s), ASIN, σχόλια, MBID

Φορείς

Μορφοποίηση, λίστα τραγουδιών

Ηχογραφήσεις

Τίτλος, καλλιτέχνης, διάρκεια, ISRC, PUIDs, σχέσεις, σχόλια, MBID

Εργασίες

Τίτλος, ISWC,σχέσεις , σχόλια, MBID

Ετικέτες- προσδιορισμός

όνομα, επίθετο ,ψευδώνυμο, χώρα, είδος, κωδικός, ημερομηνία έναρξης και τέλους, σχόλια, MBID

Σχέσεις και URLs

Οι σχέσεις είναι ο τρόπος για να συνδεθούν τα παραπάνω μεταξύ τους και επιτρέπουν στο MusicBrainz να συλλάβει τα περισσότερα από τα στοιχεία που περιέχονται στο εσωτερικό του CD.

CD Stubs

Τίτλος, καλλιτέχνης, barcode, disc ID, σχόλια.

Συμπληρωματικά στοιχεία

Τα συμπληρωματικά στοιχεία περιλαμβάνουν: σχόλια χρήστη, ετικέτες και βαθμολογίες, στατιστικά στοιχεία, κατάλογοι αναζήτησης, επεξεργασία ιστορικού, μη προσωπικά δεδομένα χρήστη. Η βάση δεδομένων του Musicbrainz δίνει τη δυνατότητα στους χρήστες να ανακαλύψουν τις δημοσιεύσεις των αγαπημένων τους καλλιτεχνών, ποια άλμπουμ λείπουν από την προσωπική τους συλλογή, να ενημερώνονται σχετικά με νέες κυκλοφορίες, να μαθαίνουν τους δημιουργούς ενός άλμπουμ ή τραγουδιού και να συνεισφέρουν με δεδομένα. Μια άλλη χρήση είναι η αναζήτηση για πληροφορίες μέσα από media players καθώς τα περισσότερα CDs δεν περιέχουν τα μεταδεδομένα της εγγραφής, ωστόσο το MusicBrainz μπορεί να χρησιμοποιήσει τα ψηφιακά χαρακτηριστικά ενός CD (αριθμό και διάρκεια τραγουδιών) ώστε να αναζητήσει τα μεταδεδομένα και να χρησιμοποιήσει τις πληροφορίες τους.

3.3. Picard

Το Picard επισημαίνει με ετικέτες mp3 αρχεία και επιτρέπει την αναγνώριση και την κατηγοριοποίηση της μουσικής με ετικέτες και ονόματα αρχείων. Μια πολύ χρήσιμη ιδιότητα του είναι ότι μπορεί να δημιουργήσει αποτυπώματα και εν συνεχεία να αναγνωρίσει ακόμα και τραγούδια τα οποία δεν έχουν επισημανθεί, όπως για παράδειγμα σε περιπτώσεις που κάποιος χρήστης δεν γνωρίζει τις πληροφορίες σχετικά με το άλμπουμ ή τον καλλιτέχνη. Για να υπάρξει αναγνώριση το πρόγραμμα δημιουργεί ένα ψηφιακό αποτύπωμα το οποίο αναλύεται και συγκρίνεται με την βάση δεδομένων του MusicBrainz. Για παράδειγμα κάποιος διαλέγει ένα άγνωστο ηχητικό αρχείο και πατεί το κουμπί Scan στο πρόγραμμα. Αν το αρχείο είχε το όνομα "Track.mp3" στον φάκελο "Unknown Artist", τότε αυτόματα αναγνωρίζεται και μετονομάζεται σε "Red Hot Chili Peppers\Californication\07 Easily.mp3". Το Picard μπορεί να πραγματοποιήσει αυτή την αναγνώριση ακόμα και απευθείας από ένα CD.

3.4. Musicbrainz API και audio fingerprinting

Το Musicbrainz έχει χρησιμοποιήσει έναν αριθμό από συστήματα ακουστικού αποτυπώματος τα οποία λειτουργούν περίπου με τον ίδιο τρόπο. Πρώτα ο ήχος χρησιμοποιείται για την δημιουργία του αποτυπώματος, το οποίο στην πορεία υποβάλλεται σε έναν third-party server. Ο server αναλύει το αποτύπωμα και το συγκρίνει με άλλα. Αφού ολοκληρωθεί αυτό αποφασίζει το αν είναι αρκετά διαφορετικό από γνωστά αποτυπώματα ώστε να θεωρηθεί νέο ID. Με την ολοκλήρωση αυτού του βήματος το αποτύπωμα υπολογίζεται για κάθε αρχείο ώστε να μπορεί να χρησιμοποιηθεί σαν αναζήτηση για αντίστοιχα IDs. Αυτό το ID συνδέεται με ένα δοσμένο τραγούδι ή ηχογράφιση και από εκεί συλλέγονται τα μεταδεδομένα.

TRM

Το TRM ήταν το πρώτο σύστημα που χρησιμοποίησε το MusicBrainz. Χρησιμοποιήθηκε στην αρχική εφαρμογή ετικετών και η βιβλιοθήκη που απευθυνόταν στους πελάτες ήταν ανοιχτού κώδικα και ολοκληρωνόταν στο libmusicbrainz. Δούλευε εξαιρετικά καλά με το ξεκαθάρισμα δίπλων αρχείων σε έναν υπολογιστή, ωστόσο αντιμετώπιζε προβλήματα με συγκρούσεις (διαφορετικούς ήχους με το ίδιο ID) και διπλότυπα (ίδιοι ήχοι με διαφορετικά IDs). Ένα ακόμα πρόβλημα ήταν ότι ο server δεν ήταν ικανός να διαχειριστεί τον αριθμό των TRMs που χρειαζόντουσαν για το Musicbrainz.

PUID

Το PUIDs είναι το δεύτερο σύστημα ακουστικού αποτυπώματος που χρησιμοποίησε το Musicbrainz και αγοράστηκε από το Gracenote τον Ιούνιο του 2011. Ήταν καλύτερο σύστημα από το TRM αλλά αντιμετώπιζε κάποια σοβαρά προβλήματα. Το σύστημα δεν ήταν ανοιχτού κώδικα και σαν αποτέλεσμα είχε να μη μπορεί να συμπεριληφθεί στο Picard. Η διαδικασία παραγωγής αποτυπωμάτων ήταν πολύ αργή τόσο στη μεριά του πελάτη όσο και του server. Τα PUIDs είναι αρκετά αδιαφανή. Δεν είναι τίποτε περισσότερο από ένα μοναδικό αριθμό που παραπέμπει σε μια βάση δεδομένων έξω από τον έλεγχο του MusicBrainz, και αν αυτή η βάση δεδομένων πάψει να υπάρχει τότε δεν έχουν καμία χρήση.

AcoustID

Το AcoustID είναι το τρίτο σύστημα του Musicbrainz και αυτό που χρησιμοποιεί αυτή τη στιγμή. Τα πλεονεκτήματά του είναι ότι είναι ανοιχτού κώδικα, είναι εξελίξιμο σε συνδυασμό με κάποιο υποστηρικτικό λογισμικό και δίνει την δυνατότητα της σύγκρισης μουσικής παραστατικά. Τα αποτυπώματα που παράγει έχουν καταγεγραμμένη τη διάρκεια τους κάτι το οποίο κάνει ευκολότερο το να απορρίπτονται ορισμένες εσφαλμένες συνδέσεις μεταξύ ηχογραφήσεων και acoustIDs

Η διαδικασία του ακουστικού αποτυπώματος είναι μια σχετικά μικρή διαδικασία. Αναλύονται περίπου 2 λεπτά του track μέσω της ανοιχτής βιβλιοθήκης libofa ώστε να υπολογιστούν τα ακουστικά αποτυπώματα. Στην συνέχεια λαμβάνονται 2-3 δευτερόλεπτα για κάθε track. Με τα συγκεκριμένα δεδομένα μπορεί να πραγματοποιηθεί μόνο μια αναζήτηση μέσα από το MusicDNS web-service, η οποία επιστρέφει ένα PUID αν έχει υπάρξει μια αρκετά ικανοποιητική ταυτοποίηση μέσω της ανάλυσης.

3.5. Συνεργασίες

Το MusicBrainz διαθέτει μια τεράστια βάση δεδομένων ελεύθερη προς το ευρύ κοινό και όπως είναι φυσικό μεγάλες μουσικές υπηρεσίες συνεργάζονται μαζί του ώστε να έχουν πρόσβαση στα δεδομένα που προσφέρει. Μουσικές υπηρεσίες όπως το Last.FM, GrooveShark, Pandora, Music και Echo nest χρησιμοποιούν τα δεδομένα του MusicBrainz αλλά ακόμα και το BBC συνεργάζεται μαζί του και έχει εμπλουτίσει τις διαδικτυακές μουσικές σελίδες του με τα δεδομένα του Musicbrainz. Η συνεργασία με το echo nest βοήθησε στην δημιουργία του echoprint το οποίο έχει αναλυθεί σε προηγούμενο κεφάλαιο. Το Musicbrainz παρέχει όλα τα δεδομένα της

βάσης δεδομένων του στο echoprint για χρήση από προγραμματιστές και απλούς χρήστες και δέχεται πίσω τα ακουστικά αποτυπώματα που δημιουργούνται μέσα από τις αναζητήσεις. Κάτι που είναι και μια από τις προϋποθέσεις για την χρήση του echoprint.

Το ίδιο το echoprint δίνει τα εργαλεία τα οποία πραγματοποιούν τρία βήματα:

1. Χρησιμοποιεί το API του Echo Nest ώστε να προβλέψει το όνομα του τραγουδιού και του καλλιτέχνη από ένα αρχείο ήχου.
2. Ταιριάζει τις λίστες των tracks με τις κυκλοφορίες του Musicbrainz.
3. Αναγνωρίζει τις κυκλοφορίες του Musicbrainz που έχουν τον ίδιο αριθμό tracks με τα αρχεία του Echo Nest.

Κεφάλαιο 4

1. ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ

1.1. Εισαγωγή

Ο σκοπός είναι η δημιουργία μιας ιστοσελίδας η οποία θα συνδέεται με μια online βάση δεδομένων και θα επιτρέπει την αναγνώριση τραγουδιών είτε μέσα από τον σκληρό δίσκο είτε μέσω ηχογράφησης με μικρόφωνο (μέσω αέρα). Ένα κομμάτι θα επιλέγεται μέσα από τον σκληρό δίσκο, θα αποστέλλεται στην βάση και θα επιστρέφεται το αποτέλεσμα το οποίο αποτελείται από το όνομα του καλλιτέχνη, το όνομα του τραγουδιού το άλμπουμ στο οποίο βρίσκεται, την ημερομηνία κυκλοφορίας καθώς και ένα μικρό δείγμα του τραγουδιού που μπορεί να ακουστεί πατώντας το κουμπί αναπαραγωγής. Αντίστοιχα η ίδια διαδικασία θα συμβαίνει μετά από ηχογράφηση ενός τραγουδιού μέσω μικρόφωνου. Ως βάση δεδομένων αρχικά είχε επιλεγεί η βάση δεδομένων του echoprint. Ωστόσο η Echonest αποφάσισε να μην το διαθέτει πλέον για τέτοιους σκοπούς οπότε και υπήρξε η ανάγκη να βρεθεί κάποια άλλη λύση. Σε αυτή τη περίπτωση και μετά από μια μικρή έρευνα στο διαδίκτυο πλέον το Doreso είναι το νέο εργαλείο που θα βοηθήσει στην επίτευξη της ιστοσελίδας.

1.2. Doreso

Η Doreso είναι μια εταιρία με βάση την Κίνα, που σχετίζεται με την αυτόματη αναγνώριση περιεχομένου και ειδικεύεται στην αναγνώριση ήχου και video. Συνεργάζεται με μεγάλες εταιρίες όπως οι Samsung, Huawei, HTC και China mobile και έχει αναπτύξει εφαρμογές για μουσική αναγνώριση μέσω κινητού τηλεφώνου.

Η Doreso διαθέτει μια βάση δεδομένων η οποία περιέχει περίπου 47 εκατομμύρια αποτυπώματα μουσικών κομματιών. Δίνει τη δυνατότητα αναγνώρισης ενός κομματιού μέσα σε έναν χρονικό μέσο όρο περίπου 2-5 δευτερόλεπτων και η αναγνώριση μπορεί να γίνει ακόμα και τραγουδώντας την μελωδία με το στόμα (humming). Οι διαφορές του με την echonest είναι ότι πρέπει να σταλεί ο ήχος και όχι το αποτύπωμα ώστε να υπάρξει η αναγνώριση και δίνει την δυνατότητα για 80 αναζητήσεις την ώρα. Για την χρησιμοποίηση των υπηρεσιών του Doreso όπως και με την echonest είναι απαραίτητο να υπάρχει ένα API KEY το οποίο δίνεται μετά από αίτημα στην εταιρία.

1.2.1. Doreso API & API key

Ένα API (application programming interface) είναι ένα σύνολο από ρουτίνες (routines), δομές δεδομένων (data structures), κλάσσεις αντικειμένων (object classes) και/ή πρωτόκολλα (protocols) που υποστηρίζουν την ανάπτυξη λογισμικού και υπηρεσιών βασισμένων σε μια πρωτογενή πηγή (υπολογιστικό σύστημα, βιβλιοθήκη, εφαρμογή). Ένας από τους βασικούς σκοπούς του είναι να ορίζει και να διατυπώνει το σύνολο των λειτουργιών-υπηρεσιών που μπορεί να παρέχει μια βιβλιοθήκη ή ένα λειτουργικό σύστημα σε άλλα προγράμματα, χωρίς να επιτρέπει πρόσβαση στον κώδικα που υλοποιεί αυτές τις υπηρεσίες. Η διεπαφή, ένα

«συμβόλαιο κλήσης» μεταξύ καλούντος και καλούμενου, διαχωρίζει την προγραμματιστική υλοποίηση κάποιων υπηρεσιών από τη χρήση τους.

Ένα application programming interface key (API key) είναι ένας κώδικας ενός προγράμματος ηλεκτρονικού υπολογιστή ο οποίος καλεί ένα API για να αναγνωρίσει το πρόγραμμα κλήσης, τον προγραμματιστή του ή τον χρήστη του Web site. Ουσιαστικά ένα API key χρησιμοποιείται ώστε να εντοπίζει και να ελέγχει κατά ποιον τρόπο χρησιμοποιείται ένα API.

Το API του Doreso επιτρέπει στους προγραμματιστές να ενσωματώσουν τις υπηρεσίες του στις εφαρμογές τους. Το API, δίνει την δυνατότητα στους προγραμματιστές να επιτρέπουν στους χρηστές των εφαρμογών τους να εξερευνούν και να ανακτούν πληροφορίες πάνω σε διαφορετικά πεδία (μουσική /TV videos).

Το API του Doreso επιτρέπει να καλούνται μέθοδοι γραμμένοι σε JSON. Αρχικά έπρεπε να γίνει εγγραφή στο <http://developer.doreso.com> ώστε να αποκτηθεί το API key και στην πορεία να είναι δυνατή η χρήση του API. Όπως αναφέρθηκε παραπάνω οι χρηστές έχουν έναν όριο (rate limit) των 80 κλήσεων την ώρα. Κάθε φορά που καλείται προς χρήση το API επιστρέφεται μαζί με το αποτέλεσμα και μια επικεφαλίδα που αναφέρει την κλήση αυτή. Η επικεφαλίδα περιέχει τρία πεδία τα οποία σχετίζονται με το rate limit:

X-RateLimit-Limit – το τρέχον όριο για το συγκεκριμένο κλειδί.

X-RateLimit-Used – ο αριθμός των κλήσεων που έγιναν από το συγκεκριμένο κλειδί την δεδομένη στιγμή.

X-RateLimit-Remaining - ο εκτιμώμενος αριθμός των υπόλοιπων κλήσεων που επιτρέπονται από αυτό το κλειδί τη δεδομένη στιγμή.

Το API key εισάγεται στον κώδικα με την μορφή του παραδείγματος :

```
doreso = Doreso('my API key')  
  
d = doreso.song_identify_file('./filename.mp3')  
  
print json.dumps(d, indent=4)
```

1. Το API μπορεί να αναγνωρίσει αρχεία ήχου οποιασδήποτε μορφοποίησης. Η μέθοδος απαιτεί ένα HTTP Post request. Η ανάκτηση των μουσικών δεδομένων μπορεί να γίνει με δυο τρόπους:
2. Ένα HTTP POST request με Content-Type "application/octet-stream", και με το τοπικό αρχείο σαν το σώμα του request, και τις παραμέτρους του στο URL. Ένα HTTP POST request με Content-Type "multipart/form-data" όπου τα αρχεία ήχου βρίσκονται στο "track" τμήμα.

1.3. Εργαλεία

Η υλοποίηση της ιστοσελίδας και ο προγραμματισμός αυτής έγινε σε περιβάλλον linux και πιο συγκεκριμένα Ubuntu. Ο text editor που χρησιμοποιήθηκε για τις γλώσσες προγραμματισμού είναι ο sublime text 2.

Τα εργαλεία που χρησιμοποιήθηκαν είναι τα εξής :

- Python, και πιο συγκεκριμένα το flask framework της python για την κατασκευή του server και το python API του Django.
- HTML 5 για την κατασκευή της ιστοσελίδας.
- Css για την μορφοποίηση της ιστοσελίδας.
- Javascript για την ηχογράφηση του ήχου και συγκεκριμένα το recorder.js το οποίο είναι ένα plugin γραμμένο σε javascript για την ηχογράφηση μέσω μικρόφωνου.
- Το framework του bootstrap για την κατασκευή των κουμπιών (views) της ιστοσελίδας.
- Deezer API

Το **Flask** είναι ένα web application framework (WAF) δηλαδή είναι ένα λογισμικό πλαίσιο το οποίο έχει σχεδιαστεί ώστε να συνεισφέρει στον σχεδιασμό δυναμικών ιστοσελίδων web εφαρμογών και υπηρεσιών. Το Flask ανήκει στην κατηγορία των micro-framework. Είναι κανονικά frameworks που εξαρτώνται από λίγο ή καθόλου σε εξωτερικές βιβλιοθήκες. Πρόκειται για ένα ελαφρύ framework με μικρές ανάγκες για αναβαθμίσεις και προσοχή για security bugs. Κάποιες φορές οι διεργασίες πρέπει να γίνονται από τον χρήστη όπως το να αυξάνει τις λίστες των εξαρτήσεων του (dependencies) προσθέτοντας plugins. Για το Flask, οι εξαρτήσεις του είναι τα : Werkzeug μια WSGI (Web Server Gateway Interface) βιβλιοθήκη γενικής χρήσης και το jinja2 το οποίο πρόκειται για ένα template engine.

Το **recorder.js** είναι ένα plugin γραμμένο σε javascript για ηχογράφηση αρχείων ήχου και εξαγωγή τους μέσω download. Αυτό που κάνει είναι να ηχογραφεί ήχο μέσω του μικρόφωνου του υπολογιστή και να το αποθηκεύει στον σκληρό δίσκο σε μορφή WAV. Το recorder.js χρησιμοποιεί τα HTML media capture και WEB AUDIO API της εταιρίας W3C.

Το **HTML Media Capture** περιγράφεται σαν μια HTML επέκταση που εξασφαλίζει στον χρήστη πρόσβαση στον μηχανισμό δέσμευσης πληροφορίας μιας συσκευής. Τέτοιες συσκευές μπορεί να είναι μια κάμερα ή ένα μικρόφωνο και η λειτουργία γίνεται μέσα από ένα αρχείο έλεγχου του upload και με αποτέλεσμα την επιτόπου δέσμευση πληροφορίας.

Το **Web audio API** είναι ένα JavaScript API για επεξεργασία και σύνθεση ήχου σε web εφαρμογές. Ένα βασικό παράδειγμα είναι αυτό ενός γραφήματος δρομολόγησης ήχου όπου ένας αριθμός από Audio Node αντικείμενα (τα δομικά στοιχεία ενός *AudioContext*, μια διασύνδεση που αποτελείται από πηγές ήχου, τον προορισμό του ήχου, και τα ενδιάμεσα τμήματα επεξεργασίας) συνδεδεμένα μεταξύ τους ώστε να προσδιορίζουν τη συνολική απόδοση ήχου. Η πραγματική επεξεργασία θα λάβει χώρα στην εφαρμογή που εφαρμόζεται, με απευθείας επεξεργασία και

σύνθεση, με Javascript να υποστηρίζεται επίσης. Κάποια από τα χαρακτηριστικά του είναι :

- Ανεξάρτητες δρομολογήσεις για απλές ή σύνθετες mixing/effect αρχιτεκτονικές, συμπεριλαμβανομένου πολλαπλές αποστολές και υπομίξεις.
- Αναπαραγωγή δειγμάτων ήχου με χαμηλό latency για μουσικές εφαρμογές.
- Επεξεργασία αρχείων ήχου μέσα από μια audio ή video πηγή.
- Επεξεργασία live ηχητικών εισόδων χρησιμοποιώντας ένα MediaStream από την συνάρτηση getUserMedia().
- Επεξεργασία ήχου από μια απομακρυσμένη ηχητική πηγή.
- Αποστολή ενός αναπαραγόμενου ή επεξεργασμένου αρχείου ήχου σε έναν απομακρυσμένο peer μέσω μιας ηχητικής πηγής.
- Μια μηχανή για μεγάλου εύρους γραμμικά εφέ και πιο συγκεκριμένα εφέ δωματίου.
- Αποτελεσματική ανάλυση του τομέα χρόνου/συχνοτήτων σε πραγματικό χρόνο και υποστήριξη music visualizer.
- Αποτελεσματικά biquad φίλτρα για lowpass, highpass, και άλλα κοινά φίλτρα.
- Ηχογράφηση ήχου από μια ηχητική πηγή.

Το **Deezer** είναι μια διαδικτυακή streaming υπηρεσία που επιτρέπει στους χρήστες να ακούνε μουσική σε διάφορες συσκευές όπως κινητά τηλέφωνα είτε είναι online είτε offline. Το Deezer έχει το δικό του API και μια βάση δεδομένων με πάνω από 18 εκατομμύρια καταχωρήσεις. Σε συνδυασμό με την Echo Nest έχουν καταφέρει όλες οι κλήσεις από το Echo Nest API να συνδυάζονται με τα περιεχόμενα της βάσης δεδομένων του Deezer και να υπάρχει μια αλληλεπίδραση μεταξύ των δυο υπηρεσιών. Συνδυάζοντας αυτά τα δυο μαζί οι προγραμματιστές μπορούν να δημιουργούν εφαρμογές που υποστηρίζουν full audio stream playback για οποιονδήποτε είναι συνδρομητής στο Deezer, με δείγματα των 30 δευτερολέπτων για όλους τους χρήστες. Οι πληροφορίες που προσφέρονται μέσα από τον κατάλογο του περιλαμβάνουν εξώφυλλο άλμπουμ, ένα μικρό δείγμα του τραγουδιού, όνομα καλλιτέχνη και τραγουδιού, ημερομηνία κυκλοφορίας και πολλά άλλα.

Το **Bootstrap** είναι μια συλλογή εργαλείων ανοιχτού κώδικα για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Περιέχει HTML και CSS για τις μορφές τυπογραφίας, κουμπιά πλοήγησης και άλλων στοιχείων του περιβάλλοντος, καθώς και προαιρετικές επεκτάσεις JavaScript. Αποτελείται ουσιαστικά από μια σειρά στυλ (stylesheet) που εφαρμόζουν τα διάφορα συστατικά του πακέτου εργαλείων. Ένα στυλ που ονομάζεται bootstrap.less περιλαμβάνει τα συστατικά stylesheets. Οι προγραμματιστές μπορούν να προσαρμόσουν το αρχείο Bootstrap, επιλέγοντας τα στοιχεία που θέλουν να χρησιμοποιήσουν στο έργο τους.

Για το στήσιμο του server με τον οποίο τρέχει η ιστοσελίδα χρησιμοποιήθηκαν οι υπηρεσίες του okeanos (<https://okeanos.grnet.gr>) και πιο συγκεκριμένα η υπηρεσία του με το όνομα Κυκλάδες. Πρόκειται για μια cloud υπηρεσία από την ερευνητική και ακαδημαϊκή ελληνική κοινότητα η οποία δίνει την δυνατότητα για κατασκευή virtual machines και virtual networks. Έχει σχεδιαστεί από το Εθνικό Δίκτυο Έρευνας και Τεχνολογίας (ΕΔΕΤ) και μπορεί να χρησιμοποιηθεί από φοιτητές, καθηγητές και ερευνητές με τα ίδια πιστοποιητικά με τα οποία θα συνδεόντουσαν στις πανεπιστημιακές σχολές ή τα ερευνητικά τους κέντρα.

1.4. Υλοποίηση

Η υλοποίηση της ιστοσελίδας μπορεί να περιγραφεί σε 4 βήματα :

- Αρχικά ορίστηκαν τα URL routes για την αρχική σελίδα, την αναγνώριση και την πρόσβαση των ανεβασμένων κομματιών.
- Στη συνέχεια σχεδιάστηκαν τα html views της εφαρμογής με το index (αρχική σελίδα) να είναι στατικό και το result.html (σελίδα αποτελεσμάτων) δυναμικό χρησιμοποιώντας την web template μηχανή που χρησιμοποιεί το Flask (Jinja2).
- Η δυνατότητα ηχογράφησης από τον browser προστέθηκε στην αρχική σελίδα με τη χρήση του recorder.js plugin το οποίο με τη σειρά του κάνει χρήση των HTML Media Capture και Web Audio API της W3C.
- Τέλος προστέθηκε μορφοποίηση των views με χρήση CSS από το Bootstrap framework

Όπως αναφέρθηκε και παραπάνω η πρώτη κίνηση είναι να αποκτηθεί το Doreso API key ώστε να είναι δυνατή η πρόσβαση στο API του Doreso και στη βάση δεδομένων του, από την οποία και θα ανακτηθούν οι πληροφορίες για τα τραγούδια. Επίσης αποκτήθηκε και το API key του Echo Nest. Στη συνέχεια δημιουργήθηκε ένα virtual machine με την υπηρεσία Κυκλάδες του ωκεανού. Το συγκεκριμένο virtual machine τρέχει σε λογισμικό Ubuntu server 12.04 έχει έναν πυρήνα και 1GB ram. Μετά την δημιουργία του έγινε η εγκατάσταση των προαπαιτούμενων πακέτων (python, ffmpeg κλπ) ώστε να τρέξει η εφαρμογή και τέλος έγινε deploy η εφαρμογή app.py. Στη συνέχεια σχεδιάστηκε η λογική του Controller της web εφαρμογής η οποία βρίσκεται στο app.py. Ορίστηκαν τα requests που θα δέχεται ο server, την μορφή που θα έχουν τα URL και αναπτύχθηκε το τι θα επιστρέφεται σαν απάντηση στον server.

Το app.py αφορά την control λογική του server. Δηλαδή την λογική που αφορά την λήψη και εξυπηρέτηση των http requests που στέλνει και τα responses που δέχεται ο client από τον server. Στην συγκεκριμένη εφαρμογή τα responses είναι html σελίδες. Το τραγούδι στέλνεται για αναγνώριση στο Doreso και αφού επιστραφούν τα στοιχεία, μέσω της συνάρτησης *song.search* του Echo Nest βρίσκει το τραγούδι με τα συγκεκριμένα στοιχεία που επιστρέφονται. Αυτά στην συνέχεια αντιστοιχούνται με το ανάλογο track στο Deezer και αφού ανακτηθεί ο χαρακτηριστικός κωδικός του το Deezer επιστρέφει την πρώτη επιλογή από τα αποτελέσματα που θα βρει.

Μέσω του Flask ορίστηκαν τα εξής routes:

Το (/) το οποίο δρομολογεί τα http get requests που έχουν σαν url το αρχικό url του server και επιστρέφει την αρχική σελίδα στον χρήστη.

Το (/identify) το οποίο δέχεται τα post requests που στέλνονται από την αρχική σελίδα και την φόρμα αποστολής αρχείων που βρίσκεται σε αυτή. Αυτό που επιστρέφεται είναι η δυναμική σελίδα *results.html* με τα δεδομένα που δεχόμαστε από το Doreso.

```

26 #Αν απλα ζητηθει το αρχικο url επιστρεφεται η συναρτηση
27 #που επιστρεφει τον χρηστη στην αρχικη σελιδα
28 @app.route('/')
29 def index():
30     return render_template('index.html')
31
32
33 # Route που θα επεξεργαστει το ανεβασμα του αρχειου
34 @app.route('/identify', methods=['POST'])
35 def upload():

```

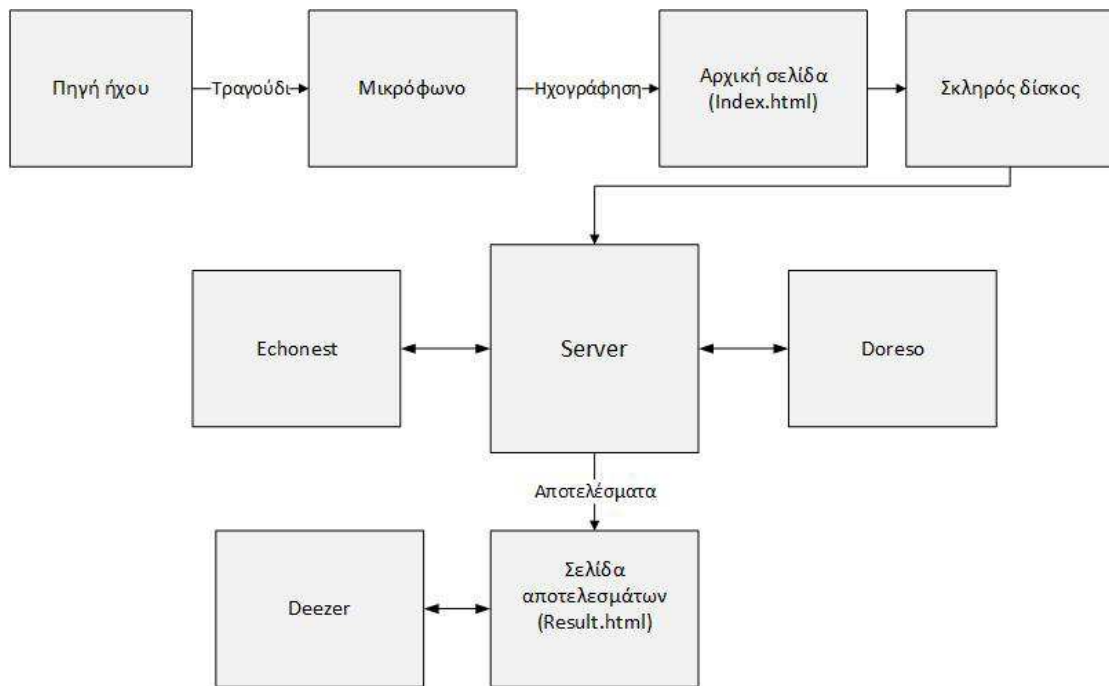
Εικόνα 8: Ο ορισμός των routes στο app.py

Όταν ο Server δέχεται το παραπάνω identify request, εξάγει το όνομα του ανεβασμένου αρχείου και κάνει τον απαραίτητο έλεγχο για το αν υπάρχει αρχείο και αν η μορφοποίηση του είναι η αποδεκτή μέσω της συνάρτησης *allowed_file(file.filename)* του Flask. Στην συγκεκριμένη εφαρμογή έχουν οριστεί σαν αποδεκτές μορφοποιήσεις οι mp3 και WAV. Στην συνέχεια το αρχείο αποθηκεύεται στον server για μελλοντική χρήση όπως για παράδειγμα την δημιουργία μιας βάσης δεδομένων μέσω των ανεβασμένων αρχείων.

Επόμενο βήμα είναι η δημιουργία ενός instance του Doreso μέσω του Doreso app key και εμφανίζεται ως:

doreso=pydoreso.Doreso('wqGyoPaBZHF2dBHpRsoijplwmH0qzF6t9kO6BeZ7LQE')

όπου αυτό είναι και το κλειδί το οποίο δόθηκε από την ίδια την εταιρία. Στη συνέχεια καλείται η συνάρτηση *Song identify* του Doreso και σαν όρισμα παίρνει το αρχείο το οποίο ανέβηκε από τον χρήστη. Το αποτέλεσμα που λαμβάνεται μέσα από το Doreso δίνεται στην αναζήτηση του Echo Nest όπου με τα στοιχεία που λαμβάνει ελέγχει αν υπάρχει το αντίστοιχο track στο Deezer και ζητεί τον κωδικό αυτού. Αφού δοθεί σαν όρισμα στην render template συνάρτηση του Flask ώστε να δημιουργηθεί η δυναμική σελίδα result.html, στην πορεία δίνονται τα αποτελέσματα στην σελίδα και αφού αναγνωρίσει τον κωδικό του track από το Deezer, με το που φορτώσει η σελίδα με JQuery, λαμβάνονται οι πληροφορίες από το API του Deezer για τον συγκεκριμένο κωδικό που εμφανίζεται στην τελική σελίδα. Αυτή η διαδικασία μπορεί να παρουσιαστεί σχηματικά στην εικόνα 9.



Εικόνα 9 : Σχηματική αναπαράσταση της λειτουργίας της εφαρμογής

Οι εντολές που υπάρχουν στην σελίδα results.html είναι γραμμένες με βάση την template γλώσσα του Jinja που χρησιμοποιεί το Doreso και είναι υπεύθυνη για τη δημιουργία της σελίδας επιστροφής των αποτελεσμάτων την οποία βλέπει ο χρήστης. Τα αποτελέσματα που επιστρέφονται είναι ο τίτλος του τραγουδιού, όνομα καλλιτέχνη και άλμπουμ, ημερομηνία κυκλοφορίας, και μια μικρή προεπισκόπηση του κομματιού. Με λίγα λόγια οπτικοποιεί σε μορφή HTML τα Json αποτελέσματα που επιστρέφει το Doreso. Τέλος η εφαρμογή ορίζεται να υπακούει στην public ip του virtual machine στο οποίο τρέχει ο server.

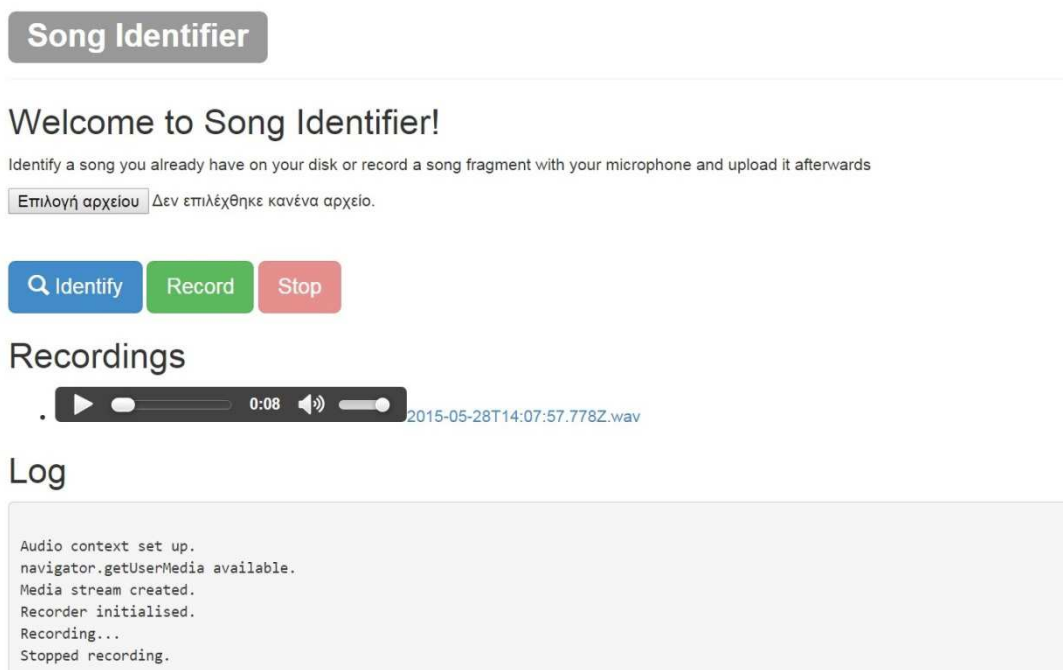
Η αρχική σελίδα της web εφαρμογής περιλαμβάνει επίσης το Script.js το οποίο και περιέχει τον απαραίτητο javascript κώδικα για την ηχογράφηση από το μικρόφωνο .

Πιο συγκεκριμένα με το που φορτωθεί από τον χρήστη η αρχική σελίδα της εφαρμογής, καλείται η συνάρτηση `window.onload` η οποία θα κάνει τους απαραίτητους ελέγχους για την υποστήριξη ηχογράφησης μέσα από τον browser που χρησιμοποιεί την δεδομένη στιγμή ο χρήστης. Αυτός ο έλεγχος θα γίνει μέσα από το `recorder.js` το οποίο σε περίπτωση μη συμβατότητας τυπώνει το ανάλογο μήνυμα. Αλλιώς αρχικοποιεί το `recorder.js` ώστε να είναι έτοιμο για την ηχογράφηση. Η διαδικασία αυτή για τον χρήστη γίνεται μέσω ενός ερωτήματος για το αν επιτρέπει τη χρήση του μικρόφωνου του και είναι κάτι το οποίο θα συμβαίνει κάθε φορά που θα φορτώνεται αυτή η σελίδα.

Με το που φορτωθεί η σελίδα ο χρήστης έχει τις εξής επιλογές :

- Επιλογή ενός αρχείου μέσα από τον σκληρό του δίσκο.
- Identify, όπου το αρχείο θα ανέβει στον server και θα ξεκινήσει η διαδικασία ώστε να επιστραφούν τα αποτελέσματα.
- Record, όπου ο χρήστης θα ηχογραφήσει το ηχητικό απόσπασμα μέσω μια εξωτερικής πηγής ήχου.
- Stop ώστε να σταματήσει η ηχογράφηση.

Μετά το τέλος της ηχογράφησης το ηχογραφημένο απόσπασμα θα εμφανιστεί στην ίδια σελίδα με τις επιλογές της αναπαραγωγής και του κατεβάσματος στον σκληρό δίσκο. Αφού το αρχείο κατέβει και αποθηκευτεί στον σκληρό δίσκο τότε ο χρήστης μέσα από την επιλογή αρχείου και το identify μπορεί να λάβει τις πληροφορίες που επιθυμεί από τη βάση δεδομένων του Doreso. Η ηχογράφηση δεν χρειάζεται να διαρκέσει για μεγάλο χρονικό διάστημα αφού το Doreso είναι ικανό να κάνει την αναγνώριση και με ένα απόσπασμα πέντε δευτερολέπτων. Όπως είναι λογικό επίσης ο χρόνος που θα χρειαστεί για να επιστραφούν οι πληροφορίες του τραγουδιού είναι ανάλογος του μεγέθους του αποσπάσματος που θα σταλεί στο Doreso. Στις εικόνες 10 και 11 αντίστοιχα, φαίνεται αρχική σελίδα της εφαρμογής αφού έχει φορτωθεί και αφού έχει επιστρέψει τα αποτελέσματα της αναζήτησης.



Εικόνα 10 : Η αρχική σελίδα της εφαρμογής

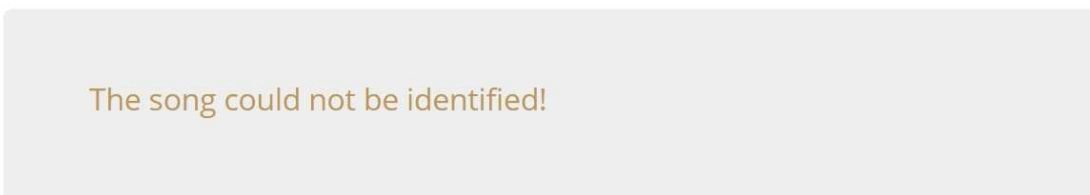
Song Identifier



Εικόνα 11 : Η σελίδα αποτελεσμάτων της εφαρμογής

Στην περίπτωση αποτυχίας αναγνώρισης του μουσικού κομματιού στην σελίδα των αποτελεσμάτων θα εμφανιστεί το μήνυμα « The song could not be identified!» όπως φαίνεται και στην εικόνα 12 .

Song Identifier



Εικόνα 12: Μήνυμα που επιστρέφεται σε περίπτωση αποτυχίας αναγνώρισης

Η web εφαρμογή βρίσκεται στην διεύθυνση <http://snf-646146.vm.oceanos.grnet.gr/> και οι κώδικες για την δημιουργία της βρίσκονται στο παράρτημα της πτυχιακής στην σελίδα 83.

2. Στατιστικά

Για να δοκιμαστεί η αποτελεσματικότητα της Web εφαρμογής επιλέχτηκε ένας αριθμός μουσικών κομματιών τεσσάρων διαφορετικών κατηγοριών. Οι κατηγορίες χωρίζονται σε *Rock& metal*, *Commercial (pop, house, rnb)*, *Classical*, *Ελληνικά*. Ο σκοπός είναι να δούμε το ποσοστό επιτυχίας αναγνώρισης των κομματιών ανάλογα με το είδος αλλά και το πόσο διαδεδομένα είναι. Όλα τα ηχητικά αποσπάσματα που

θα αποσταλούν στο Dogeso θα είναι από δέκα μέχρι είκοσι δευτερόλεπτα. Από κάθε κατηγορία έχουν επιλεγεί 25 τραγούδια. Ο μέσος Όρος του χρόνου που χρειάζεται να ανέβουν τα κομμάτια μέχρι να επιστραφούν τα αποτελέσματα υπολογίστηκε στα 12 δευτερόλεπτα. Όπως έχει αναφερθεί και παραπάνω ο χρόνος που χρειάζεται να επιστραφούν τα δεδομένα είναι ανάλογος του χρόνου του ηχογραφημένου αποσπάσματος. Σκοπός ήταν να μην είναι στην πλειοψηφία τους από τα πιο εμπορικά και γνώριμα στο ευρύ κοινό. Στην επόμενη σελίδα παρουσιάζεται ο πίνακας με τις αναζητήσεις και αναγνωρίσεις των μουσικών κόμματων ανά κατηγορία.

Μπάντα/καλλιτέχνης	Τραγούδι	Επιτυχία αναγνώρισης
	ROCK & METAL	
Ben Howard	Oats in the water	Ναι
Pink Floyd	Money	Ναι
Anathema	Anathema	Ναι
Steelheart	She is gone	Όχι
Antimatter	Epitaph	Όχι
Pearl jam	Just breathe	Ναι
Peter Carlsen	Pull the brakes	Ναι
Amorphis	You I need	Ναι
Crematory	Tears of time	Ναι
Danzing	Am I Demon	Όχι
Fleetwood Mac	Dreams	Ναι
Louisiana Red	Bring it on home to me	Ναι
Johnny Lang	Lie to me	Ναι (no artist)
Behemoth	Blow your trumpets Gabriel	Όχι
Kyus	One inch man	Όχι
Waterboys	Fisherman blues	Όχι
James	Getting away	Ναι (no artist)
Bob Dylan	Hurricane	Ναι
Black rebel motorcycle club	Beat the devils tattoo	Ναι
Craig Walker	Siamese	Όχι
Eric Clapton	Knocking on heavens door	Ναι
Golden Earing	Radar love	Ναι
Pantera	Cemetery gates	Όχι
Black keys	too afraid to love you	Όχι
The Raconteurs	Love in a thrashcan	Ναι
Dark tranquility	Auctioned	Ναι
	COMMERCIAL	
Μπάντα/καλλιτέχνης	Τραγούδι	Επιτυχία αναγνώρισης
John Legend	All of me	Ναι
Steve Aoki	Love it when you cry	Όχι
Ellie Goulding	Love me like you do	Ναι
Natalie La Rose	Somebody	Ναι
Nicolas Jaar	Mi mujer	Ναι
Andru Donalds	Mishale	Ναι

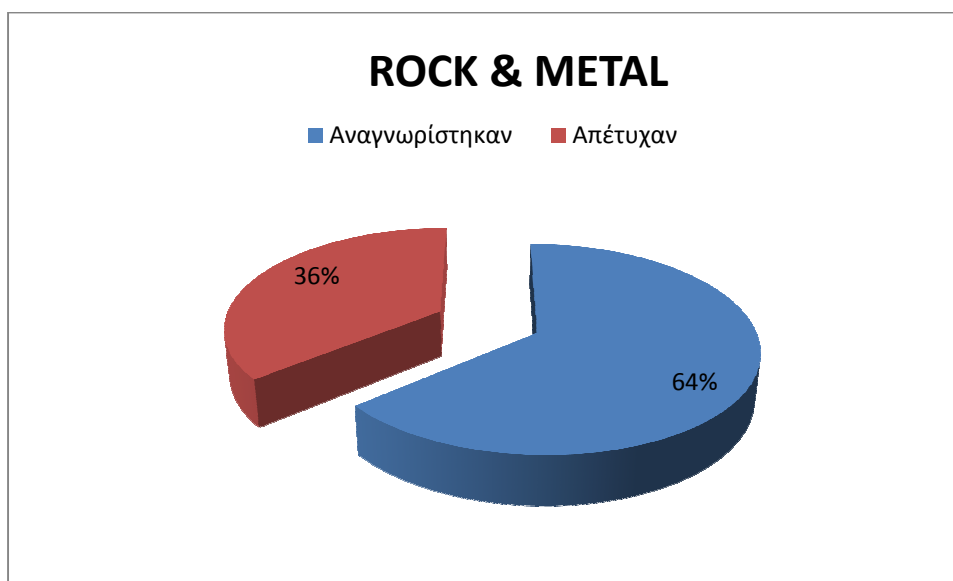
Μπάντα/καλλιτέχνης	Τραγούδι	Επιτυχία αναγνώρισης
Empire of the sun	We are the people	Ναι
Schiller	Let me love you	Όχι
Eddie Murphy	Party all the time	Όχι
Avicii	Addicted to you	Ναι
David Guetta	Love is gone	Όχι
Yves la rock	Rise up	Όχι
Wamdue Project	king of my castle 09	Ναι
Passenger	Let Her Go (ALCEEN & SERGIO Remix)	Όχι
James Morrison	Broken strings	Ναι
Marina and the Diamonds	Primadonna	Ναι
Cheryl Cole	Fight for this love	Όχι
Sean Paul	She doesn't mind	Ναι
Steve Aoki	Boneless	Όχι
Snow	Informer	Ναι
Morandi	In and out of love	Όχι
Ronan Keating	When you say nothing at all	Ναι
Black eyed peace	Tonight's gonna be a good night	Όχι
Snoop Dog & Dr Dre	Still Dre	Ναι (no artist)
	CLASSICAL	
Συνθέτης	Τραγούδι	Επιτυχία αναγνώρισης
Zbigniew Preisner	Lacrimosa	Ναι
Mozart	Requiem Mass in D minor	Όχι
Beethoven	Fur elise	Όχι
Yanni	Nightigalle	Όχι
Vivaldi	Spring	Ναι
Schubert	Serenade	Όχι
Smetana	Die moldau	Όχι
Beethoven	Ode to joy	Ναι
Ludovico Einaudi	Una Mattina	Όχι
Hans Zimmer	Time	Όχι
Lux Aeterna	Requiem for a dream	Όχι
Cinematic orchestra	Arrival of the birds	Όχι
Giovani Battista Pergolesi	Stabat mater	Όχι
Gregorio Allegri	Miserere	Όχι
Maria Callas	Carmen	Ναι

Συνθέτης	Τραγούδι	Επιτυχία αναγνώρισης
Bach	Jesus Bleibet meine freude	Όχι
Piotr Anderszewski	Mazurkas op :59 : N3 in f sharp minor	Ναι
Korsakov	Flight of Bumblebee	Όχι
John Williams	Imperial march	Ναι
Wagner	Ride of the valkyries	Όχι
Adam Hurst	Seduction	Όχι
Chopin	Nocturne	Όχι
Bach	Toccata and fugue	Όχι
Edvard Grieg	In the Hall of the Mountain King	Όχι
	ΕΛΛΗΝΙΚΑ	
Μπάντα/καλλιτέχνης	Τραγούδι	Επιτυχία αναγνώρισης
Αλκίνοος Ιωαννίδης	Δεν μπορώ	Ναι
Σάκης Ρουβάς	Ραδιόφωνα	Ναι
Γιάννης Χαρούλης	Τι λάθος κάνω	Ναι
Τρύπες	Ταξιδιάρια ψυχή	Ναι
One	Μωρό μου	Ναι
Τριαντάφυλλος	Κράτα τα όλα	Ναι
Βασίλης Τσιτσάνης	Με παρέσυρε το ρέμα	Ναι
Stavento	Για σένα που λατρεύω	Όχι
Γιώργος Μαζωνάκης	Τέρμα	Ναι
Πάολα	Έχω μια ζωή	Ναι
Διάφανα κρίνα	Μέρες αργίας	Όχι
Νίκος Παπάζογλου	Εγώ δεν είμαι ποιητής	Όχι
Θανάσης Παπακωνσταντίνου	Στις χαραυγιές ξεχνιέμαι	Ναι
Ελεονόρα Ζουγανέλη	Κόψε και μοίρασε	Όχι
Παντελής Παντελίδης	Αλκοολικές οι νύχτες	Όχι
Μαρινέλα	Εσύ ποτέ	Ναι
Μίλτος Πασχαλίδης	Ακροβάτης	Ναι
Στέλιος Ρόκος	Άγγελοι είμαστε όλοι	Όχι
Αντύπας	Άραγε	Όχι
Απόστολος Ρίζος	Τι να θυμηθώ	Όχι
Σπύρος Ζαγοραίος	Προσευχή	Όχι
Ντομένικα	Άχρηστα ρολόγια	Όχι
Active Member	Αύριο	Όχι
Δημήτρης Υφαντής	Καίγομαι και σιγολιώνω	Όχι
Βασίλης Τσιτσάνης	Με παρέσυρε το ρέμα	Ναι

Πίνακας 2 : Αναγνώριση τραγουδιών ανά κατηγορία

Rock& metal

Σε αυτή τη κατηγορία απέτυχαν να αναγνωριστούν 9 κομμάτια κάτι που δίνει ποσοστό επιτυχίας 64%. Οι πιθανοί λόγοι αποτυχίας των κομματιών που δεν αναγνωρίστηκαν είναι κυρίως η εμπορικότητα του συγκροτήματος ή του καλλιτέχνη όπως επίσης και η ποιότητα της ηχογράφησης από την εφαρμογή με βασικούς λόγους την κακή ποιότητα του τραγουδιού, τον θόρυβο βάθους και την κακή ηχογράφηση λόγω του είδους του τραγουδιού. Σε μια περίπτωση είδαμε ότι ενώ υπήρξε αναγνώριση του τραγουδιού δεν υπήρξε η αναγνώριση του καλλιτέχνη κάτι που οδηγεί στο συμπέρασμα ότι πιθανότητα η προσθήκη του στην βάση δεδομένων ήταν ελλιπής.



Εικόνα 13 : Ποσοστά επιτυχίας αναγνώρισης κατηγορίας *Rock & metal*

Commercial

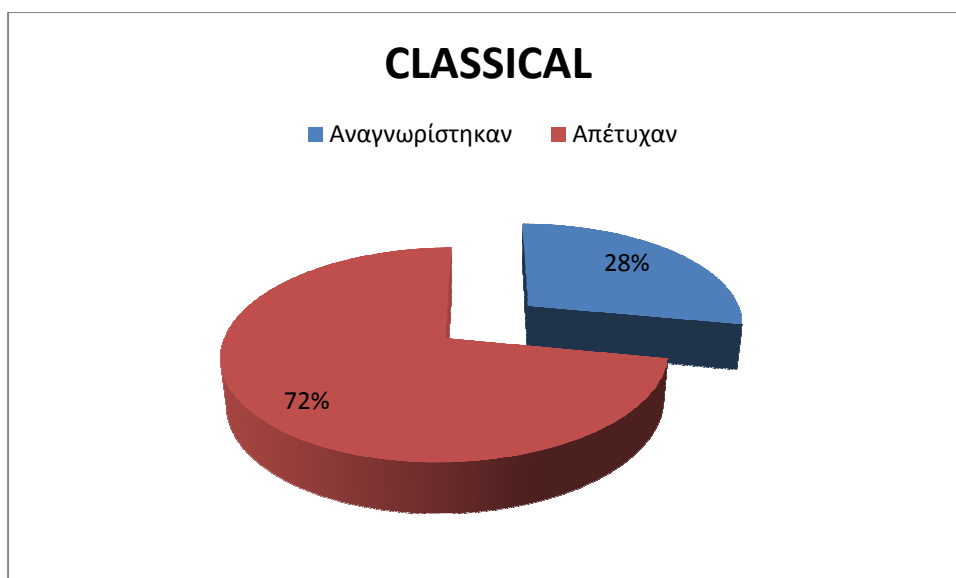
Σε αυτή τη κατηγορία απέτυχαν να αναγνωριστούν 10 κομμάτια κάτι που δίνει ποσοστό επιτυχίας 60%. Οι πιθανοί λόγοι αποτυχίας σε αυτή τη κατηγορία είναι η ποιότητα της ηχογράφησης μέσα από την εφαρμογή κυρίως λόγω του είδους της μουσικής, η εμπορικότητα του συγκεκριμένου τραγουδιού και η πιθανότητα επανεκτέλεσης ή όπως βλέπουμε στην τελευταία περίπτωση να πρόκειται για κάποιο remix το οποίο δεν είναι καταχωρημένο στη βάση δεδομένων.



Εικόνα 14 : Ποσοστά επιτυχίας αναγνώρισης κατηγορίας *Commercial*

Classical

Σε αυτή τη κατηγορία υπήρξε το μεγαλύτερο ποσοστό αποτυχίας με 18 κομμάτια να μη μπορούν να αναγνωριστούν. Αυτό δίνει ένα ποσοστό επιτυχίας του 28%. Οι πιο πιθανοί λόγοι της αποτυχίας είναι κυρίως η εμπορικότητα του συγκεκριμένου είδους, η ποιότητα της ηχογράφησης από την εφαρμογή λόγω του ίδιου του είδους της μουσικής. (κακή ποιότητα ήχου λόγω παλαιότητας εκτέλεσης, σημεία με χαμηλές δυναμικές) και τέλος το ότι τα μουσικά κομμάτια του συγκεκριμένου είδους τις περισσότερες φορές έχουν πολλούς διαφορετικούς εκτελεστές και βρίσκονται σε πολλές διαφορετικές συλλογές. Στην περίπτωση του *Vivaldi- Spring* όπως φαίνεται και στον πίνακα καταφέραμε να ανακτήσουμε πληροφορίες για τον εκτελεστή και τον δίσκο στον οποίο βρίσκεται το κομμάτι.



Εικόνα 14: Ποσοστά επιτυχίας αναγνώρισης κατηγορίας *Classical*

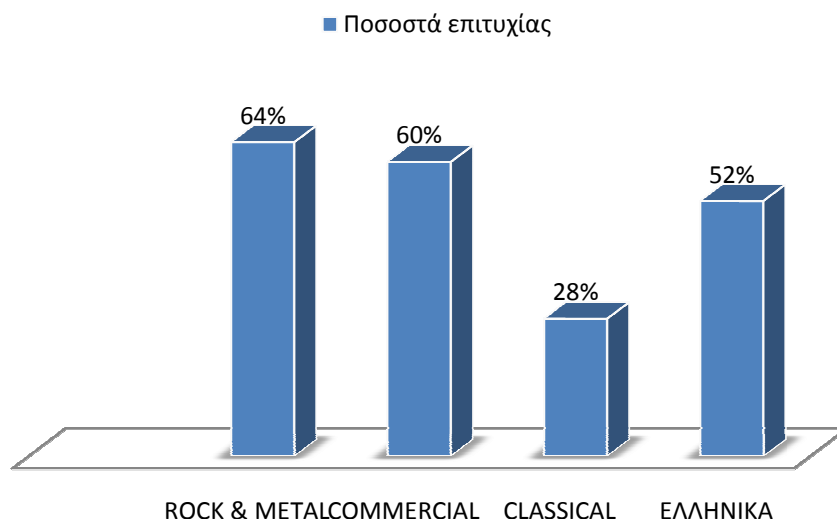
Ελληνικά

Σε αυτή τη κατηγορία απέτυχαν να αναγνωριστούν 12 κομμάτια κάτι που δίνει ποσοστό επιτυχίας 52%. Οι πιο πιθανοί λόγοι σε αυτή την κατηγορία για την αποτυχία αναγνώρισης είναι η εμπορικότητα των συγκεκριμένων κομματιών ή καλλιτεχνών καθώς και η ποιότητα της ηχογράφησης μέσω του μικρόφωνου.



Εικόνα 15 : Ποσοστά επιτυχίας αναγνώρισης κατηγορίας Ελληνικά

Ανάμεσα στις τέσσερις κατηγορίες είναι εμφανές ότι το μεγαλύτερο ποσοστό επιτυχίας το είχαν οι κατηγορίες Rock & metal και commercial, κυρίως λίγο της απήχησης που έχουν στο ευρύ κοινό παγκοσμίως και της εμπορικότητας τους. Στην εικόνα 16 απεικονίζονται συνολικά οι τέσσερις κατηγορίες και τα ποσοστά επιτυχίας που είχαν κατά την αναγνώριση.



Εικόνα 16 : Σύγκριση ποσοστών επιτυχίας των κατηγοριών

Αυτό που κρατάμε από τους παραπάνω πίνακες και τα ποσοστά αναγνώρισης είναι κυρίως το γιατί η αναγνώριση μπορεί να αποτύχει. Οι δυο σημαντικότεροι λόγοι είναι 1) η ποιότητα της ηχογράφησης από το μικρόφωνο, η οποία μπορεί να δημιουργεί προβλήματα αν για παράδειγμα το αρχείο που αναπαράγουμε δεν έχει το ίδιο την απαραίτητη ποιότητα ή αν υπάρχει μεγάλος θόρυβος βάθους. 2) Το τραγούδι ή ο καλλιτέχνης που προσπαθεί ο χρήστης να αναγνωρίσει να μην είναι ευρύτερα γνωστά και εμπορικά, άρα και να μην έχουν προστεθεί στη βάση δεδομένων που η συγκεκριμένη εφαρμογή χρησιμοποιεί. Είναι πιθανό επίσης να μην επιστραφούν όλες οι πληροφορίες που ο χρήστης περιμένει, καθώς η βάση δεδομένων δεν είναι επαρκώς ενημερωμένη όσον αφορά τα συγκεκριμένα τραγούδια ή τους καλλιτέχνες.

2.1. Διασκευές (covers)

Μια διασκευή ορίζεται ως η απόδοση ενός μουσικού κομματιού από διαφορετικούς καλλιτέχνες από αυτόν που το έχει εκτελέσει αρχικά ή από τον ίδιο καλλιτέχνη εκτελεσμένο διαφορετικά. Όταν διαφορετικοί μουσικοί ή μπάντες εκτελούν τα ίδια τραγούδια πολλές φορές αλλάζουν το tempo, το στυλ, την ενορχήστρωση του τραγουδιού, την τονικότητα ή την γλώσσα των στίχων. Πάντα όμως παραμένουν αρκετά στοιχεία του μελωδικού και αρμονικού περιεχομένου ώστε να διατηρείται η ταυτότητα του κομματιού. Υπάρχουν πολλές διαφορετικές κατηγορίες διασκευής ενός τραγουδιού. Οι πιο κοινές από αυτές είναι:

- Ορχηστρική: Εκδοχές κατά τις οποίες έχουν αφαιρεθεί οι στίχοι και έχει παραμείνει μόνο το ορχηστρικό κομμάτι.
- Ζωντανή εκτέλεση: Η ζωντανή εκτέλεση και ηχογράφηση ενός τραγουδιού από τον αρχικό εκτέλεση ή κάποιον άλλον καλλιτέχνη.
- Ακουστική εκτέλεση: Το τραγούδι εκτελείται με, ακουστικά όργανα σε μια πιο απαλή και οικεία εκτέλεση.

- **Remix:** Ένα remix είναι η δημιουργία ενός τραγουδιού αλλάζοντας κάποια σημαντικά χαρακτηριστικά του αρχικού κομματιού, προσθέτοντας νέα ή συνδυάζοντας τα μέρη διαφορετικών τραγουδιών.

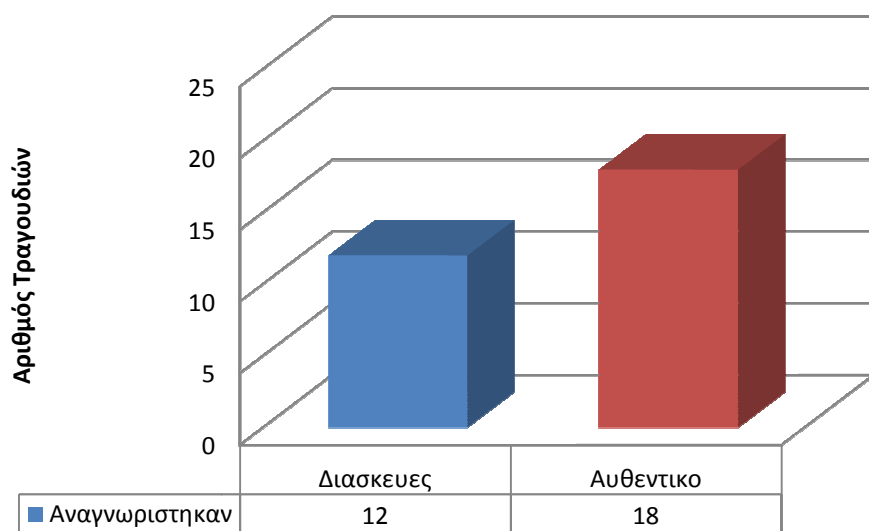
Η αυτόματη αναγνώριση διασκευών μέσα από μια συλλογή είναι μια μάλλον δύσκολη διαδικασία. Οι εφαρμογές για αναγνώριση διασκευών περιλαμβάνουν αυτόματη μουσική αναγνώριση, ανίχνευση παραβίασης πνευματικών δικαιωμάτων, και μουσικολογική ερεύνα. Ένα σημαντικό έργο στην αναγνώριση διασκευών είναι ο ποσοτικός προσδιορισμός μουσικών ζευγών και σύγκριση ομοιότητας μεταξύ τραγουδιών, ώστε να προσδιοριστεί αν τα τραγούδια αυτά μπορούν να θεωρηθούν διασκευές μεταξύ τους. Έχει προταθεί ένας αριθμός μεθόδων για την αναπαράσταση και μοντελοποίηση μουσικών σημάτων για αναγνώριση διασκευών. Μέθοδοι όπως τεχνικές ευθυγράμμισης, ανάλυση, cross-correlation, cross-recurrence analysis και time series prediction. Όλα αυτά προϋποθέτουν ότι ένας αλγόριθμος αναγνωρίζει για ένα ερώτημα ενός ηχητικού αποσπάσματος, άλλες ηχογραφήσεις παρόμοιας σύνθεσης ή διασκευών.

Η πιο συνηθισμένη διαδικασία για την μέτρηση ομοιότητας μεταξύ δυο διασκευών είναι να εκμεταλλευτούν τα χαρακτηριστικά που μπορεί να μοιράζονται μεταξύ τους. Όπως αναφέρθηκε παραπάνω τα κοινά χαρακτηριστικά που υπόκεινται σε μεταβολή είναι πολλά και τα συστήματα αναγνώρισης διασκευών πρέπει να είναι εύρωστα απέναντι σε αυτές τις μεταβολές. Τα στοιχεία των υπαρχόντων εφαρμογών για αναγνώριση διασκευών μπορούν να χωριστούν σε τέσσερις βασικές κατηγορίες: εξαγωγή χαρακτηριστικών, σταθερότητα μουσικού κλειδιού, σταθερότητα ρυθμού, σταθερότητα δομής. Ένας άλλος δρόμος προς την αναγνώριση είναι η ανίχνευση μουσικών αναφορών. Στην κλασική μουσική υπάρχουν πολλοί συνθέτες οι οποίοι δανείζονται μουσικές φράσεις ή μοτίβα από άλλους συνθέτες. Στην pop υπάρχουν επίσης πολλές αναφορές και στην ηλεκτρονική μουσική υπάρχει ένας μεγάλος δανεισμός αποσπασμάτων από μουσικά κομμάτια για την δημιουργία άλλων. Επειδή πρόκειται για μικρής διάρκειας αποσπάσματα οι αλγόριθμοι θα πρέπει να είναι προσαρμοσμένοι ώστε να μπορούν να τα εντοπίσουν. Είναι πολύ πιθανό βέβαια οι διασκευές και όλα τα στοιχεία τους να υπάρχουν ήδη στην βάση δεδομένων από την οποία αντλούμε πληροφορίες σχετικά με το κομμάτι που είναι να αναγνωριστεί. Οι πληροφορίες που θα επιστραφούν είναι πιθανόν έκτος από τις πληροφορίες για τον εκτελεστή της διασκευής να δίνουν και τις αντίστοιχες πληροφορίες για την αρχική εκτέλεση του κομματιού.

Παρακάτω παρουσιάζεται μια λίστα με διασκευές γνωστών τραγουδιών και το αν υπήρξε αναγνώριση από την Web εφαρμογή. Ταυτόχρονα έγινε αναγνώριση και στις αρχικές εκτελέσεις τους και σύγκριση μεταξύ τους. Τα τραγούδια επιλέχτηκαν μέσα από τον σκληρό δίσκο.

Μπάντα/Καλλιτέχνης	Τραγούδι	Αρχικός εκτελεστής	Επιτυχία αναγνώρισης	Αναγνώριση από αρχικό εκτελεστή
The wallflowers	We can be heroes	David Bowie	Όχι	Ναι
Sentenced	White wedding	Billy Idol	Ναι	Όχι
Lions	Girl from the north country	Bob Dylan	Ναι	Ναι
Battleme	Hey hey, my my	Neil Young	Ναι	Όχι
Mona	Stand by me	Ben E. King	Όχι	Ναι
Richard Cheese	Smells like teen spirit	Nirvana	Όχι	Ναι
Scala&Colansky brothers	Viva la vida	Coldplay	Ναι	Όχι
Feist	Limit to your love	James Blake	Ναι	Ναι
Coeur de piratte	Aint no sunshine	Bill Withers	Όχι	Ναι
Jeff Buckley	Hallelujah	Leonard Cohen	Ναι	Ναι
Blind Guardian	To France	Mike Oldfield	Ναι	Ναι
Nosa alma canta	Should I stay or should I go	The clash	Όχι	Ναι
Beth Hart	I'd rather go blind	Etta James	Όχι	Όχι
Katey Sagal	Bird on a wire	Leonard Cohen	Όχι	Ναι
Nouvele Vag	Just can't get enough	Depeche mode	Ναι	Ναι
London grammar	Wicked game	Chris Isaak	Όχι	Ναι
Patti Smith	Gimme shelter	Rolling stones	Ναι	Ναι
Adele	Lovesong	The cure	Όχι	Όχι
Peggy Lee	Hard days night	The beatles	Όχι	Ναι
Alpha blondy	Wish you were here	Pink Floyd	Όχι	Ναι
Johny Cash	One	U2	Ναι	Ναι
Marilyn Manson	Personal Jesus	Depeche mode	Ναι	Ναι
Santana	While my guitar genlty weeps	The beatles	Ναι	Ναι

Εικόνα 17 : Σύγκριση αναγνώρισης διασκευών με αυθεντικές εκτελέσεις



Εικόνα 18 : Σύγκριση της επιτυχίας αναγνώρισης μεταξύ διασκευών και αυθεντικών

Όπως είναι εμφανές από τον πίνακα και το γράφημα 6 στο κομμάτι των διασκευών δεν υπήρχε μεγάλο ποσοστό αναγνωρίσεων. Το ποσοστό ήταν 48% με 12 αναγνωρίσεις. Αν προσέξει κάποιος τον πίνακα μπορεί να διακρίνει ότι οι διασκευές που δεν αναγνωρίστηκαν ανήκουν κυρίως σε καλλιτέχνες που δεν είναι αρκετά γνωστοί ή εμπορικοί. Αυτό οδηγεί στο συμπέρασμα ότι ίσως να μην υπάρχουν στην βάση δεδομένων. Η βάση δεδομένων παίζει μεγάλο ρόλο καθότι το Doreso σαν μια νέα υπηρεσία που τώρα αναπτύσσεται δεν έχει αναπτύξει τους εύρωστους και πολύπλοκους αλγόριθμους που χρειάζονται για αναγνώριση διασκευών, όπως επίσης και δεν διαθέτει τον αριθμό των τραγουδιών στην βάση δεδομένων του που διαθέτουν άλλες βάσεις όπως για παράδειγμα του musicbrainz. Στις αυθεντικές εκτελέσεις όπως ήταν αναμενόμενο υπήρξε μεγαλύτερη επιτυχία στις αναγνωρίσεις των κομματιών με 18 κομμάτια από τα 25 και με ένα ποσοστό του 72%. Μεγάλο ρόλο στην επιτυχία ή αποτυχία των αναγνωρίσεων μέσα από τον σκληρό δίσκο παίζει η ποιότητα του αρχείου και ο τρόπος που αποκτήθηκε. Για παράδειγμα αν έχει γίνει rip από ένα cd ή αν έχει αποκτηθεί μέσω download από το διαδίκτυο. Πολλές φορές τα αρχεία έχουν υποστεί μεγάλες αλλοιώσεις και αυτό δημιουργεί πρόβλημα στην αναγνώριση. Να τονιστεί επίσης ότι όσο καλύτερη η ποιότητα του αρχείου τόσο μεγαλύτερο και το μέγεθος του. Αυτό όπως είναι λογικό αυξάνει και τον χρόνο που χρειάζεται να επιστραφούν οι πληροφορίες σχετικά με το τραγούδι καθώς αυξάνει ο χρόνος που χρειάζεται για να ανέβει.

1.3. Συμπεράσματα

Τα συμπεράσματα που βγαίνουν από το πειραματικό μέρος αφορούν αρχικά την δημιουργία εφαρμογών αναγνώρισης και ανάκτησης δεδομένων. Η ανάπτυξη του συγκεκριμένου τομέα και του διαδικτύου έχουν κάνει πολύ πιο εύκολη την δημιουργία τέτοιων εφαρμογών σήμερα από ότι ήταν πριν κάποια χρόνια. Με την κατάλληλη αναζήτηση ακόμα και ένας μη ιδιαίτερα έμπειρος προγραμματιστής

μπορεί να βρει τα κατάλληλα εργαλεία ώστε να δημιουργήσει την εφαρμογή που θέλει. Σε παλαιότερες εποχές θα έπρεπε να καταφύγει στην συγγραφή κώδικα κάτι που φυσικά είναι πολύ χρονοβόρο. Αυτό βεβαίως δεν σημαίνει ότι και σήμερα δεν είναι απαραίτητο καθώς μπορεί τα εργαλεία να υπάρχουν και να δίνουν την δυνατότητα να χρησιμοποιηθούν ελεύθερα, όμως σε πολλές περιπτώσεις θα πρέπει να τροποποιηθούν ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα. Συνδυάζοντας λοιπόν τα κατάλληλα εργαλεία ανάλογα την εφαρμογή που επιθυμεί κανείς να φτιάξει με τις ανάλογες online υπηρεσίες, ο προγραμματιστής μπορεί να δημιουργήσει ακριβώς την εφαρμογή που ζητεί και να την προσαρμόσει ανάλογα με το τι θέλει αυτή να κάνει. Η επιτυχία των εφαρμογών αυτών εξαρτάται άμεσα από τις υπηρεσίες που θα χρησιμοποιηθούν (Echonest, Musicbrainz, Doreso κλπ) και το τι επιλογές προσφέρουν. Όπως είναι λογικό όσο μεγαλύτερη και διαδεδομένη η εταιρία που προσφέρει τις υπηρεσίες τόσο μεγαλύτερη ακρίβεια θα έχουν οι εφαρμογές που θα δημιουργηθούν και ταυτόχρονα αυξάνεται η αξιοπιστία τους. Για μεγαλύτερη ακρίβεια και αξιοπιστία φυσικά μπορεί να γίνει και συνδυασμός αυτών των υπηρεσιών.

Η αξιοπιστία και η επιτυχία των εφαρμογών εξαρτώνται από πολλούς παράγοντες. Όπως αναφέρθηκε παραπάνω ένας παράγοντας είναι τα ίδια τα εργαλεία που χρησιμοποιήθηκαν για την δημιουργία τους. Μεγάλο ρόλο παίζει το ίδιο το αρχείο που ζητείται η αναγνώριση του. Η ποιότητα του αρχείου και το μέγεθος του όταν πρόκειται για ηχογράφηση είναι δυο πολύ βασικοί παράγοντες. Συνήθως όταν πρόκειται για αρχεία που έχουν προέρθει από ηχογραφήσεις πολύ μεγάλο ρόλο παίζει και το περιβάλλον στο οποίο έγινε η ηχογράφηση. Μεγάλος θόρυβος βάθους ή, αλλοιωμένη ηχογράφηση λόγω της έντασης του ήχου είναι οι δυο μεγαλύτεροι παράγοντες που μπορούν να επηρεάσουν το αρχείο. Τέλος η μη εμπορικότητα ενός τραγουδιού ή καλλιτέχνη είναι ένας παράγοντας που κυρίως για μικρές βάσεις δεδομένων παίζει μεγάλο ρόλο, καθώς μπορεί να μην έχει συμπεριληφθεί σε αυτές. Σε κάποιες περιπτώσεις, όπως συμβαίνει για παράδειγμα με τις διασκευές τραγουδιών, ο καλλιτέχνης που κάνει την εκτέλεση είναι εντελώς άγνωστος ή ερασιτέχνης και δεν υπάρχει μέσα σε κάποια βάση δεδομένων. Αν η εφαρμογή δεν τρέχει αλγόριθμους που να ειδικεύονται στην αναζήτηση και αναγνώριση παρόμοιων κομματιών τότε είναι πολύ δύσκολο να υπάρξει επιτυχές αποτέλεσμα.

Η εφαρμογή που δημιουργήθηκε για τις ανάγκες αυτής της πτυχιακής χρησιμοποίησε τις βάσεις δεδομένων από δυο διαφορετικές υπηρεσίες με την κεντρική και βασική να είναι αυτή του Doreso και έναν συνδυασμό των εργαλείων που προσφέρουν (APIs) με πρόσθετα εργαλεία (plugins) όπως το recorder.js. Από άποψη λειτουργίας και επιτυχίας δεν είναι όσο δυνατή όσο οι πιο διάσημες εφαρμογές που κυκλοφορούν, είναι όμως μια εφαρμογή που μπορεί να δώσει αποτελέσματα από ένα εύρος μουσικών ειδών και με ένα ποσοστό επιτυχίας άνω του 50% για τα περισσότερα είδη μουσικής. Όπως φάνηκε στο γράφημα 5 δεν δίνει υψηλά ποσοστά επιτυχίας από ηχογράφηση μέσω αέρα με το μεγαλύτερο να είναι αυτό που ανήκει στην κατηγορία Rock & Metal με 64%. Το υψηλότερο ποσοστό φάνηκε στην σύγκριση των διασκευών με τα αυθεντικά κομμάτια, με την επιλογή τους να γίνεται μέσα από τον σκληρό δίσκο. Το ποσοστό επιτυχίας ανέβηκε στο 72% κάτι που δείχνει ότι η συγκεκριμένη εφαρμογή έχει μεγαλύτερη πιθανότητα επιτυχίας όταν πρόκειται να αναγνωρίσει τραγούδια τα οποία δεν έχουν ηχογραφηθεί από κάποιο μικρόφωνο ή άλλη συσκευή. Η ταχύτητα αναγνώρισης και ανάκτησης δεδομένων

εξαρτάται από το μέγεθος του αρχείου που γίνεται upload και το μέγεθος της βάσης δεδομένων. Στην συγκεκριμένη εφαρμογή ο χρόνος που χρειάζεται από τη στιγμή που θα πατηθεί το κουμπί identify μέχρι τη στιγμή που θα επιστραφούν οι πληροφορίες κυμαίνεται από 20 μέχρι 30 δευτερόλεπτα, ανάλογα το μέγεθος του αρχείου. Για παράδειγμα για το αρχείο *3 Doors Down - Here By Me.mp3* με μέγεθος 5,8 MB χρειάστηκαν 23 δευτερόλεπτα και υπήρξε επιτυχής αναγνώριση. Ο περισσότερος χρόνος καταναλώνεται στο ανέβασμα του αρχείου αφού ο χρόνος που χρειάζεται για την αναζήτηση στη βάση δεδομένων και την ανάκτηση των πληροφοριών είναι μόλις 7 δευτερόλεπτα.

Κεφάλαιο 5

Επίλογος

Στην παρούσα πτυχιακή εργασία αναλύθηκε το πρόβλημα της ανάκτησης μουσικών πληροφοριών από βάσεις δεδομένων. Ξεκίνησε από την βάση του συγκεκριμένου κλάδου (ανάκτηση πληροφοριών), η οποία είναι η μηχανική μάθηση και επεκτάθηκε στις δυο πιο βασικές κατηγορίες όσον αφορά τη μουσική και τον ήχο. Δηλαδή το music information retrieval και το audio fingerprinting. Αναλύθηκαν όλα τα χαρακτηριστικά αυτών των δυο τομέων, καθώς και ο τρόπος λειτουργίας τους και παρουσιάστηκαν παραδείγματα βάσεων δεδομένων και εφαρμογών ανοιχτών προς το ευρύ κοινό. Στην συνέχεια επιχειρήθηκε η δημιουργία μιας web εφαρμογής η οποία θα μπορεί να ανατρέξει σε μια από αυτές τις βάσεις δεδομένων και με τον κατάλληλο έλεγχο να επιστρέψει τις πληροφορίες σχετικά το αρχείο ήχου για το οποίο θα κάνει την αναζήτηση ο χρήστης. Η διαδικασία ήταν σχετικά απλή με το μόνο βασικό πρόβλημα να είναι η αναζήτηση νέας βάσης δεδομένων που να προσφέρει ανοιχτό τις υπηρεσίες της όταν η πρώτη έπαψε την λειτουργία της. Η δημιουργία της εφαρμογής ήταν επιτυχής και η χρήση της είναι εξαιρετικά απλή.

Η μουσική είναι και ήταν από αρχαίων χρόνων αναπόσπαστο κομμάτι της ανθρωπινής ζωής και δραστηριότητας. Όσο η τεχνολογία προοδεύει και ακμάζει τόσο ανοίγουν και οι ορίζοντες από άποψη επιστημονική και ψυχαγωγική. Οι άνθρωποι πάντα θα ψάχνουν ευκαιρίες και τρόπους ώστε να κάνουν την εμπειρία της μουσικής πιο ζωντανή, πιο αληθινή και πιο απολαυστική. Η τεχνολογία και η ερευνά τους έχουν δώσει ήδη τους τρόπους αυτούς και έχουν οδηγήσει την τέχνη σε νέους δρόμους και άλλα επίπεδα που δεν μπορούσαμε να φανταστούμε παλιότερα. Υπάρχουν ακόμα πολλά να ανακαλυφθούν, να δημιουργηθούν και νέοι ορίζοντες να εξερευνηθούν.

Βιβλιογραφία

1. **Alpaydm, Ethem.** *Introduction to Machine Learning*. London : The MIT Press, 2010.
2. **Tao Li, Mitsunori Ogihara, Bo Shao and DingdingWang.** Machine Learning Approaches for Music Information Retrieval. [συγγρ. βιβλίου] Meng Joo Er and Yi Zhou. *Theory and Novel Applications of Machine Learning*. s.l. : InTech, 2009, σ. 376.
3. **Learning, Machine.** *Tom M. Mitchell*. Portland : McGraw-Hill Science/Engineering/Math, 1997.
4. **Nilsson, Nils J.** *INTRODUCTION TO MACHINE LEARNING - AN EARLY DRAFT OF A PROPOSED TEXTBOOK*. Stanford : s.n., 1998.
5. **Francesco Camastra, Alessandro Vinciarelli.** *Machine Learning for Audio, Image and Video Analysis*. s.l. : Springer, 2007.
6. **Ratsch, Gunnar.** A Brief Introduction into Machine Learning. Tübingen, Germany : Friedrich Miescher Laboratory of the Max Planck Society.
7. **Domingos, Pedro.** A Few Useful Things to Know about Machine Learning. Seattle U.S.A. : s.n.
8. **DAVID W. AHA, DENNIS KIBLER, MARC K. ALBERT.** Instance-Based Learning Algorithms. s.l., The Netherlands. : J.R. Quinlan, 1991. σ. 30.
9. **William J. Frawley, Gregory Piatesky-Shapiro, and Christopher J. Matheus.** Knowledge Discovery in Databases: An Overview. *AI Magazine Volume 13 Number 3*. 1992.
10. **Roger B. Dannenberg, Belinda Thom, David Watson.** A Machine Learning Approach to Musical Style Recognition. 1997.
11. **Ayodele, Taiwo Oladipupo.** Types of Machine Learning Algorithms. [συγγρ. βιβλίου] Yagang Zhang. *New Advances in Machine Learning*. s.l. : InTech, 2010, σ. 374.
12. **Jensen, Jesper Højvang.** Feature Extraction for Music Information Retrieval. Aalborg Ø, Denmark : s.n., 2009.
13. **Thoshkahna, Balaji.** Algorithms for Music Information Retrieval. Bangalore : s.n., April 2006.
14. **J. Stephen Downie, Sally Jo Cunningham.** *Toward a Theory of Music Information Retrieval Queries: System Design Implications*. Hamilton, New Zealand : s.n., June 2002.
15. **Michael A. Casey, Remco Veltkamp, Masataka Goto, Marc Leman, Christophe Rhodes, and Malcolm Slaney.** Content-Based Music Information Retrieval: Current Directions and Future Challenges. *Proceedings of the IEEE | Vol. 96, No.* April 2008.
16. **Orio, Nicola.** Music Retrieval: A Tutorial and Review. *Foundations and Trends in Information Retrieval Vol. 1, No 1*. November 2006.

17. **Rainer Typke, Frans Wiering, Remco C. Veltkamp.** A SURVEY OF MUSIC INFORMATION RETRIEVAL SYSTEMS. Utrecht, The Netherlands 2005.
18. **Downie, J. Stephen.** The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoust. Sci. & Tech.* 29, 4. 2008.
19. **Müller, M.** *Information Retrieval for Music and Motion.* Berlin : Springer-Verlag, 2007.
20. **Meurer, Will.** *Aspects of Music Information Retrieval.* Texas : s.n.
21. **Larsen, Vegard Andreas.** Combining Audio Fingerprints. June 2008.
22. **Palmer, Nick.** Review of audio fingerprinting algorithms and looking into a multi-faceted approach to fingerprint generation. May 2011.
23. **Wang, Avery Li-Chun.** An Industrial-Strength Audio Search Algorithm. 29 October 2003.
24. **Porter, Alastair.** Evaluating musical fingerprinting systems. Montreal, Canada : s.n., December 2012.
25. **Doets, Peter Jan Otto.** Modeling Audio Fingerprints: Structure, Distortion, Capacity. Amsterdam, The Netherlands : s.n., 20 october 2010.
26. **Heinrich A. van Nieuwenhuizen, Willie C. Venter and Leenta M.J. Grobler.** Comparison of Algorithms for Audio Fingerprinting.
27. **PEDRO CANO, ELOI BATLLE, TON KALKER, JAAP HAITSMAS.** A Review of Audio Fingerprinting. *Journal of VLSI Signal Processing* 41,, 2005.
28. **Matthew L. Miller, Manuel Acevedo Rodriguez.** Audio Fingerprinting: Nearest Neighbor Search in High Dimensional Binary Spaces. London : s.n.
29. **Jaap Haitsma, Ton Kalker.** A Highly Robust Audio Fingerprinting System. Eindhoven, The Netherlands : s.n.
30. **Pedro Cano, Eloi Batlle, Emilia Gómez, Leandro de C.T.Gomes, and Madeleine Bonnet.** Audio Fingerprinting: Concepts And Applications. [συγγρ. βιβλίου] P. Cano et al. *Audio Fingerprinting: Concepts And Applications, Studies in Computational Intelligence.* Berlin : Springer-Verlag, 2005.
31. **Qingmei Xiao, Motoyuki Suzuki, Kenji Kita.** FAST HAMMING SPACE SEARCH FOR AUDIO FINGERPRINTING SYSTEMS. *12th International Society for Music Information Retrieval Conference (ISMIR 2011).* Tokushima, Japan : s.n., 2011.
32. **Daniel P.W. Ellis, Brian Whitman, Alastair Porter.** ECHOPRINT - AN OPEN MUSIC IDENTIFICATION SERVICE. s.l. : International Society for Music Information Retrieval, 2011.
33. **Jerome Lebosse, Luc Brun, Jean Claude Pailles.** A ROBUST AUDIO FINGERPRINT EXTRACTION ALGORITHM. Caen, France : s.n.

34. **Shumeet Baluja, Michele Covell.** Content Fingerprinting Using Wavelets. s.l. : Google, Inc.
35. **CHING-HUA CHUAN, ELAINE CHEW.** AUDIO ONSET DETECTION USING MACHINE LEARNING TECHNIQUES: THE EFFECT AND APPLICABILITY OF KEY AND TEMPO INFORMATION. Los Angeles, California, USA : s.n.
36. **Tao Li and Mitsunori Ogiwara.** Towards Intelligent Music Information Retrieval.
37. **Cano, Pedro.** CONTENT-BASED AUDIO SEARCH: FROM FINGERPRINTING TO SEMANTIC AUDIO RETRIEVAL. 2006.
38. **Gravell, Dan.** <http://www.blisshq.com/>. [Ηλεκτρονικό] 21 August 2012.
<http://www.blisshq.com/music-library-management-blog/2012/08/21/what-is-audio-fingerprinting/>.
39. [github.com. *GitHub*](https://github.com/hbredin/PyAFE/wiki). [Ηλεκτρονικό] 3 January 2012.
<https://github.com/hbredin/PyAFE/wiki>.
40. [blog.echonest.com. *Echonest*](http://blog.echonest.com). [Ηλεκτρονικό] 24 April 2010.
<http://blog.echonest.com/post/545323349/the-echo-nest-musical-fingerprint-enmf>.
41. [echoprint.me/. *Echoprint*](http://echoprint.me/). [Ηλεκτρονικό] <http://echoprint.me/>.
42. [wiki.musicbrainz.org/Fingerprinting. *Musicbrainz*](http://wiki.musicbrainz.org/Fingerprinting). [Ηλεκτρονικό] 19 August 2014.
<http://wiki.musicbrainz.org/Fingerprinting>.
43. [github.com/echonest/echoprint-codegen. *github*](https://github.com/echonest/echoprint-codegen). [Ηλεκτρονικό]
<https://github.com/echonest/echoprint-codegen>.
44. [developer.echonest.com. *exhonest*](http://developer.echonest.com). [Ηλεκτρονικό]
<http://developer.echonest.com/docs/v4/song.html#identify>.
45. **Willis, Nathan.** [lwn.net. *lwn*](https://lwn.net/Articles/449650/). [Ηλεκτρονικό] 29 June 2011.
<https://lwn.net/Articles/449650/>.
46. **Jones, Richard.** [blog.last.fm. *lastfm*](http://blog.last.fm). [Ηλεκτρονικό] 29 August 2007.
<http://blog.last.fm/2007/08/29/audio-fingerprinting-for-clean-metadata>.
47. [acoustid.org/chromaprint. *acoustid*](https://acoustid.org/chromaprint). [Ηλεκτρονικό] <https://acoustid.org/chromaprint>.
48. [echonest.github.io/pyechonest/. *echonest*](http://echonest.github.io/pyechonest/). [Ηλεκτρονικό]
<http://echonest.github.io/pyechonest/>.
49. [musicmachinery.com. *musicmachinery*](http://musicmachinery.com). [Ηλεκτρονικό]
<http://musicmachinery.com/2011/06/25/finding-duplicate-songs-in-your-music-collection-with-echoprint/>.
50. [wikipedia.org/wiki/Support_vector_machine. *wikipedia*](http://en.wikipedia.org/wiki/Support_vector_machine). [Ηλεκτρονικό] 4 May 2015.
http://en.wikipedia.org/wiki/Support_vector_machine.

51. en.wikipedia.org/wiki/Acoustic_fingerprint. *wikipedia*. [Ηλεκτρονικό] 3 January 2015.
http://en.wikipedia.org/wiki/Acoustic_fingerprint.
52. en.wikipedia.org/wiki/Machine_learning#Algorithm_types. *wikipedia*. [Ηλεκτρονικό] 15 May 2015. http://en.wikipedia.org/wiki/Machine_learning#Algorithm_types.
53. [ee.columbia.edu/ln/rosa/matlab/audfprint/](http://www.ee.columbia.edu/ln/rosa/matlab/audfprint/). *columbia*. [Ηλεκτρονικό]
<http://www.ee.columbia.edu/ln/rosa/matlab/audfprint/>.
54. forum.kodi.tv/showthread.php?tid=37230. *kodi*. [Ηλεκτρονικό]
<http://forum.kodi.tv/showthread.php?tid=37230>.
55. en.wikipedia.org/wiki/List_of_online_music_databases. *wikipedia*. [Ηλεκτρονικό] 7 May 2015. http://en.wikipedia.org/wiki/List_of_online_music_databases.
56. developer.doreso.com/doc3.html. *doreso*. [Ηλεκτρονικό]
<http://developer.doreso.com/doc3.html>.
57. github.com/echonest/DeezerPlaylistDemo. *github*. [Ηλεκτρονικό]
<https://github.com/echonest/DeezerPlaylistDemo>.
58. github.com/lempere/Recorderjs. *github*. [Ηλεκτρονικό]
<https://github.com/lempere/Recorderjs>.
59. [oceanos.grnet.g. oceanos](https://oceanos.grnet.gr/home/). [Ηλεκτρονικό] <https://oceanos.grnet.gr/home/>.
60. developer.echonest.com/sandbox/deezer.html. *echonest*. [Ηλεκτρονικό]
<http://developer.echonest.com/sandbox/deezer.html>.
61. **Wiering, Frans**. Can Humans Benefit from Music Information Retrieval? Utrecht, Netherlands : s.n.
62. **Meurer, Will**. www.ischool.utexas.edu/~i385df04/StudentPPT/HTML/meurer_w/. [Ηλεκτρονικό]
https://www.ischool.utexas.edu/~i385df04/StudentPPT/HTML/meurer_w/.
63. [music-ir.org](http://www.music-ir.org). [Ηλεκτρονικό] <http://www.music-ir.org/evaluation/tools.html>.
64. **William Birmingham, Colin Meek, Kevin O'Malley, Bryan Pardo, and Jonah Shifrin**. [collaboration.cmc.ec.gc.ca](http://collaboration.cmc.ec.gc.ca/science/rpn/biblio/ddj/Website/articles/DDJ/2003/0309/0309f/0309f.htm). [Ηλεκτρονικό]
<http://collaboration.cmc.ec.gc.ca/science/rpn/biblio/ddj/Website/articles/DDJ/2003/0309/0309f/0309f.htm>.
65. teachingmir.wikispaces.com. [Ηλεκτρονικό]
<http://teachingmir.wikispaces.com/Datasets>.
66. <http://www.ismir.net/>. [Ηλεκτρονικό] <http://www.ismir.net/>.
67. [unf.edu](http://www.unf.edu/~c.chuan/cis4930/MIR_databases.pdf). [Ηλεκτρονικό] 2011.
http://www.unf.edu/~c.chuan/cis4930/MIR_databases.pdf.

68. **Ayodele, Taiwo Oladipupo.** Types of Machine Learning Algorithms. *New Advances in Machine Learning*. 2010.
69. wikipedia.org. [Ηλεκτρονικό] [http://en.wikipedia.org/wiki/Weka_\(machine_learning\)](http://en.wikipedia.org/wiki/Weka_(machine_learning)).
70. jmlr.org. [Ηλεκτρονικό] <http://jmlr.org/mloss/>.
71. en.wikipedia.org. [Ηλεκτρονικό]
http://en.wikipedia.org/wiki/List_of_machine_learning_concepts.
72. dmoz.org. [Ηλεκτρονικό]
http://www.dmoz.org/Computers/Artificial_Intelligence/Machine_Learning/Software/.
73. <http://flask.pocoo.org/>. [Ηλεκτρονικό] <http://flask.pocoo.org/>.
74. getbootstrap.com/. [Ηλεκτρονικό] <http://getbootstrap.com/>.
75. archive.ics.uci.edu. [Ηλεκτρονικό] <http://archive.ics.uci.edu/ml/index.html>.
76. cs.toronto.edu/~delve/data/datasets. *cs.toronto.edu*. [Ηλεκτρονικό]
<http://www.cs.toronto.edu/~delve/data/datasets.html>.
77. expdb.cs.kuleuven.be/expdb/. *expdb.cs.kuleuven.be*. [Ηλεκτρονικό]
<http://expdb.cs.kuleuven.be/expdb/>.
78. open-fp.sourceforge.net/. *open-fp.sourceforge*. [Ηλεκτρονικό] <http://open-fp.sourceforge.net/>.
79. **Wolchover, Natalie.** quantamagazine.org/20130723-as-machines-get-smarter-evidence-they-learn-like-us/. *quantamagazine.org*. [Ηλεκτρονικό] 23 July 2013.
<https://www.quantamagazine.org/20130723-as-machines-get-smarter-evidence-they-learn-like-us/>.

Παράρτημα

Παρακάτω εμφανίζονται οι κώδικες που χρησιμοποιήθηκαν για την κατασκευή της web εφαρμογής .

App.py

Το εργαλείο για τον έλεγχο της λειτουργίας του server.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import os
from flask import Flask, render_template, request, redirect, url_for,
send_from_directory, jsonify
from werkzeug import secure_filename
import json
import pydoreso
from pyechonest import song
from pyechonest import config
config.ECHO_NEST_API_KEY="ZYERT7D8IETCXB9NG"

# αρχικοποιεί την εφαρμογή Flask
app = Flask(__name__)

# το path για το upload directory
app.config['UPLOAD_FOLDER'] = 'uploads/'
# extension που επιτρέπεται να ανέβουν
app.config['ALLOWED_EXTENSIONS'] = set(['mp3', 'wav'])

# Επιστροφές που δέχεται ο server
def allowed_file(filename):
    return '.' in filename and \
           filename.rsplit('.', 1)[1] in
app.config['ALLOWED_EXTENSIONS']

#Αν απλά ζητηθεί το αρχικό url επιστρέφεται η συνάρτηση
#που επιστρέφει τον χρήστη στην αρχική σελίδα
@app.route('/')
def index():
    return render_template('index.html')

# Route που θα επεξεργαστεί το ανέβασμα του αρχείου
@app.route('/identify', methods=['POST'])
def upload():
    # πάρε το όνομα του αρχείου που ανέβηκε
    file = request.files['file']
    # ελέγχει αν το αρχείο έχει τη μορφοποίηση που επιτρέπεται
    if file and allowed_file(file.filename):
        # Μετατρέπει το όνομα σε ασφαλές
        filename = secure_filename(file.filename)
        # μεταφέρει το αρχείο από έναν προσωρινό φάκελο
        # στον upload φάκελο που ορίζουμε
        file.save(os.path.join(app.config['UPLOAD_FOLDER'],
filename))
```

```

        #δίνουμε το API key του doreso
        doreso =
pydoreso.Doreso('wqGyoPaBZHF2dBHpRsoijplwmH0qzF6t9kO6BeZ7LQE')
        #στελνει το αποθηκευμένο αρχείο για αναγνώριση
        d =
doreso.song_identify_file(os.path.join(app.config['UPLOAD_FOLDER'],
filename))
        #βρίσκει τον κωδικό που έχει το συγκεκριμένο τραγούδι στο
deezer μέσω του echonest
        if d["status"]:
            rkp_results =
song.search(artist=d["data"][0]["artist_name"],
title=d["data"][0]["name"],buckets=['id:deezer', 'tracks'],
limit=True)
            if rkp_results:
                echonest_res = rkp_results[0]
                track = echonest_res.get_tracks('deezer')[0]
                return render_template('result.html', d=d,
song=track)
            else:
                return render_template('result.html', d= None, song=
None)
        else:
            return render_template('result.html', d= None, song=
None)
        else:
            return render_template('result.html', d= None, song= None)

#βρίσκει όλες τις διαθέσιμες ip του server
#και απαντάει σε όλα τα requests που έρχονται από αυτές
if __name__ == '__main__':
    app.run(
        host="0.0.0.0",
        port=80
        debug=True
    )

```

Script.js

Το εργαλείο για τον έλεγχο της λειτουργίας της αρχικής σελίδας.

```

function __log(e, data) {
    log.innerHTML += "\n" + e + " " + (data || '');
}

var audio_context;
var recorder;

function startUserMedia(stream) {
    var input = audio_context.createMediaStreamSource(stream);
    __log('Media stream created.');
```

```

    recorder = new Recorder(input);
    __log('Recorder initialised.');
```

```

}

function startRecording(button) {
    //καλεί την συνάρτηση για την ηχογράφηση
    recorder && recorder.record();
}

```

```

//απενεργοποιεί το κουμπί record
button.disabled = true;
//ενεργοποιεί το κουμπί stop
button.nextElementSibling.disabled = false;
// δείχνει στον χρήστη ότι ηχογραφεί
__log('Recording...');
}

// ίδια λογική με το start αλλά για stop
function stopRecording(button) {
recorder && recorder.stop();
button.disabled = true;
button.previousElementSibling.disabled = false;
__log('Stopped recording.');
```

// καλεί τη συνάρτηση create WAV download link χρησιμοποιώντας audio data blob

```

createDownloadLink();
//καθαρίζει το recorder για νέα ηχογράφιση
recorder.clear();
}

function createDownloadLink() {
recorder && recorder.exportWAV(function(blob) {
//δημιουργία url για το αποτέλεσμα
var url = URL.createObjectURL(blob);
//δημιουργεί ένα αντικείμενο τύπου html list
//που περιέχει το αρχείο και το link για να το κατεβάσουμε
var li = document.createElement('li');
var au = document.createElement('audio');
var hf = document.createElement('a');
```

```

au.controls = true;
au.src = url;
hf.href = url;
// μορφή του ονοματος του link του αρχείου
hf.download = new Date().toISOString() + '.wav';
hf.innerHTML = hf.download;
li.appendChild(au);
li.appendChild(hf);
recordingslist.appendChild(li);
});
}
//αρχικοποιούνται οι μεταβλητές που χρειάζονται για να γίνει η ηχογράφιση.
//αν ο browser δεν υποστηρίζει ηχογράφιση ή δεν υπάρχει μικρόφωνο τυπώνει τα παρακάτω
//καλείται με το που φορτώσει η σελίδα
window.onload = function init() {
try {
// webkit shim
window.AudioContext = window.AudioContext ||
window.webkitAudioContext;
navigator.getUserMedia = navigator.getUserMedia ||
navigator.webkitGetUserMedia;
window.URL = window.URL || window.webkitURL;
```

```

audio_context = new AudioContext;
__log('Audio context set up.');
```

```

__log('navigator.getUserMedia ' + (navigator.getUserMedia ?
'available.' : 'not present!'));
```



```

    } catch (e) {
      alert('No web audio support in this browser!');
    }

    navigator.getUserMedia({audio: true}, startUserMedia, function(e)
    {
      __log('No live audio input: ' + e);
    });
  };
};

```

Recorder.js

Μαζί με το recorderWorker.js παρακάτω είναι τα εργαλεία για τον έλεγχο του μικροφώνου και την ηχογράφηση της εφαρμογής.

```

(function(window){

  var WORKER_PATH = 'static/recorderWorker.js';

  var Recorder = function(source, cfg){
    var config = cfg || {};
    var bufferLen = config.bufferLen || 4096;
    var numChannels = config.numChannels || 2;
    this.context = source.context;
    this.node = (this.context.createScriptProcessor ||
    this.context.createJavaScriptNode).call(this.context,
      bufferLen, numChannels, numChannels);
    var worker = new Worker(WORKER_PATH);
    worker.postMessage({
      command: 'init',
      config: {
        sampleRate: this.context.sampleRate,
        numChannels: numChannels
      }
    });
    var recording = false,
        currCallback;

    this.node.onaudioprocess = function(e){
      if (!recording) return;
      var buffer = [];
      for (var channel = 0; channel < numChannels; channel++){
        buffer.push(e.inputBuffer.getChannelData(channel));
      }
      worker.postMessage({
        command: 'record',
        buffer: buffer
      });
    }

    this.configure = function(cfg){
      for (var prop in cfg){
        if (cfg.hasOwnProperty(prop)){
          config[prop] = cfg[prop];
        }
      }
    }
  }
};

```

```

    this.record = function(){
        recording = true;
    }

    this.stop = function(){
        recording = false;
    }

    this.clear = function(){
        worker.postMessage({ command: 'clear' });
    }

    this.getBuffer = function(cb) {
        currCallback = cb || config.callback;
        worker.postMessage({ command: 'getBuffer' })
    }

    this.exportWAV = function(cb, type){
        currCallback = cb || config.callback;
        type = type || config.type || 'audio/wav';
        if (!currCallback) throw new Error('Callback not set');
        worker.postMessage({
            command: 'exportWAV',
            type: type
        });
    }

    worker.onmessage = function(e){
        var blob = e.data;
        currCallback(blob);
    }

    source.connect(this.node);
    this.node.connect(this.context.destination); //this should not
be necessary
};

Recorder.forceDownload = function(blob, filename){
    var url = (window.URL || window.webkitURL).createObjectURL(blob);
    var link = window.document.createElement('a');
    link.href = url;
    link.download = filename || 'output.wav';
    var click = document.createEvent("Event");
    click.initEvent("click", true, true);
    link.dispatchEvent(click);
}

window.Recorder = Recorder;
})(window);

```

recorderWorker.js

```

var recLength = 0,
    recBuffers = [],
    sampleRate,
    numChannels;

this.onmessage = function(e){
    switch(e.data.command){

```

```

    case 'init':
        init(e.data.config);
        break;
    case 'record':
        record(e.data.buffer);
        break;
    case 'exportWAV':
        exportWAV(e.data.type);
        break;
    case 'getBuffer':
        getBuffer();
        break;
    case 'clear':
        clear();
        break;
}
};

function init(config){
    sampleRate = config.sampleRate;
    numChannels = config.numChannels;
    initBuffers();
}

function record(inputBuffer){
    for (var channel = 0; channel < numChannels; channel++){
        recBuffers[channel].push(inputBuffer[channel]);
    }
    recLength += inputBuffer[0].length;
}

function exportWAV(type){
    var buffers = [];
    for (var channel = 0; channel < numChannels; channel++){
        buffers.push(mergeBuffers(recBuffers[channel], recLength));
    }
    if (numChannels === 2){
        var interleaved = interleave(buffers[0], buffers[1]);
    } else {
        var interleaved = buffers[0];
    }
    var dataview = encodeWAV(interleaved);
    var audioBlob = new Blob([dataview], { type: type });

    this.postMessage(audioBlob);
}

function getBuffer(){
    var buffers = [];
    for (var channel = 0; channel < numChannels; channel++){
        buffers.push(mergeBuffers(recBuffers[channel], recLength));
    }
    this.postMessage(buffers);
}

function clear(){
    recLength = 0;
    recBuffers = [];
    initBuffers();
}

```

```

function initBuffers(){
    for (var channel = 0; channel < numChannels; channel++){
        recBuffers[channel] = [];
    }
}

function mergeBuffers(recBuffers, recLength){
    var result = new Float32Array(recLength);
    var offset = 0;
    for (var i = 0; i < recBuffers.length; i++){
        result.set(recBuffers[i], offset);
        offset += recBuffers[i].length;
    }
    return result;
}

function interleave(inputL, inputR){
    var length = inputL.length + inputR.length;
    var result = new Float32Array(length);

    var index = 0,
        inputIndex = 0;

    while (index < length){
        result[index++] = inputL[inputIndex];
        result[index++] = inputR[inputIndex];
        inputIndex++;
    }
    return result;
}

function floatTo16BitPCM(output, offset, input){
    for (var i = 0; i < input.length; i++, offset+=2){
        var s = Math.max(-1, Math.min(1, input[i]));
        output.setInt16(offset, s < 0 ? s * 0x8000 : s * 0x7FFF, true);
    }
}

function writeString(view, offset, string){
    for (var i = 0; i < string.length; i++){
        view.setUint8(offset + i, string.charCodeAt(i));
    }
}

function encodeWAV(samples){
    var buffer = new ArrayBuffer(44 + samples.length * 2);
    var view = new DataView(buffer);

    /* RIFF identifier */
    writeString(view, 0, 'RIFF');
    /* RIFF chunk length */
    view.setUint32(4, 36 + samples.length * 2, true);
    /* RIFF type */
    writeString(view, 8, 'WAVE');
    /* format chunk identifier */
    writeString(view, 12, 'fmt ');
    /* format chunk length */
    view.setUint32(16, 16, true);
    /* sample format (raw) */
    view.setUint16(20, 1, true);
    /* channel count */

```

```

view.setUint16(22, numChannels, true);
/* sample rate */
view.setUint32(24, sampleRate, true);
/* byte rate (sample rate * block align) */
view.setUint32(28, sampleRate * 4, true);
/* block align (channel count * bytes per sample) */
view.setUint16(32, numChannels * 2, true);
/* bits per sample */
view.setUint16(34, 16, true);
/* data chunk identifier */
writeString(view, 36, 'data');
/* data chunk length */
view.setUint32(40, samples.length * 2, true);

floatTo16BitPCM(view, 44, samples);

return view;
}

```

Index.html

Ο κώδικας της αρχικής σελίδας της εφαρμογής :

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"
></script>
    <link
href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css"
"
      rel="stylesheet">
  </head>
  <body>
    <div class="container">
      <div class="header">
        <h1 ><a href="/" class="label label-default">Song
Identifier</a></h1>
      </div>
      <hr/>
      <div>
        <h2>Welcome to Song Identifier!</h2>
        <p>Identify a song you already have on your disk or record a
song fragment with your microphone and upload it afterwards</p>
        <form action="identify" method="post" enctype="multipart/form-
data" role="form">

          <input type="file" name="file"><br /><br />

          <button type="submit" class="btn btn-primary btn-lg"><span
class="glyphicon glyphicon-search"></span> Identify</button>
          <button onclick="startRecording(this);" class="btn btn-
success btn-lg">Record</button>
          <button onclick="stopRecording(this);" disabled class="btn
btn-danger btn-lg">Stop</button>

        </form>
      </div>

```

```

<h2>Recordings</h2>
<ul id="recordingslist"></ul>

<h2>Log</h2>
<pre id="log"></pre>

<script type=text/javascript src="{{url_for('static',
filename='script.js') }}"></script>
<script type=text/javascript src="{{url_for('static',
filename='recorder.js') }}"></script>
<script type=text/javascript src="{{url_for('static',
filename='recorderWorker.js') }}"></script>
</body>
</html>

```

Result.html

Ο κώδικας της σελίδας αποτελεσμάτων της εφαρμογής :

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"
></script>
    <link
href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css"
"
      rel="stylesheet">
    <style type="text/css">
      body {
        font-family: 'Open Sans', sans-serif;
      }
      #header {
        margin-top:20px;
        text-align:center;
        width: 100%;
      }
      #footer {
        margin-top:30px;
        text-align:center;
        width:100%;
        font-size:10px;
      }
      .sdiv {
        margin:30px;
      }
      .timg {
        margin-right: 20px;
        width:170px;
        height:170px;
      }
      .title {
        font-weight: bold;
        color: green;
        margin-bottom: 6px;
      }
      #result {
        width:500px;
        margin-left:auto;

```

```

        margin-right:auto;
    }
    #info {
        width:80%;
        text-align:center;
        height: 14px;
        margin-left:auto;
        margin-right:auto;
        margin-top:20px;
    }
    .player {
        margin-top:20px;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="header">
            <h1 ><a href="/" class="label label-default">Song
Identifier</a></h1>
        </div>
        <hr/>
        <div>
            <div class="jumbotron">
                {% if d["status"] %}
                <div id='result'> </div>
                {% else %}
                <p class="text-warning">The song could not be identified!</p>
                {% endif %}
            </div>
        </div>
        </div>
        {% if d["status"] %}
        <script type="text/javascript">

var audio
jQuery.ajaxSettings.traditional = true;

function createPlayer(audio) {
    var player = $("<audio class='player' preload='none'
controls='controls'>").attr("src", audio);
    return player;
}

$(document).ready(
    function() {
        //var tid = song.tracks[0].foreign_id.split(':')[2];
        var result = $("#result");

        var div = $("<div class='sdiv' id=" + '{{ song["id"] }}' + "-
div>");
        result.append(div);

        var tid = '{{ song["foreign_id"].split(':')[2] }}'
        var url = 'http://API.deezer.com/2.0/track/' + tid +
'?callback=?'

        jQuery.getJSON(url, { output:'jsonp'},
            function(data) {

```

```

        var link = $("<a target='deezer'>").attr('href',
data.link);
        var cover = $("<img class='timg'>").attr('src',
data.album.cover).attr({
            style: "float:left",
            width: "200",
            height: "200"
        });
        link.append(cover);
        div.append(link);

        var tdiv = $("<div class='tdiv'>");
        tdiv.append( $("<div class='title'>").text(data.title));
        tdiv.append( $("<div>").text(data.artist.name));
        tdiv.append( $("<div>").text(data.album.title));
        tdiv.append( $("<div>").text(data.album.release_date));
        div.append(tdiv);
        div.append(createPlayer(data.preview));
        div.append($("<br clear='left'>"));
    }
    );
}
);

</script>
{% endif %}
</body>
</html>

```

Pydoreso.py

To API του Doreso :

```

# Copyright 2014 Doreso
#
# Licensed under the Apache License, Version 2.0 (the "License"); you
may
# not use this file except in compliance with the License. You may
obtain
# a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT
# WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the
# License for the specific language governing permissions and
limitations
# under the License.
"""A Doreso API client library.

```

For full API documentation, visit <https://developer.doreso.com/>.

Typical usage:

```

doreso = Doreso('my API key')
d = doreso.song_identify_file('./test.mp3')
print json.dumps(d, indent=4)

```



```

"""

import requests
import subprocess
import json

class Doreso(object):
    def __init__(self, API_key, base_url=None, ffmpeg_path=None):
        self.API_key = API_key
        self.base_url = base_url or
'http://developer.doreso.com/API/v1'
        self.ffmpeg = ffmpeg_path or 'ffmpeg'

    def song_identify_file(self, filepath, start=0, duration=10):
        """Identify an audio file.

        This method will post 10 seconds wav buffer for recognition
        by default.
        Your can change start point or duration for practical use.
        """
        wav = self.gen_wavbuffer_from_filebuffer(open(filepath,
'rb').read(), start=start, duration=duration)
        r = requests.post(self._url('song/identify'), data=wav,
headers={'Content-Type': 'application/octet-stream'})
        return r.json()

    def gen_wavbuffer_from_filebuffer(self, file_buffer, start=0,
duration=10):
        proc = subprocess.Popen([self.ffmpeg, '-i', '-', '-ac', '1',
'-ar', '8000', '-f', 'wav', '-'], stdout=subprocess.PIPE,
stdin=subprocess.PIPE, stderr=open('/dev/null'))
        wav_buffer = proc.communicate(input=file_buffer)[0]
        if len(wav_buffer) < (start*128*1024)/8:
            return wav_buffer
        return
wav_buffer[(start*128*1024)/8:((start+duration)*128*1024)/8]

    def _url(self, path):
        return '%s/%s?API_key=%s' % (self.base_url, path,
self.API_key)

```