
Υπολογισμός ρεύματος δικτύου τροφοδοσίας ολοκληρωμένων με τεχνικές Deep Learning

Πανεπιστήμιο Θεσσαλίας

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ



Σεβαστή Υφαντή

Επιβλέπων Καθηγητής: Σταμούλης Γεώργιος
Συνεπιβλέπων Καθηγητής: Ευμορφόπουλος Νέστωρ
Συνεπιβλέπων Καθηγητής: Μπαργιώτας Δημήτριος

Ιούλιος 2020

Υπολογισμός ρεύματος δικτύου τροφοδοσίας ολοκληρωμένων με τεχνικές Deep Learning

Πανεπιστήμιο Θεσσαλίας

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ



Σεβαστή Υφαντή

Επιβλέπων Καθηγητής: Σταμούλης Γεώργιος
Συνεπιβλέπων Καθηγητής: Ευμορφόπουλος Νέστωρ
Συνεπιβλέπων Καθηγητής: Μπαργιώτας Δημήτριος

Ιούλιος 2020

IC current analysis using deep learning techniques

University of Thessaly

School of Engineering

Department of Electrical and Computer Engineering



Sevasti Yfanti

Supervisor: Prof. Stamoulis George

2nd Committee member: Prof. Evmorfopoulos Nestor

3rd Committee member: Prof. Bariotas Dimitrios

July 2020

Στην οικογένειά μου και τους
φίλους μου

Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής μου εργασίας , έρχεται σε πέρας η φοίτηση μου στο τμήμα των ηλεκτρολόγων μηχανικών και μηχανικών υπολογιστών του Πανεπιστημίου Θεσσαλίας. Πριν την παρουσίαση της αισθάνομαι την υποχρέωση αλλά και την επιθυμία να ευχαριστήσω ορισμένους ανθρώπους που βοήθησαν στην διεκπεραίωση των πρώτων ακαδημαϊκών μου βημάτων.

Πρώτο απο όλους θέλω να ευχαριστήσω τον επιβλέποντα καθηγητή της διπλωματικής εργασίας, καθηγητή κ. Γεώργιο Σταμούλη για την συνεχή καθοδήγηση του. Στην συνέχεια , θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτορα Δημήτριο Γαρυφάλλου για την επιβλεψη και επίλυση αποριών.

Θα ήθελα να εκφράσω την ευγνωμοσύνη μου στους γονείς μου, τον αδερφό μου και την γιαγιά μου που με στηρίζουν πάντα και μου έδωσαν την δυνατότητα και την ώθηση να αποκτήσω αυτή την εκπαίδευση και τέλος, τους φίλους μου που μοιραστήκαμε όλη την πανεπιστημιακή εμπειρία.

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

«Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής».

Η Δηλούσα
Σεβαστή Υφαντή
15-07-2020

Περίληψη

Στην παρούσα διπλωματική εργασία γίνονται ταξινομήσεις των ρευμάτων δικτύου τροφοδοσίας ολοκληρωμένων με διάφορους αλγορίθμους machine learning, προκειμένου να μελετήσουμε το ρεύμα σε σχέση με τις εισόδους του.

Στο πρώτο κεφάλαιο αναλύεται η κατανάλωση ισχύος. Εξηγείται το πρόβλημα της πτώσης τάσης και του ρεύματος στα ολοκληρωμένα κυκλώματα, ενώ περαιτέρω εκτίθενται οι λόγοι για τους οποίους είναι σημαντικό να αναλυθούν τα δίκτυα τροφοδοσίας ολοκληρωμένων με τεχνικές Machine learning.

Στο δεύτερο κεφάλαιο, γίνεται ανάλυση του πως διαμορφώθηκε το σύνολο των δεδομένων. Αρχικά, χρειάστηκε η κατανόηση ενός προγράμματος προσομοίωσης και μοντελοποίησης ηλεκτρονικών κυκλωμάτων, του HSPICE. Στο αρχείο τύπου hspice που δόθηκε, χρειαζόταν η προσθήκη κάποιων λειτουργιών, προκειμένου να παραχθούν τα δεδομένα (ρεύμα και τιμές εισόδων). Έπειτα, με τον υπολογισμό των ρευμάτων σε μια συγκεκριμένη χρονική περίοδο και μέσα από ένα σύνολο αλγορίθμων επεξεργασίας σε python κρατούνται οι τιμές τις χρονικές στιγμές των μεταβολών των εισόδων από 0 σε 1 και αντίθετα. Κατ' αυτόν τον τρόπο δημιουργούνται 4 σύνολα δεδομένων.

Το τρίτο κεφάλαιο εμπεριέχει τη θεωρητική εξήγηση κάποιων μοντέλων ταξινόμησης από την βιβλιοθήκη sklearn, τα οποία στην συνέχεια χρησιμοποιούνται, προκειμένου να διαπιστωθεί ποιο προσφέρει τα καλύτερα αποτελέσματα. Πιο συγκεκριμένα, τα δεδομένα χωρίζονται σε Train και Test και κατηγοριοποιούνται ανάλογα με το ποιο συνδυασμοί εισόδων έχουν υψηλό ρεύμα, ποιο μέσο και ποιο χαμηλό.

Στο τέταρτο κεφάλαιο δίνονται τα τελικά αποτελέσματα, ενώ η διπλωματική εργασία ολοκληρώνεται με τα συμπεράσματα που βγήκαν και σκέψεις για μελλοντική μελέτη.

Abstract

In this dissertation, the currents of a integrated power supply network are being classified by various machine learning techniques. In this way , we can study the relation between the current and the network's inputs.

In the first chapter, the power consumption is analyzed. The problem of voltage and current drop in integrated circuits is explained, while further reasons are given for why it is important to analyze the integrated supply networks with Machine learning techniques.

In the second chapter, an analysis is made of how the data set was configured. Initially, it was necessary to understand a program of simulation and modeling of electronic circuits, HSPICE. In the given hspice file, it was imperative to add some functions in order to generate the data (current and input values). Then, by calculating the currents in a given period of time and through a set of processing algorithms in python,we receive the time values where the input is alternated from 0 to 1 and the opposite.In this way, 4 data sets are created.

The third chapter contains the theoretical explanation of some classification models from the library sklearn, which are then used thus determining which one offers the best results.More specifically, the data is divided into Train and Test set and is categorized according to which input combinations have high current, which medium and which low.

In the fourth chapter the final results are given, while the dissertation is completed with conclusions and thoughts for future study.

Περιεχόμενα

1	Εισαγωγή	10
1.1	Κατανάλωση Ισχύος	10
1.2	Πτώση τάσης	11
1.3	Μηχανική μάθηση	12
2	Δημιουργία Dataset	13
2.1	HSpice	13
2.1.1	Θεωρία HSpice	13
2.1.2	Εκτέλεση HSpice	15
2.2	Python	16
2.2.1	Επεξεργασία του συνόλου δεδομένων	16
2.2.2	Χωρισμός του συνόλου δεδομένων	17
3	Ταξινόμηση	19
3.1	Γενικότερη θεωρία	19
3.1.1	Ταξινόμηση	19
3.1.2	sklearn βιβλιοθήκη	19
3.2	Μοντέλα Ταξινόμησης	19
3.2.1	Logistic Regression	19
3.2.2	Support Vector Machines	20
3.2.3	Random Forests	23
3.2.4	Neural Network	24
3.2.5	Multi class Classification	24
3.3	Εφαρμογή	25
4	Αποτελέσματα	28
5	Επίλογος	31
	Βιβλιογραφία	32

Κατάλογος Σχημάτων

1.1	Κυματομορφή της δυναμικής πτώσης τάσης	12
2.1	Παλμοί εισόδων όπως φαίνονται με χρήση Waveview	16
2.2	cross validation	18
3.1	Logistic Regression	20
3.2	Support Vector Machines	22
3.3	συνάρτηση απόφασης για ένα γραμμικά διαχωρίσιμο πρόβλημα, με τρία δείγματα στα όρια του περιθωρίου, που ονομάζονται 'διανύσματα υποστήριξης' "Support Vectors":	23
3.4	Random Forests	24
3.5	Διάγραμμα χρόνου - ρεύματος	26

Κατάλογος Πινάκων

2.1	Κατάλογος συντομογραφιών σύνταξης παλμών	14
2.2	Κατάλογος συντομογραφιών σύνταξης PWL	15
2.3	Είδη συνόλων δεδομένων	17
4.1	Πίνακας Αποτελεσμάτων	30

Κεφάλαιο 1

Εισαγωγή

1.1 Κατανάλωση Ισχύος

Η εξέλιξη της τεχνολογίας έχει οδηγήσει σε μια νέα γενιά ημιαγωγών ικανούς να προσφέρουν αποδοτικότερα και οικονομικότερα κυκλώματα. Το κόστος του τρανζίστορ συνεχώς μειώνεται και είναι σημαντικό να γίνεται πλήρη εκμετάλλευση των δυνατοτήτων που προσφέρονται. Βέβαια, η μείωση της καθυστέρησης της πύλης, αντισταθμίζεται από την αύξηση του πλήθους των πυλών σε μια χωρική μονάδα το οποίο οδηγεί στην ανάγκη επίλυσης νέων προβλημάτων όπως την διαχείριση της ισχύος.

Η κατανάλωση Ισχύος, αποτελεί ένα από τα σημαντικότερα θέματα της σύγχρονης σχεδίασης. Θεωρητικά, η σταθερή κλιμάκωση του μεγέθους, οδηγεί σε σταθερή αύξηση της πυκνότητας ισχύος και η πυκνότητα ισχύος αυξάνεται αργά ακολουθώντας την αύξηση του μεγέθους του ολοκληρωμένου κυκλώματος. Στην πράξη όμως, η κατανάλωση ισχύος εκτοξεύτηκε, λόγω της ραγδαίας αύξησης των συχνοτήτων ρολογιού. Μια αύξηση πολύ μεγαλύτερης ταχύτητας από ότι προβλέπαμε με την κλασσική κλιμάκωση καθώς και η τάση τροφοδοσίας VDD παρέμεινε σε υψηλά επίπεδα. [3] Η Κατανάλωση ισχύος του κυκλώματος είναι ανάλογη του ρεύματος εισόδου, πράγμα που σημαίνει ότι οι δύο αυτές ποσότητες μπορούν να χρησιμοποιούνται εναλλακτικά η μια της άλλης.

$$P(t) = V_{DD} * I(t)$$

Η ισχύς είναι η ενέργεια που καταναλώνουμε ανά μονάδα χρόνου. Στα ολοκληρωμένα κυκλώματα, υπάρχουν τρεις τρόποι κατανάλωσης ισχύος:

Δυναμική κατανάλωση ισχύος:

Η συγκεκριμένη κατανάλωση ισχύος παρατηρείται μόνο όταν η πύλη αλλάζει κατάσταση, και συγκεκριμένα όταν η είσοδος κάνει μετάβαση από το 0 στο 1, εξ αιτίας της φόρτισης και αποφόρτισης των παρασιτικών χωρητικότητων των εσωτερικών κόμβων ενός κυκλώματος. Μια δυναμική κατανάλωση ισχύος είναι και η ισχύς βραχυκυκλώματος, που αποτελεί μέρος της συνολικής δυναμικής κατανάλωσης ισχύος αλλά αντιμετωπίζεται χωριστά. Πιο συγκεκριμένα, σε μια μετάβαση της στατικής πύλης CMOS, τα PMOS και NMOS τρανζίστορ είναι ταυτόχρονα ενεργοποιημένα για ένα χρονικό διάστημα. Εκείνη την στιγμή, σχηματίζεται μια απευθείας διαδρομή μεταξύ Vdd-Gnd. Αυτό έχει ως αποτέλεσμα η ισχύς του ρεύματος αυτού να καταναλώνεται στα τρανζίστορ χωρίς να συνεισφέρει στην φόρτιση της χωρητικότητας της εξόδου. Με τα σημερινά δεδομένα, η δυναμική ισχύς, καλύπτει το μεγαλύτερο μέρος της συνολικής ισχύος που καταναλώνει το ολοκληρωμένο σύστημα. Ένα ποσοστό όμως που δεν πρόκειται να αυξηθεί σημαντικά στα επόμενα χρόνια, γιατί αν συμβεί αυτό θα είναι

δύσκολο και δαπανηρό να παραμείνει το ολοκληρωμένο κύκλωμα σε κανονική θερμοκρασία.

Στατική κατανάλωση ισχύος:

Παρατηρείται εξ αιτίας των ρευμάτων διαρροής στα τρανζίστορ, ενώ υφίσταται ακόμη και όταν το κύκλωμα είναι ανενεργό (idle state). Τα ρεύματα διαρροής, αποτελούν μια στατική κατανάλωση ισχύος. Ακόμα και όταν η έξοδος της πύλης δεν έχει μεταβάσεις, κάθε τρανζίστορ διαρρέεται από ρεύμα, η ποσότητα του οποίου είναι αντιστρόφως ανάλογη του μεγέθους του τρανζίστορ. Επομένως, παρατηρείται κατανάλωση ισχύος ακόμα και τότε. Το ρεύμα διαρροής αυξάνει με την άνοδο της θερμοκρασίας, με την μείωση της τάσης κατωφλίου των τρανζίστορ, ενώ σχετίζεται άμεσα και με την τάση τροφοδοσίας του κυκλώματος. Σημαντικό είναι να αναφερθεί πως ενώ μέχρι πρότετως αποτελούσε ένα μικρό ποσό της συνολικής κατανάλωσης ισχύος, στις μερες μας η στατική κατανάλωση από τα ρεύματα διαρροής αποτελούν έως και το 30% της συνολικής κατανάλωσης ισχύος.

1.2 Πτώση τάσης

Στον σχεδιασμό ολοκληρωμένων κυκλωμάτων γίνεται η υπόθεση πως υπάρχει μια ιδανική πηγή τροφοδοσίας ικανή να προσφέρει οποιαδήποτε ποσότητα ρεύματος προκειμένου να διατηρήσει την απαιτούμενη τάση. Στην πραγματικότητα φυσικά, υπάρχει αντίσταση, η οποία προκαλεί την πτώση τάσης στο ολοκληρωμένο. Όσο πιο στενές είναι οι γραμμές του μετάλλου, τόσο αυξάνεται και η αντίσταση. Αυτή η ωμική πτώση τάσης υποβαθμίζει τις στάθμες των σημάτων, κάτι ιδιαίτερα σημαντικό στα δικτυώματα διανομής ισχύος, όπου τα επίπεδα των ρευμάτων μπορούν να είναι της τάξης των μερικών Ampere. Οι τροποποιημένες τιμές της τάσης τροφοδοσίας μειώνουν τα περιθώρια θορύβου καθώς και αλλάζουν τις λογικές στάθμες, πάντα αναλογικά με την απόσταση από τους αιροδέκτες τροφοδοσίας. Ως συνέχεια αυτού, η διαφορά στις λογικές στάθμες που προκαλείται από την πτώση τάσης I_P πάνω στα καλώδια τροφοδοσίας, μπορεί να προκαλέσει μερική αγωγή στα τρανζίστορ, καταλήγοντας σε απροσδόκητη εκφόρτιση του προφορτισμένου δυναμικού κόμβου ή στατική κατανάλωση ισχύος εάν η συνδεδεμένη πύλη είναι στατική. Με λίγα λόγια, οι παλμοί του ρεύματος από την λογική υλοποίησης ενός ολοκληρωμένου κυκλώματος στις μνήμες και στους αιροδέκτες εισόδου και εξόδου προκαλούν πτώσεις τάσης πάνω στο δίκτυο διανομής τροφοδοσίας και είναι οι βασικότερες πηγές δημιουργίας θορύβου τροφοδοσίας μέσα στο κύκλωμα. Φυσικά, εκτός του κινδύνου ως προς την αξιοπιστία του κυκλώματος, οι πτώσεις τάσης I_R στο δίκτυο τροφοδοσίας επιδρούν επίσης και στις επιδόσεις του συστήματος. καθώς μπορεί να προκληθεί αύξηση της καθυστέρησης. [8]

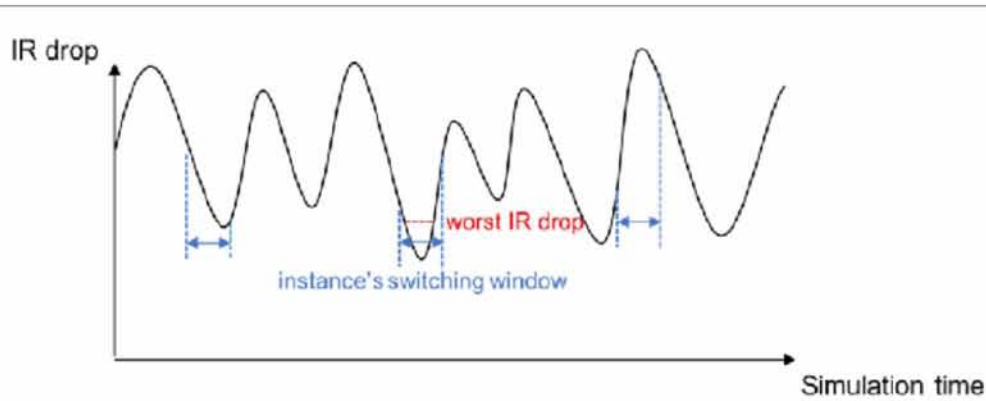


Figure 1.1: Κυματομορφή της δυναμικής πτώσης τάσης

1.3 Μηχανική μάθηση

Ο καθορισμός του δικτύου διανομής ισχύος αποτελεί μια δύσκολη εργασία. Το πλέγμα ισχύος μπορεί να μοντελοποιηθεί ως ένα δίκτυωμα από αντιστάτες και πηγές ρεύματος που περιλαμβάνει εκατομμύρια στοιχεία. Πολύ συχνά υπάρχουν πολλές διαδρομές μεταξύ των ακροδεκτών ισχύος και των συνδέσεων τροφοδοσίας μιας μονάδας του ολοκληρωμένου κυκλώματος ή μιας πύλης. Αν και γενικά το ρεύμα ακολουθεί της διαδρομή με την μικρότερη αντίσταση, η ακριβής διαδρομή εξαρτάται από παράγοντες όπως το ρεύμα που ρέει στις γειτονικές μονάδες που μοιράζονται το ίδιο δίκτυωμα. Η ανάλυση γίνεται σύνθετη λόγω του γεγονότος ότι τα μέγιστα ρεύματα που διαρρέουν τις συνδεδεμένες μονάδες είναι κατανεμημένα ως προς το χρόνο. Η πτώση τάσης είναι ένα δυναμικό φαινόμενο. Επομένως, δεν μπορούμε να έχουμε μια ακριβή εικόνα των κινδύνων που οφείλονται στην πτώση τάσης IR αν δεν επαληθεύουμε την λειτουργία ολόκληρου του ολοκληρωμένου κυκλώματος, σαν να ήταν μια οντότητα. Τα εργαλεία που χρησιμοποιούνται για τον σκοπό αυτό θα πρέπει να έχουν την ικανότητα να αναλύουν εκατομμύρια πλεγμάτων αντιστατών. [2]

Στα πλαίσια αυτής της ανάγκης, να έχουμε δηλαδή μια ακριβής, συνολική εικόνα του ολοκληρωμένου κυκλώματος, εντάσσεται και η ταξινόμηση των ρευμάτων ανάλογα με την κατάσταση των εισόδων τους σε σχέση με την τιμή τους σε Ampere.

Κεφάλαιο 2

Δημιουργία Dataset

2.1 HSpice

2.1.1 Θεωρία HSpice

Το SPICE (Simulation Program with Integrated Circuit Emphasis) είναι ένα πρόγραμμα προσομοίωσης και μοντελοποίησης που χρησιμοποιείται από μηχανικούς, για να προβλέψει μαθηματικά τη συμπεριφορά των ηλεκτρονικών κυκλωμάτων. Αναπτύχθηκε στο Πανεπιστήμιο της Καλιφόρνια στο Μπέρκλεϊ. Το SPICE μπορεί να χρησιμοποιηθεί για την προσομοίωση κυκλωμάτων σχεδόν όλων των περιπλοκών. Ωστόσο, χρησιμοποιείται γενικά για την πρόβλεψη της συμπεριφοράς κυκλωμάτων χαμηλής έως μέσης συχνότητας (DC έως περίπου 100MHz).[9]

Σε ένα αρχείο, οι πηγές και οι παθητικές συσκευές ορίζονται με θετικό και αρνητικό κόμβο ($N+$, $N-$). Το ρεύμα ρέει μέσα από αυτά τα στοιχεία από τον θετικό προς τον αρνητικό κόμβο. Είναι ιδιαίτερα σημαντικό για μια τρέχουσα πηγή να αναφερθεί αυτό, επειδή το καθορισμένο ρεύμα ρέει ελαφρώς αντίθετα διαισθητικά μέσα από το στοιχείο. Συνήθως, σε πρώτο στάδιο ορίζονται όλες οι πηγές τάσης και ρεύματος, ενώ έπονται οι παθητικές και ενεργές συσκευές. Για τις πηγές, η είσοδος προσδιορισμού είναι DC, AC ή μια παροδική συνάρτηση (transient function), όπως ισχύει για την αντίστοιχη ανάλυση. Μια πηγή μπορεί να καθοριστεί και με τις τρεις και η ανάλυση θα επιλέξει την κατάλληλη παράμετρο.[1]

Τουλάχιστον η τιμή DC καθορίζεται, ενώ για ένα σήμα AC, κάποιος καθορίζει το μέγεθος (π.χ. V1 σε gnd AC 1).

Για παροδικές συναρτήσεις ενδιαφέροντος υπάρχουν οι:

1. Pulse Train.

Το HSpice έχει μια λειτουργία τραπεζοειδούς παλμού, η οποία ξεκινά με μια αρχική καθυστέρηση από την αρχή της προσομοίωσης. Κατά τη διάρκεια της έναρξης, η τάση ή το ρεύμα αλλάζουν γραμμικά από την αρχική τους τιμή, στην τιμή του παλμού. Μετά, το οροπέδιο παλμού - το χρονικό διάστημα, δηλαδή, στο οποίο ο παλμός παραμένει στην μέγιστη τιμή του-, η τάση ή το ρεύμα κινείται γραμμικά κατά μήκος μιας ράμπας ανάκτησης, πίσω στην αρχική του τιμή.

Η σύνταξη του:

```
PULSE (<v1 v2 <td <tr <tf <pw <per>>> < >
```

Παράδειγμα:

```
VIN 3 0 PULSE(-1 1 2NS 2NS 2NS 50NS 100NS)
```

όπου,

v1	αρχική τιμή της τάσης ή του ρεύματος, πριν από την έναρξη του παλμού
v2	τιμή όροπεδίου παλμού
td	χρόνος καθυστέρησης σε δευτερόλεπτα από την αρχή έως την πρώτη ράμπα έναρξης (προεπιλογή = 0,0 και για μικρότερο του 0,0, χρησιμοποιείται επίσης 0,0)
tr	διάρκεια της εκκίνησης, από την αρχική τιμή έως την τιμή του παλμού (χρόνος αντίστροφης διέλευσης) (προεπιλογή = TSTEP)
tf	διάρκεια της ανάκαμψης, από το οροπέδιο παλμού στην αρχική τιμή (χρόνος μετάβασης προς τα εμπρός) (προεπιλογή = TSTEP)
pw	πλάτος παλμού (προεπιλογή = TSTEP)
per	περίοδος επανάληψης παλμών σε δευτερόλεπτα (προεπιλογή = TSTEP, το μεταβατικό χρονικό βήμα)

Table 2.1: Κατάλογος συντομογραφιών σύνταξης παλμών

2. Piece-wise Linear

Η φόρμα PWL περιγράφει μια γραμμική κυματομορφή ανά τμήμα. Κάθε ζεύγος τιμών τάσης-χρόνου καθορίζει μια γωνία της κυματομορφής. Η τάση τις χρονικές στιγμές μεταξύ των γωνιών αποτελεί τη γραμμική παρεμβολή των τάσεων στις γωνίες. Εάν ο χρόνος του πρώτου ζεύγους δεν είναι μηδέν, τότε η τάση DC της πηγής θα χρησιμοποιηθεί ως αρχική τιμή. Εάν η προσομοίωση συνεχίζεται πέρα από το χρόνο του τελευταίου ζεύγους, τότε η τάση αυτού του ζεύγους θα διατηρηθεί για το υπόλοιπο της προσομοίωσης.

Η σύνταξή του:

```
PWL (<t1 v1 <t2 v2 t3 v3...> <R <=repeat> <TD=delay> <>>
```

Παράδειγμα:

```
V3 10 5 PWL(0us 0V 1us 0V 1.3us 2V 2us 2.5V 3us 0.5V 3.4us 0.5V)
```

όπου,[6]

v1	Στιγμιαία τάση σε Volts ή Amps
t1	στιγμή σε δευτερόλεπτα
R	προκαλεί την επανάληψη της λειτουργίας
repeat	καθορίζει το σημείο εκκίνησης της κυματομορφής που πρόκειται να επαναληφθεί
TD	είναι λέξη-κλειδί για χρονική καθυστέρηση προτού αρχίσει το κομμάτι
delay	καθορίζει το χρονικό διάστημα για να καθυστερήσει η γραμμική συνάρτηση

Table 2.2: Κατάλογος συντομογραφιών σύνταξης PWL

2.1.2 Εκτέλεση HSpice

Για την κατανόηση του Hspice αρχείου καθώς για και να το τρέξω χρειάστηκε να ακολουθήσω το VLSI Tutorial [7]

Για να εκτελέσω το hspice :

1. *hspice yourSpiceFile.sp*
2. Επειδή ήθελα να λάβω αρχείο καταγραφής εξόδου, έτρεξα το εξής: *hspice yourSpiceFile.sp > yourSpiceFile.out*

Με αυτόν τον τρόπο μπορούσα να ελέγγω απο το αρχείο καταγραφής εξόδου για το εάν υπάρχει προειδοποίηση ή σφάλμα. Εάν το μήνυμα αναφέρει ότι η εργασία ολοκληρώθηκε (job concluded), αυτό σημαίνει ότι η προσομοίωση εκτελέστηκε με επιτυχία, διαφορετικά, εάν αναφέρει ότι η εργασία ματαιώθηκε (job aborted) ή κάποιο άλλο μήνυμα, η προσομοίωσή μου δεν είχε ολοκληρωθεί.

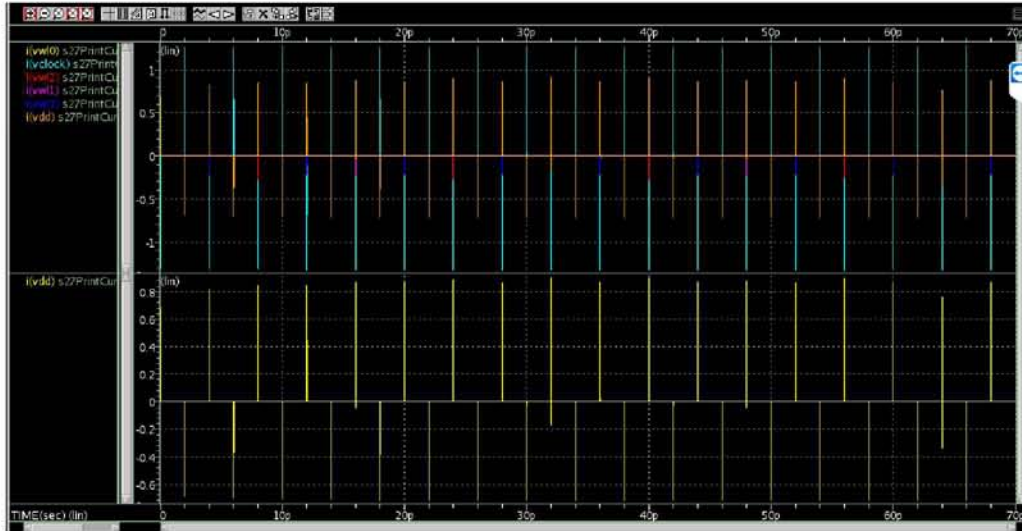
Η κυματομορφή των αποτελεσμάτων της παροδικής (transient) ανάλυσής αποθηκεύεται στο *yourSpiceFile.tr0*. Έτσι, μπορώ να χρησιμοποιώ το πρόγραμμα προβολής κυματομορφής (WaveView) για να το ανοίξω.

Για την εκτέλεση του WaveView :

1. *wv &*
2. ανοίγετε το αρχείο προσομοίωσης κάνοντας κλικ στο κουμπί Import Waveform File
3. Διαλέγετε το αρχείο

4. επιλογή του top level και μετά των σημάτων που επιθυμείτε να εμφανιστούν

Το αποτέλεσμα που προκύπτει είναι το ακόλουθο:



Σχήμα 2.1: Παλμοί εισόδων όπως φαίνονται με χρήση Waveview

Τα προσωρινά αποτελέσματα μέτρησης ανάλυσης αποθηκεύονται στο yourSpice-File.mt0. και μπορούσα να χρησιμοποιήσω οποιοδήποτε πρόγραμμα επεξεργασίας κειμένου για να το διαβάσω.

2.2 Python

Για την επεξεργασία του συνόλου δεδομένων χρησιμοποιήθηκαν οι εξής βιβλιοθήκες:

1. pandas
2. numpy
3. from sklearn datasets
4. from sklearn train_test_split

Έχοντας εμφανίσει τις 7 εισόδους ρεύματος και το ρεύμα στις χρονικές στιγμές των μεταβάσεων απο 0 σε 1 το σύνολο των δεδομένων μου πέρασε απο μία σειρά επεξεργαστικών συναρτήσεων για την δημιουργία 4 διαφορετικών συνδιασμών εισόδου και εξόδου.

2.2.1 Επεξεργασία του συνόλου δεδομένων

Η εκμάθηση ενός μοντέλου ταξινόμησης και η δοκιμή της απόδοσής του στο ίδιο σύνολο δεδομένων αποτελεί μια λάθος πρακτική η οποία οδηγεί σε ποσοστό επιτυχίας 100%. Ένα ποσοστό όμως που είναι πλασματικό αφού το μοντέλο δεν θα μπορέσει να ταξινομήσει σωστά άγνωστα για αυτό δεδομένα με την ίδια ευκολία. Το φαινόμενο αυτό

Input	Output
0, 1	πραγματικές τιμές ρεύματος
0, 1	αναφορικές τιμές ρεύματος(0, 1, 2)
-1, 0, 1, 2 (μεταβάσεις ρευμάτων)	πραγματικές τιμές ρεύματος
-1, 0, 1, 2	αναφορικές τιμές ρεύματος(0, 1, 2)

Table 2.3: Είδη συνόλων δεδομένων

ονομάζεται *overfitting* και προκειμένου να το αποφύγουμε, μια συνηθισμένη πρακτική κατά την εκτέλεση ενός (εποπτευόμενου) πειράματος μηχανικής μάθησης είναι να κρατάμε μέρος των διαθέσιμων δεδομένων ως σύνολο δοκιμών.

Για την καλύτερη απόδοση των αλγορίθμων ταξινόμησης, δοκιμάστηκαν 2 τρόποι διαχωρισμού. Στον πρώτο τρόπο, τα σύνολα δεδομένων χωρίστηκαν σε Train και Test set με μια απλή συνάρτηση της βιβλιοθήκης *sklearn* και στον δεύτερο, έγινε χρήση ενός αλγορίθμου *cross validation*.

2.2.2 Χωρισμός του συνόλου δεδομένων

Το Train set είναι το δείγμα δεδομένων που χρησιμοποιείται για την εκπαίδευση του μοντέλου μας ενώ το test set αποτελεί ένα δείγμα δεδομένων που χρησιμοποιείται για την αμερόληπτη αξιολόγηση ενός τελικού μοντέλου.

Η αναλογία διαχωρισμού συνόλου δεδομένων εξαρτάται κυρίως από 2 πράγματα. Πρώτον, ο συνολικός αριθμός δειγμάτων στα δεδομένα και δεύτερον, το μοντέλο εκπαίδευσης. Στην προκειμένη περίπτωση επειδή στα μοντέλα που χρησιμοποιήσα έχουν πολύ λίγες παραμέτρους, η αναλογία του test και train set είναι της τάξης 1 προς 10.

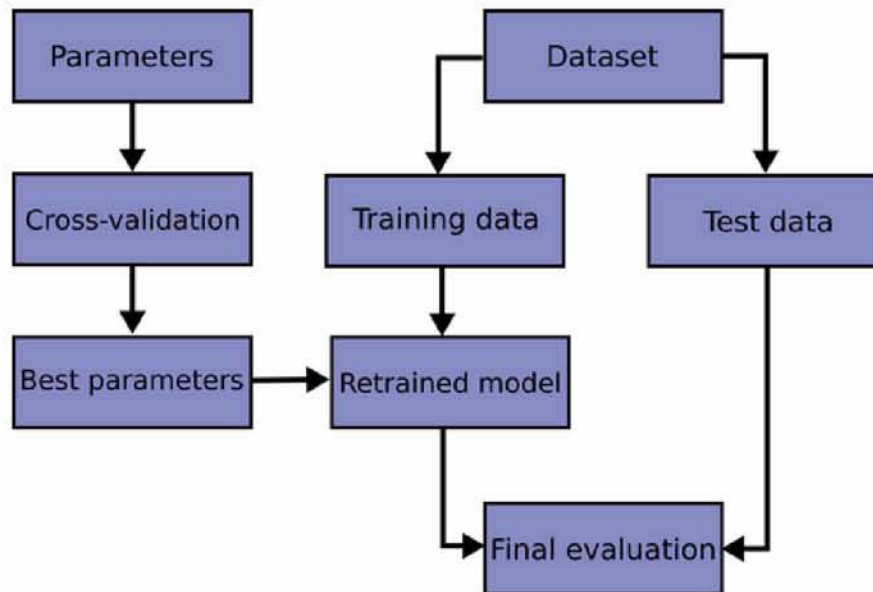
Επειδή σε κάποια μοντέλα ταξινομητών, υπάρχουν κάποιες παράμετροι που υπολογίζονται χειροκίνητα υπάρχει ο κίνδυνος *overfitting* στο σύνολο των δοκιμών. Ιδιαίτερα στα SVM που οι παράμετροι μπορούν να τροποποιηθούν έως ότου ο εκτιμητής αποδώσει βέλτιστα. Για την επίλυση αυτού του προβλήματος, ένα ακόμη μέρος του συνόλου δεδομένων μπορεί να θεωρηθεί ως το λεγόμενο σετ επικύρωσης (Validation set): το training προχωρά στο σύνολο εκπαίδευσης, μετά γίνεται αξιολόγηση στο σύνολο επικύρωσης και όταν το πείραμα φαίνεται να είναι επιτυχές, η τελική αξιολόγηση μπορεί να γίνει στο σύνολο δοκιμών.

Ωστόσο, διαχωρίζοντας τα διαθέσιμα δεδομένα σε τρία σύνολα, μειώνουμε δραστηκά τον αριθμό των δειγμάτων που μπορούν να χρησιμοποιηθούν για την εκμάθηση του μοντέλου και τα αποτελέσματα μπορούν να εξαρτηθούν από μια συγκεκριμένη τυχαία επιλογή για το ζεύγος (train, validation). Μια λύση σε αυτό το πρόβλημα είναι μια διαδικασία που ονομάζεται διασταυρούμενη επικύρωση(*cross validation*). Ένα σετ δοκιμών θα πρέπει ακόμη να πραγματοποιηθεί για τελική αξιολόγηση, αλλά το σετ επικύρωσης δεν απαιτείται πλέον. Στη βασική προσέγγιση, που ονομάζεται *k-fold CV*, το σετ εκπαίδευσης χωρίζεται σε *k* μικρότερα σύνολα (άλλες προσεγγίσεις περιγράφονται παρακάτω, αλλά γενικά ακολουθούν τις ίδιες αρχές). Ακολουθείται η ακόλουθη διαδικασία για κάθε ένα από τα *k* "folds":

1. Το μοντέλο εκπαιδεύεται χρησιμοποιώντας *k-1* folds ως σετ εκπαίδευσης
2. το μοντέλο που προκύπτει επικυρώνεται στο υπόλοιπο μέρος των δεδομένων (δηλαδή, χρησιμοποιείται ως σύνολο δοκιμών για τον υπολογισμό ενός μέτρου α-

πόδοσης όπως η ακρίβεια).

Το μέτρο απόδοσης που αναφέρεται από το k-fold cross-validation είναι τότε ο μέσος όρος των τιμών που υπολογίζονται στο loop. Αυτή η προσέγγιση μπορεί να είναι υπολογιστικά δαπανηρή, αλλά δεν σπαταλά πάρα πολλά δεδομένα, το οποίο είναι ένα σημαντικό πλεονέκτημα σε προβλήματα όπου ο αριθμός των δειγμάτων είναι πολύ μικρός.



Σχήμα 2.2: cross validation

Κεφάλαιο 3

Ταξινόμηση

3.1 Γενικότερη θεωρία

3.1.1 Ταξινόμηση

Η ταξινόμηση στη μηχανική μάθηση (Classification in Machine Learning) είναι μια προσέγγιση εποπτευόμενη κατά την οποία ο υπολογιστής μαθαίνει από τα δεδομένα που του δίνονται και κάνει νέες παρατηρήσεις ή ταξινομήσεις. Λέγεται Εποπτευόμενη επειδή γνωρίζουμε την σωστή έξοδο και ο υπολογιστής συγκρίνει το αποτέλεσμα του με αυτήν. Ο κύριος στόχος είναι να προσδιοριστεί σε ποια κατηγορία θα εμπίπτουν τα νέα δεδομένα. Ο Ταξινομητής (Classifier) επιδιώκει την αντιστοίχιση πραγματικών τιμών σε προεπιλεγμένες κατηγορίες.

3.1.2 sklearn βιβλιοθήκη

Η βιβλιοθήκη της sklearn παρέχει ένα σύνολο απο αλγορίθμους επίλυσης προβλημάτων, οι οποίοι εντάσσονται σε δυο κατηγορίες. Την εποπτευόμενη μάθηση, και την μη εποπτευόμενη. Η ταξινόμηση, σύμφωνα με την οποία χωρίζω τα ρεύματα που έχουν παραχθεί απο το αρχείο hspace ανήκει στην εποπτευόμενη μάθηση (supervised learning). Από το σύνολο δεδομένων, πρέπει να έχω δείγματα και απο τις 3 πιθανές τάξεις (χαμηλό, μεσαίο και υψηλό ρεύμα), στις οποίες κάνω 'fit' έναν εκτιμητή για να μπορέσω να προβλέψω τις τάξεις στις οποίες ανήκουν τα 'άγνωστα' ακόμα δείγματα. Στο scikit-learn, ένας εκτιμητής για την ταξινόμηση είναι ένα αντικείμενο Python που εφαρμόζει τις μεθόδους 'fit' (X, y) και 'predict' (T)

3.2 Μοντέλα Ταξινόμησης

Παρακάτω αναλύονται 5 μοντέλα ταξινόμησης.

3.2.1 Logistic Regression

[4]

Είναι ένας αλγόριθμος ταξινόμησης στη μηχανική μάθηση που χρησιμοποιεί μία ή περισσότερες ανεξάρτητες μεταβλητές για τον προσδιορισμό ενός αποτελέσματος. Το

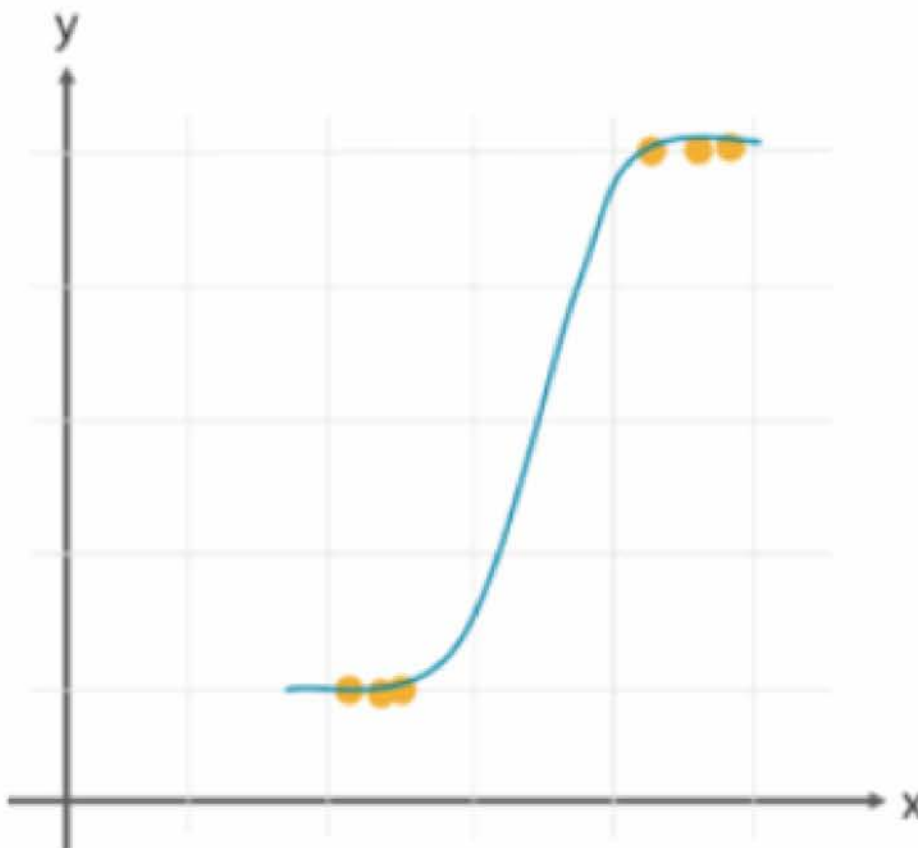
αποτέλεσμα μετρίεται με μια διχότομη μεταβλητή που σημαίνει ότι θα έχει μόνο δύο πιθανά αποτελέσματα.

Ο στόχος της λογιστικής παλινδρόμησης (Logistic Regression) είναι η εύρεση μιας καλύτερης σχέσης μεταξύ της εξαρτημένης μεταβλητής και ενός συνόλου ανεξάρτητων μεταβλητών. είναι μια ειδική περίπτωση γραμμικής παλινδρόμησης (Linear Regression) όπου προβλέπουμε μόνο το αποτέλεσμα σε μια κατηγορηματική μεταβλητή. Προβλέπει την πιθανότητα να συμβεί μια κατάσταση χρησιμοποιώντας τη συνάρτηση σελεστλαν-γυαρεενγλισηλογ.

Λογιστις Ρεγρεσσιον εκυατιον:

$$p = 1 / (1 + e^{- (\beta_0 + \beta_1 X_1 + \beta_2 X_2 \dots + \beta_n X_n)})$$

[4]



Σχήμα 3.1: Logistic Regression

3.2.2 Support Vector Machines

Το Support Vector Machines (SVMs) είναι ένας τύπος αλγορίθμου ταξινόμησης πιο ευέλικτος καθώς έχει την δυνατότητα να ταξινομήσει τα δεδομένα μας γραμμικά , μπορεί όμως να κάνει και χρήση μη γραμμικών συναρτήσεων εφόσον το πρόβλημα ταξινόμησης το απαιτεί.

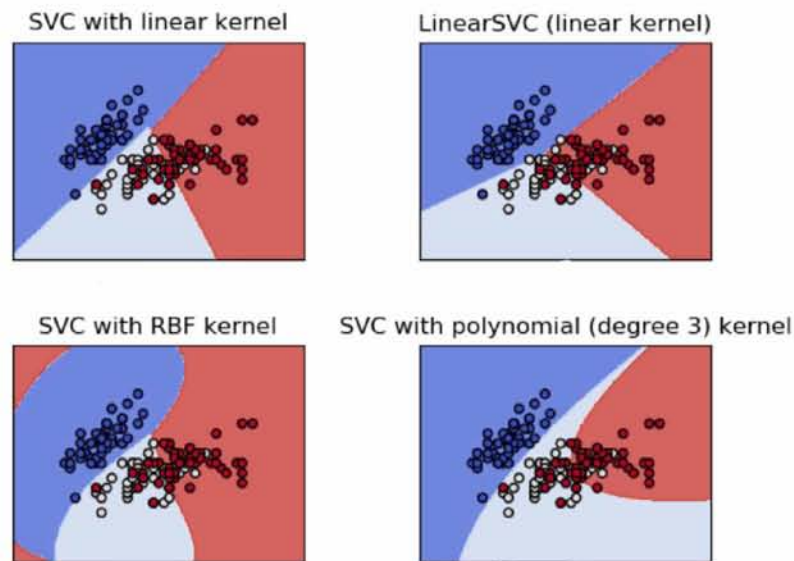
Τα πλεονεκτήματα των Support Vector Machines είναι:

1. Αποτελεσματικό σε πολυδιάστατους χώρους.
2. Αποτελεσματικό επίσης σε περιπτώσεις όπου ο αριθμός των διαστάσεων είναι μεγαλύτερος από τον αριθμό των δειγμάτων.
3. Επειδή χρησιμοποιεί υποσύνολο των σημείων εκπαίδευσης στην συνάρτηση απόφασης (decision function), τα διανύσματα υποστήριξης (support vectors), είναι πολύ αποδοτικό σε μνήμη.
4. Τέλος, είναι ένας ευέλικτος αλγόριθμος καθώς παρέχονται έτοιμοι κοινοί πυρήνες ή μπορούμε να καθορίσουμε προσαρμοσμένους πυρήνες για την λειτουργία της απόφασης.

Στα μειονεκτήματα:

1. Όταν ο αριθμός των χαρακτηριστικών είναι πολύ μεγαλύτερος από τον αριθμό των δειγμάτων, το μοντέλο μπορεί να οδηγηθεί σε overfitting και πρέπει να είμαστε προσεκτικοί στην επιλογή των λειτουργιών του πυρήνα, ενώ ακόμη, ο όρος της κανονικοποίησης είναι επίσης κρίσιμος.
2. Δεν παρέχουν άμεσα εκτιμήσεις πιθανότητας, αυτές πρέπει να υπολογιστούν χρησιμοποιώντας μια ακριβή πενταπλή διασταυρούμενη επικύρωση (five-fold cross-validation)
3. Τα SVMs σε scikit-learning υποστηρίζουν τόσο πυκνά όσο και αραιά δείγματα διανυσμάτων ως είσοδο. Ωστόσο, για να το χρησιμοποιήσουμε για να κάνουμε προβλέψεις για αραιά δεδομένα, πρέπει να ταιριάζει σε τέτοια δεδομένα. Για βέλτιστη απόδοση, χρησιμοποιούμε sparse matrix . Τα SVC , NuSVC και LinearSVC είναι τάξεις ικανές να εκτελούν δυαδική και πολλαπλών κατηγοριών ταξινόμηση σε ένα σύνολο δεδομένων.

Από το παραπάνω συμπεραίνουμε ότι για την επίλυση του δικού μας προβλήματος, δηλαδή την ταξινόμηση ρευμάτων , ο αλγόριθμος των Support Vector Machines είναι αρκετά χρήσιμος.

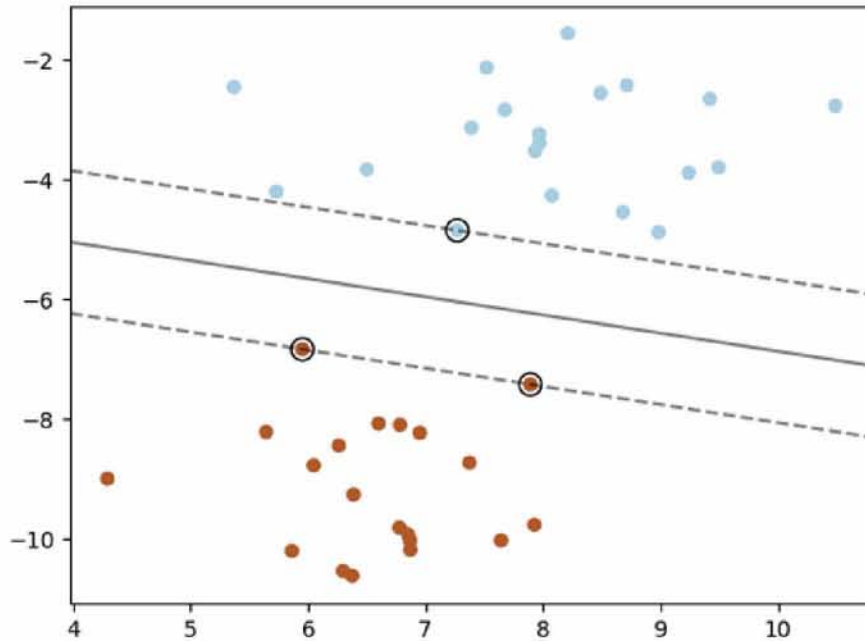


Σχήμα 3.2: Support Vector Machines

Τα SVC και NuSVC είναι παρόμοιες μέθοδοι, αλλά δέχονται ελαφρώς διαφορετικά σύνολα παραμέτρων και έχουν διαφορετικές μαθηματικές διατυπώσεις. Από την άλλη πλευρά, το LinearSVC είναι μια άλλη ταχύτερη εφαρμογή του Support Vector Classification για την περίπτωση ενός γραμμικού πυρήνα. Το LinearSVC δεν δέχεται τον πυρήνα παραμέτρων, καθώς θεωρείται ότι είναι γραμμικό. Δεν διαθέτει επίσης ορισμένα από τα χαρακτηριστικά των SVC και NuSVC, όπως το support με το οποίο μπορούμε να συλλέξουμε χρήσιμες πληροφορίες. (scikit learn)

Μαθηματική Διατύπωση:

Το SVM κατασκευάζει ένα υπερ-επίπεδο ή ένα σύνολο υπερ-επιπέδων σε έναν υψηλό ή άπειρο-διάστατο χώρο, ο οποίος μπορεί να χρησιμοποιηθεί για ταξινόμηση, regression ή άλλες εργασίες. Διαισθητικά, ένας καλός διαχωρισμός επιτυγχάνεται από το υπερ-επίπεδο που έχει τη μεγαλύτερη απόσταση από τα πλησιέστερα σημεία δεδομένων εκπαίδευσης οποιασδήποτε κλάσης (το λεγόμενο functional margin - λειτουργικό περιθώριο), καθώς γενικά όσο μεγαλύτερο είναι το περιθώριο τόσο μικρότερο είναι το σφάλμα γενίκευσης του ταξινομητή. Το παρακάτω σχήμα δείχνει τη συνάρτηση απόφασης για ένα γραμμικά διαχωρίσιμο πρόβλημα, με τρία δείγματα στα όρια του περιθωρίου, που ονομάζονται support vectors (διανύσματα υποστήριξης). Γενικά, όταν το πρόβλημα δεν διαχωρίζεται γραμμικά, οι φορείς υποστήριξης είναι τα δείγματα εντός των ορίων περιθωρίου



Σχήμα 3.3: συνάρτηση απόφασης για ένα γραμμικά διαχωρίσιμο πρόβλημα, με τρία δε-
 ίγματα στα όρια του περιθωρίου, που ονομάζονται 'διανύσματα υποστήριξης' "Support
 Vectors":

3.2.3 Random Forests

Είναι και αυτή μια μέθοδος εκμάθησης για ταξινόμηση. Γενικά, ακριβώς όπως σε ένα πραγματικό δάσος, όσο περισσότερα δέντρα διαθέτει, τόσο πιο ισχυρό φαίνεται, ομοίως, στον τυχαίο ταξινομητή δασών, όσο αυξάνονται τα 'δέντρα', τόσο μεγαλώνει και η ακρίβεια των αποτελεσμάτων. Δημιουργία τυχαίου δάσους

Βήμα 1: Δημιουργία ενός συνόλου δεδομένων Bootstrapped

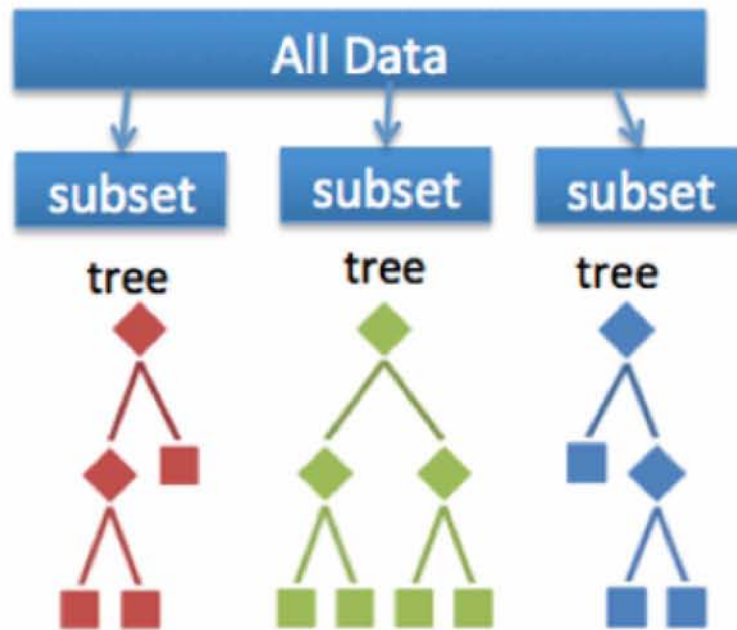
Βήμα 2: Δημιουργία δέντρων απόφασης

Η επόμενη δουλειά μας είναι να δημιουργήσουμε ένα δέντρο αποφάσεων χρησιμοποιώντας το σύνολο δεδομένων bootstrapped που δημιουργήθηκε στο προηγούμενο βήμα. Δεδομένου ότι κάνουμε ένα τυχαίο δάσος δεν θα λάβουμε υπόψη ολόκληρο το σύνολο δεδομένων που δημιουργήσαμε, αλλά θα χρησιμοποιήσουμε μόνο ένα τυχαίο υποσύνολο μεταβλητών σε κάθε βήμα.

Βήμα 3: Επιστρέψτε στο βήμα 1 και επαναλάβετε

Όπως ανέφερα νωρίτερα, το Random Forest είναι μια συλλογή από Δέντρα Απόφασης. Κάθε απόφαση απόφασης προβλέπει την κλάση εξόδου με βάση τις αντίστοιχες μεταβλητές πρόβλεψης που χρησιμοποιούνται σε αυτό το δέντρο.

Τέλος, καταγράφεται το αποτέλεσμα όλων των δέντρων απόφασης σε ένα τυχαίο δάσος και η τάξη με την πλειοψηφία υπολογίζεται ως κατηγορία εξόδου. Έτσι ακολουθώντας τα παραπάνω βήματα, το Random Forest μας θα μοιάζει με αυτό: Βήμα 4: Πρόβλεψη του αποτελέσματος ενός νέου σημείου δεδομένων



Σχήμα 3.4: Random Forests

3.2.4 Neural Network

Ένα νευρωνικό δίκτυο αποτελείται από κόμβους και νευρώνες. Είναι ένας αλγόριθμος μηχανικής μάθησης που περιλαμβάνει την τοποθέτηση πολλών κρυφών επιπέδων. Κάθε επίπεδο αντιστοιχεί και σε έναν νευρώνα. Αυτοί οι νευρώνες συνδέονται μεταξύ τους με activation functions (συναρτήσεις ενεργοποίησης) προκειμένου είτε να προβλέψουν είτε, στην περίπτωση μας, να μοντελοποιήσουν ή ταξινομήσουν δεδομένα. Στην εφαρμογή έγινε χρήση της sklearn βιβλιοθήκης, παρά το γεγονός ότι το Tensorflow και το Keras είναι πιο κατάλληλες για την προσαρμογή νευρωνικών δικτύων.

3.2.5 Multi class Classification

Ενώ η δυαδική ταξινόμηση από μόνη της είναι εξαιρετικά χρήσιμη, υπάρχουν στιγμές που θα θέλαμε να μοντελοποιήσουμε και να προβλέψουμε δεδομένα που έχουν περισσότερες από δύο κατηγορίες. Πολλοί από τους ίδιους αλγόριθμους μπορούν να χρησιμοποιηθούν με μικρές τροποποιήσεις για την ταξινόμηση σε περισσότερες από δύο ομάδες.

3.3 Εφαρμογή

[5]

1. Εισαγωγή βιβλιοθηκών

```
import pandas as pd
import numpy as np
```

2. Μετατροπή εισόδου απο 0 και 1 σε τεσσερις τιμές που δείχνουν τις μεταβάσεις των εισόδων.

```
def transition(x):
    if x == 1:
        return 0
    else:
        return 1

def input_transition(inputs):
    i = 1
    inputs_transition = [[0], [0], [0], [0], [0], [0], [0]]
    while i < len(inputs):
        j = 0
        print (i)
        while j < 7:
            if inputs[i-1][j] < inputs[i][j]:
                inputs_transition[j].append(2)
            elif inputs[i-1][j] > inputs[i][j]:
                inputs_transition[j].append(-1)
            elif inputs[i-1][j] == inputs[i][j] == 0:
                inputs_transition[j].append(0)
            else:
                inputs_transition[j].append(1)
            j += 1
        i += 1
    return inputs_transition
```

3. Για την γραφική απεικόνιση των τιμών:

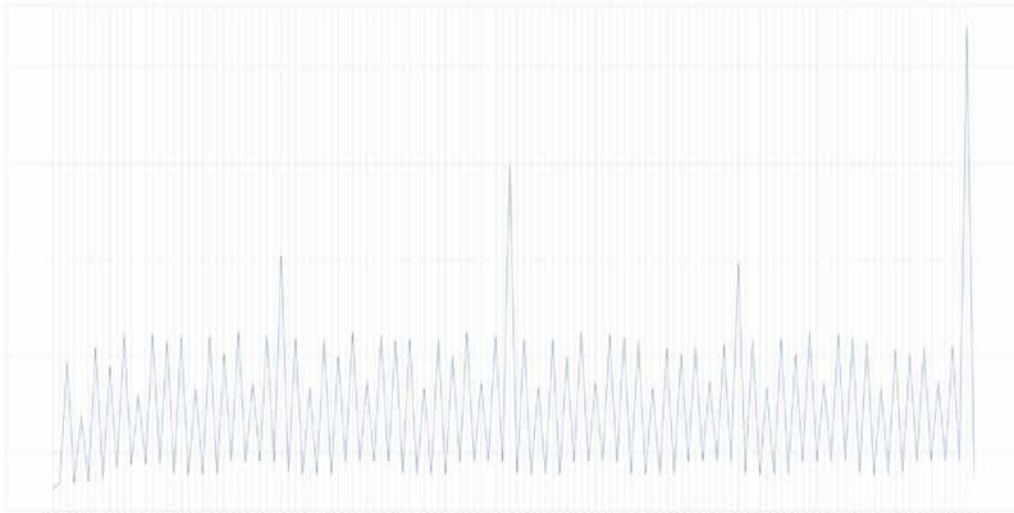
```
import matplotlib.pyplot as plt

fig= plt.figure(figsize=(6,3))

axes = fig.add_axes([5, 5, 10, 10])

axes.plot(moments,max_current)

plt.show()
```



Σχήμα 3.5: Διάγραμμα χρόνου - ρεύματος

όπου `moments` οι χρονικές στιγμές και `max_current` το ρεύμα κάθε χρονική στιγμή

4. Διαχωρισμός σε Train - Test Set:

```
X_train, X_test, y_train, y_test = train_test_split(
    final_input, max_current, test_size=0.1)
X_train = np.array(X_train)
X_test = np.array(X_test)
y_train = np.array(y_train)
y_test = np.array(y_test)

print (X_train.shape, y_train.shape)
print (X_test.shape, y_test.shape)
```

5. Cross Validation:

```
from sklearn.model_selection import KFold #import KFold
kf = KFold(n_splits=4) # Define the split - into 2 folds
kf.get_n_splits(current_transitions) #returns the number of splitting
iterations in the cross-validator

for train_index, test_index in kf.split(current_transitions):
    print("TRAIN:", train_index, "TEST:", test_index)
    X_train, X_test = current_transitions[train_index],
    current_transitions[test_index]
```

6. Δημιουργία εξόδων με κατηγορηματικές τιμές

Βρίσκω 2 τιμές και φτιάχνω 3 ομάδες. Οι τιμές των `thresholds` έχουν επιλεγεί για να είναι διαμοιρασμένα τα αποτελέσματα και αλλάζουν ανάλογα το σύνολο

δεδομένων.

```

low_threshold = 180.000000
high_threshold = 300.000000
threshold = 786.0095063

y_logistic = [0] * len(max_current)
for i in range(len(max_current)):
    if max_current[i] < low_threshold:
        y_logistic[i] = 0
    elif max_current[i] < high_threshold:
        y_logistic[i] = 1
    else:
        y_logistic[i] = 2

```

7. Χρήση μοντέλων ταξινόμησης:

```
model = linear_model.LinearRegression()
```

(a) Logistic Regression

```
model = LogisticRegression(solver='liblinear',
random_state=0)
```

(b) Linear Support Vector Machines

```
model = svm.LinearSVC()
```

(c) MultiClass Support Vector Machines

```
model = svm.SVC(decision_function_shape='ovo')
```

(d) Random Forests

```

from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100, max_depth=2,
random_state=0)

```

8. Fitting - Predicting

```

model.fit(X_train, y_train)
model.predict(X_test)

```

Κεφάλαιο 4

Αποτελέσματα

Τα παρακάτω ποσοστά επιτυχών ταξινομήσεων είναι αποτέλεσμα μιας εκπαίδευσης των μοντέλων με ένα σύνολο 130 δεδομένων, τα οποία φυσικά συμφωνούν και με μεγαλύτερα σύνολα δεδομένων.

1. Για τον αλγόριθμο Logistic Regression χρησιμοποίησα αρχικά σαν είσοδο την κατάσταση των παλμών (0 ή 1) ,σαν έξοδο τις κανονικοποιημένες τιμές των ρευμάτων (3 κατηγορίες - τιμές που είχαν αποφασιστεί βάση 2 ορίων) και χωρίζοντας τα δεδομένα μου σε Train και Test με ποσοστό 90-10. Με τα παραπάνω , το μοντέλο παράγει ένα ποσοστό επιτυχημένων ταξινομήσεων της τάξης του 83%.

Χρησιμοποιώντας στο ίδιο μοντέλο, την ίδια έξοδο και με είσοδο την μετάβαση των παλμών, η εκπαίδευση οδηγεί σε overfitting .

2. Για το μοντέλο των SVM (Support Vector Machines) και συγκεκριμένα το Linear SVC, έτρεξα και τους 4 συνδιασμούς εισόδου και εξόδου.
 - (α') Με είσοδο 0 ή 1 και έξοδο τις πραγματικές τιμές των ρευμάτων , το ποσοστό επιτυχίας είναι 84%.
 - (β') Ενώ εφαρμόζοντας το μοντέλο στις άλλες τρεις περιπτώσεις συνόλων δεδομένων, το ποσοστό επιτυχημένων ταξινομήσεων ανεβαίνει στο 92.3%.
3. Όσον αφορά την απλή εκδοχή του SVM, με τα παρακάτω χαρακτηριστικά:
SVC(C=1.0,
cache_size=200,
class_weight=None,
coef0=0.0,
decision_function_shape='ovr',
degree=3,
gamma='auto_deprecated',
kernel='rbf', max_iter=-1,
probability=False,
random_state=None,
shrinking=True,
tol=0.001,
verbose=False):

- (α') Για είσοδο 0 ή 1 και έξοδο τις πραγματικές τιμές των ρευμάτων , το ποσοστό είναι 93.2%
- (β') Για είσοδο -1, 0, 1, 2 και έξοδο τις πραγματικές τιμές των ρευμάτων, το ποσοστό είναι 76.9%.
- (γ') Για είσοδο 0 ή 1 και έξοδο κανονικοποιημένες τιμές ρευμάτων, το ποσοστό είναι 69,2%.
- (δ') Τέλος , για είσοδο -1. 0, 1, 2 και έξοδο κανονικοποιημένες τιμές ρευμάτων το μοντέλο κάνει overfitting.

4. Στο μοντέλο ταξινόμησης Random Forests :

- (α') Για είσοδο 0 ή 1 και έξοδο τις πραγματικές τιμές των ρευμάτων , το ποσοστό είναι 92.3%.
- (β') Για είσοδο -1, 0, 1, 2 και έξοδο τις πραγματικές τιμές των ρευμάτων, το μοντέλο κάνει overfitting.
- (γ') Για είσοδο 0 ή 1 και έξοδο κανονικοποιημένες τιμές ρευμάτων, το ποσοστό είναι 84,6%.
- (δ') Το ίδιο ποσοστό (84,6%) προέκυψε και για είσοδο -1. 0, 1, 2 και έξοδο κανονικοποιημένες τιμές ρευμάτων .

5. Τέλος, μετά απο διάφορες δοκιμές σε νευρωνικά δίκτυα κατέληξα σε ένα μοντέλο:

```
MLPClassifier(
solver='lbfgs',
alpha=1e-5,
hidden_layer_sizes=(5, 2),
random_state=1)
```

- (α') Για είσοδο 0 ή 1 και έξοδο τις πραγματικές τιμές των ρευμάτων , το ποσοστό είναι 53.85%
- (β') Για είσοδο -1, 0, 1, 2 και έξοδο τις πραγματικές τιμές των ρευμάτων, το ποσοστό είναι 84.62%.
- (γ') Για είσοδο 0 ή 1 και έξοδο κανονικοποιημένες τιμές ρευμάτων, το ποσοστό είναι 76.92%.
- (δ') Τέλος , για είσοδο -1. 0, 1, 2 και έξοδο κανονικοποιημένες τιμές ρευμάτων το ποσοστό είναι 61,54%.

Πίνακας Αποτελεσμάτων				
Algorithms	1o dataset	2o dataset	3o dataset	4o dataset
Logistic Regression	-	83%	-	overfitting
Support Vector Machines - Linear SVC	84%	93.2%	93.2%	93.2%
Support Vector Machines - Simple SVM	93.2%	76.9%	69.2%	overfitting
Random Forests	92.3	overfitting	84.6%	84.6%
Neural Network	53,85%	84.62%	76.92%	61.54%

Table 4.1: Πίνακας Αποτελεσμάτων

Κεφάλαιο 5

Επίλογος

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας γίνεται μία πρώτη ανάλυση της χρησιμότητας των αλγορίθμων machine learning στα δίκτυα τροφοδοσίας ολοκληρωμένων. Σε μεγάλα κυκλώματα η ανάλυση και η μοντελοποίηση των ρευμάτων περιπλέκεται, εξαιτίας του ότι τα μέγιστα ρεύματα που διαρρέουν τις συνδεδεμένες μονάδες είναι κατανεμημένα ως προς το χρόνο. Αυτό, έχει ως αποτέλεσμα τα εργαλεία που χρησιμοποιούνται για τον σκοπό αυτό να πρέπει να έχουν την ικανότητα να αναλύουν εκατομμύρια πλέγματα αντιστατών. Διαφαίνεται στο σημείο αυτό, η ανάγκη μελέτης με τεχνικές machine learning. Από τα αποτελέσματα που προέκυψαν, κατόπιν της εφαρμογής διαφόρων αλγορίθμων ταξινόμησης, παρατηρούμε πως η κατηγοριοποίηση των ρευμάτων σε σχέση με την είσοδο τους μπορεί να φτάσει σε υψηλά επίπεδα ακρίβειας. Καθώς μεγαλώνει το σύνολο δεδομένων και με την προσθήκη επιπλέον εισόδων, τα ποσοστά επιτυχίας θα μπορούσαν να βελτιωθούν ακόμα περισσότερο.

Σε δεύτερο χρόνο θα ήταν ενδιαφέρον να μελετηθεί και η πτώση τάσης στα δίκτυα τροφοδοσίας ολοκληρωμένων. Δεδομένων όλων των κινδύνων που οφείλονται στην πτώση τάσης IR, θα ήταν χρήσιμο να εφαρμόσουμε τεχνικές μηχανικής μάθησης για την πρόβλεψη της, ενώ οι ταξινομήσεις των ρευμάτων που προέκυψαν από την διπλωματική εργασία, θα μπορούσαν, επίσης, να χρησιμοποιηθούν.

Βιβλιογραφία

- [1] *Star-Hspice Manual*. Addison-Wesley, 1998. Chap. 5.
- [2] BORIVOJE NIKOLIC english JAN M. RABAEY ANANTHA CHANDRAKASAN. *Digital Integrated Circuits*. Klidarithmos, 2006. ISBN: 9602099828.
- [3] Dimitrios Karabatzakis. *Analysis methods for CMOS integrated circuits reliability issues*. 2009.
- [4] URL: <https://www.edureka.co/>.
- [5] URL: https://scikit-learn.org/stable/supervised_learning.html#supervised-learning.
- [6] Wojciech Giziewicz. *Brief Introduction to HSPICE Simulation*.
- [7] *HSPICE by Synopsys*. URL: <https://personal.utdallas.edu/~Xiangyu.Xu/hspice/>.
- [8] Sreeram Chandrasekar Nithin S K Gowryankar Shanmugam. *Dynamic Voltage (IR) Drop Analysis and Design Closure: Issues and Challenges*. IEEE.
- [9] *SPICE Simulation Overview*. URL: <https://www.ni.com/en-us/innovations/white-papers/06/spice-simulation-overview.html>.