Implementation of Naive Bayes' Classifier on hardware Using FPGA

Submitted by

Nauman Memon

NUST00000118431MSEE17457F

Supervisor:

Dr. Sajjad Haider Zaidi

Department of Electronics and Power Engineering

Pakistan Navy Engineering College (PNEC)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

June 2019

Implementation of Naive Bayes' Classifier on hardware Using FPGA

Submitted by

Nauman Memon

Supervisor:

Dr. Sajjad Haider Zaidi

This project report is presented for qualifying MS degree in Electrical (Control) Engineering

Department of Electronics and Power Engineering

Pakistan Navy Engineering College (PNEC)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

June 2019

Declaration

I, *Nauman Memon* proclaimed that the thesis titled "Implementation of Naive Bayes' Classifier on hardware Using FPGA" and the work given in this thesis is my own and has been produced by me as a result of my own research work.

I confirm that:

1.

2. The research was carried out for pursuance of a Mater degree at National University of Science and Technology.

3. No part of or content in this thesis have been done as a work of any earlier research or degree, where in so, it has been clearly attributed.

4. The contribution of this work has been clearly specified where as any other work or source to support this thesis has been clearly mentioned.

5. I have recognised all the primary roots of assistance.

Nauman Memon,

NUST00000118431MSMEE17457F

Copyright Notice

This thesis is dedicated to *my beloved parents*

Abstract

The focus of this thesis is on the implementation of machine learning algorithm on hardware using FPGA. The machine learning algorithm that is used here is Naive Bayes' classifier for classifying linearly independent classes. NBC is one of the most successful supervised learning tool, by knowing the prior knowledge and likelihood of an event one can estimate the probability of given class, but assumes that features vectors are linearly independent of each other, not only this but also it is proved quite effective where the amount of data set is large. In this thesis the applications having continuous and discrete attributes are used, so an appropriate probability distribution function may be used for this purpose, if the attributes are continuous then Gaussian distribution function is used for finding the probabilities. Initially a continuous signal is pre-processed in order to convert it into a feature vector of pre-defined length then NBC is applied on it to classify the class based on their prior probabilities using FPGA hardware. We have implemented different examples based on their probability distribution function and results show that NBC is a reliable and fast learning method.The design is carried out in Simulink environment where as it's implementation is performed on qkintex-7 FPGA hardware using HDL coder and fixed point tool.

‘ **Keywords:** *FPGA, NBC, HDL Coder, FPT, CVA, RMS, EEG, VLSI Technology, Conditional Probability*

Acknowledgments

I am highly thankful to Allah (SWT) for providing me such great opportunity to pursue my research project. I would also like to show my gratitude to my supervisor Dr. Sajjad Haider Zaidi for his patience, motivation, enthusiasm and continuous support in completing my research. I am also indebted to those whom I worked with. Each and every member of GEC committee had been very kind and supportive to provide me professional help while carrying on research. My sincere acknowledgements to my family members for their support in the starting and completion of this thesis.

My colleagues also supported and motivated me in completing my research work. My special thanks are extended to all of the faculty members for their assistance and support. I am also grateful to the National University of Science and Technology for providing such a pleasant environment and facilities to complete this research.

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations

FPGA	Field programmable gate array
NBC	Naive Bayes' Classifier
HDL	Hardware description language
FPT	Fixed point tool
FPDT	Fixed point data type
pdf	Probability distribution function
CVA	Co-efficient of varying amplitude
RMS	Root mean sqaure
EEG	Electroencephalography, Electroencephalographic
VLSI	Very large scale integrated

Chapter 1

Introduction

1.1 Motivation

As the data is growing exponentially everywhere, every 1.67 years the amount of data that is stored become doubles so there is need to handle and analyse such big data, that's why Machine learning becomes the most valuable tool to provide unlimited solutions for any kind of problems in many fields like data mining, computer science and other fields. Machine learning is a modern way to train the system from the data and discovers patterns to explain the world interactions with us. It then encapsulates data to create models for forecasting the unseen situations. Machine leaning algorithms learn how to perform a task by generalisation from the samples [1].There are many Machine learning algorithms such as Supervised learning, Unsupervised learning, Semi-supervised learning, Reinforcement learning, Multitasking learning, Neural Network and Instance based learning. Supervised learning refers to algorithm which needs external assistance like we have input variable and output variable and the objective to use mapping function so well that input data predicts the output, in unsupervised learning we have only input data and no corresponding output, it learns few features from data but whenever new data comes it uses previously learned features to recognize class of data. Semi-supervised learning technique is the intermediate state between supervised learning and unsupervised learning and it is used where the un-labeled data is available but difficult to achieve labeled data like generative model, self-training and etc [2]. A reinforcement learning is of controlling the data unlike supervised learning the labelled data is given but reinforcement is about taking the most appropriate action to amplify reward in a specific situation so it learns from the experience rather than training data [3], the other type of machine learning is multitasking learning as the name suggests it performs the multi learning tasks simultaneously and its basic aim is to generalize the performance by grasping information that a signal contains in training of related tasks parallelly using a shared representation [4] and one of the application of multitasking learning is given in "Task clustering and Gating for Bayesian Multitask learning" [5] in which general multi analysis is implemented results in fixed effects and random effects, fixed effects are same for all tasks where as random effects may change between tasks so the Bayesian approach is used, Whereas few model parameters are shared and rest are loosely interlinked through joint prior distribution which can be grasped from data, so two parts can be combined, one statistical multilevel approach and other is neural network in order to achieve the best performance.

In this thesis a well-known supervised learning algorithm "Naive Bayes' Classifier" is used. Naive Bayes' Classifier is a binary classifier based on Bayes' theorem used to predict the probability of given class based prior knowledge but assumed that all the events are independent of each other despite of such assumption it is proved to be quite efficient in various problems, particularly in a case where the data set is quite large and fast learning is required. Although the independence of events is practically indigent assumption yet it behaves practically smarter [6] and achieve the goals with above 90 percent of probability. The Naive Bayes' classifier is said to be optimal if its events are independent of each other but empirically it performs well even violating the basic condition of independence to a great margin and some problems are proved in [7] for some feature dependencies and it is proved in [7] that the degree of accuracy of classifier is not directly dependent on degree of feature dependence so the Naive Bayes' classifier works best in two cases that is first when completely independent features as expected and second when functionally dependent features which is not expected, as these are two extremes where classifier performs the best and the worst, now here the basic question is that which feature dependencies can be considered and which can be neglected? The answer is the dependencies that do not providing information about class or that do not help distinguishing between different classes [7]. There are two models of classifier that are discriminative learning and generative learning, discriminative learning model learns map directly from input 'x' to output label 'y' or learns directly a posterior p(y/x), where as generative learning model learn a model of joint probability, p(x,y) and then make a decision by adopting Bayesian rules to evaluate p(y/x), then taking the decision which is most likely so a comparative analysis showed that a discriminative and generative have its own pros and cons. Discriminative is more preferably used rather than that of generative because one should solve a classification problem directly rather than going in general classification problem which may cause in handling missing data hence the discriminative has lower asymptotic error that of generative [8]. In this thesis we focus on generalize implementation of Naive Bayes' classifier on FPGA using HDL coder and fixed-point tool and any kind of classification can be done using Naive Bayes' classifier whether it is forecasting of event or classification any disease, initially we have implemented different classification examples in Matlab Simulink environment so that we can move to any kind of hardware that is reliable and according to given requirement, we shall show in this thesis though using fixed point tool box which converts the floating point numbers in fixed word length and fixed fractional length achieving the ninety six percent of efficiency.

1.2 Thesis Organization

The organization of thesis is arranged as follows

In chapter 2 Concept building A brief background is given and concept of Naive Bayes' Classifier and its types based on probability distribution function is built. A brief overview on Bayesian theorem, Conditional Probability, Fixed-Point data type and HDL coder is discussed.

In chapter 3 Design It discusses the algorithm design, methodology and analysis using Matlab and Simulink. Different application examples are classified using NBC on Matlab and Simulink. Results with the highest accuracy are elaborated.

In chapter 4 Implementation It is the core of this thesis.It discusses the algorithm flow chart and Naive Bayes' classifier is implemented on q-kintex 7 defence vesion FPGA using HDL coder and fixed-point tool.

In chapter 5 Discussion Future works & Conclusion It defines a nut shell summary of this research and future work.

Chapter 2

Concept Building

2.1 Conditional Probability

Conditional probability can be elaborated as the probability of an event but conditionally another event had already taken place, in other words it is the likely hood of an event considering that another event had already occurred. Conditional probability is given by,

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{2.1.1}$$

Example 2.1.1. A packet contains 4 red balls and 6 white balls. we randomly pick two balls from the packet. What is the probability that second ball is red assuming that first ball is white?

There are six white and four red balls and randomly two balls are taken given that first ball is white and the probability of red ball given that first ball is red defined in the below Fig.2.1

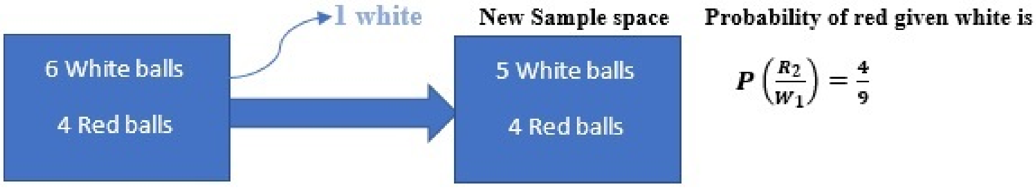

Figure 2.1: probability that second ball is red assuming that first ball is white

2.1.1 Conditional Independence

The notion of conditional autonomy is of great importance and it is the foundation of many statistical models. In probability theory, two events are independent If the knowledge of one occurrence does not alter the likelihood of the other. We can say mathematically two A and B are independent if and only if,

$$P(A|B) = P(A) \tag{2.1.2}$$

$$P(B|A) = P(B). \tag{2.1.3}$$

Example 2.1.2. When the coin is tossed , the probability that coin shows "head" is 0.5 and what if was Sunday? would this alter the probability of getting "head?" naturally not. If it is a Sunday then probability of obtaining head, is still 0.5. A day was Sunday and consequence of a tossed coin are independent events because an event its a Sunday did not alter the other event that is a probability of getting "head".

Example 2.1.3. A survey had been carried out by researchers that latest degree holders from two distinct universities in their annual revenue. The below table data shows data for 300 graduates who responded.

Annual Revenue	Institute A	Institute B	Total
Less than \$20,000	36	24	60
\$20,000 to 39,999	109	56	165
\$40,000 and over	35	40	75
Total	180	120	300

Table 2.1: Survey Data.

- Let's select from this data a random graduate.
- Are these two events "Income is \$40,000 and over" and "Attended university B" not dependent?
- Let's check on the conditional probability whether these events are independent or not.
- From the table 2.1 there are 75 out of 300 graduates who say their income is \$40,000 or above

$$P(\$40,000andover) = \frac{75}{300} = 0.25 \tag{2.1.4}$$

Now let us check What is the likelihood of a randomly chosen graduate earning $40,000 and over provided that they come from University B?

From the table 2.1 there are total 40 graduates who attended university B and having income of $40,000 and over.

$$P(\$40,000andover|Uni.B) = \frac{40}{300} = 0.1333 \tag{2.1.5}$$

We can conclude from the result that events are not independent because an event affect the probability of other event.

2.1.2 Probabilistic Concepts & Bayes' Theorem

Probability is one of the main core of data science algorithm, in fact the solution of many problems of data science are probabilistic in nature, thus concentrating on learning statistics and probability before going to algorithms. Before explaining the Bayes' theorem, it is necessary to understand basic probabilistic terminologies.

Probability of independent events

Let there are two occurrences 'A' and 'B', then probability of independent events can be calculated as

$$P(A \cap B) = P(A).P(B) \tag{2.1.6}$$

Example 2.1.4. A box containing four red marbles and three black marbles, if we take a red marble and a coin is flipped we get a head. What is the probability of wining?

Solution: There are two events, let A be the event of getting red marble and B be the event of getting head when coin is flipped.

$$P(A) = \frac{4}{7} \tag{2.1.7}$$

$$P(B) = \frac{1}{2} \tag{2.1.8}$$

We know that both events are independent so probability of wining is

$$P(A \cap B) = P(A).P(B) \tag{2.1.9}$$

$$= \frac{4}{7} \times \frac{1}{2} \tag{2.1.10}$$

$$= \frac{2}{7} \tag{2.1.11}$$

Probability of dependent events

What if the events are dependent? For the example 2.1.1, suppose 'A' be the event of getting red marble from the box. Let 'B' be the another event, that takes place only after event 'A', of getting another red marble. Would the probability of events 'B' and 'A' be the same? No. For event 'A' the chances were $\frac{4}{7}$ where as for event 'B' the chances are $\frac{3}{6}$. This is because after event 'A', there are only 6 marbles left in the box out of which three are red and three are black

Marginal Probability

Marginal Probability is the probability of single event 'A' that does not depend upon any other event 'B'

$$P(A) = P(A|B) \times P(B) + P(A| \sim B) \tag{2.1.12}$$

Another way of writing equation 2.1.11 is,

$$P(A) = P(A \cap B) + P(A \cap \sim B) \tag{2.1.13}$$

Where $\sim B$ indicates that event 'B' does not take place.

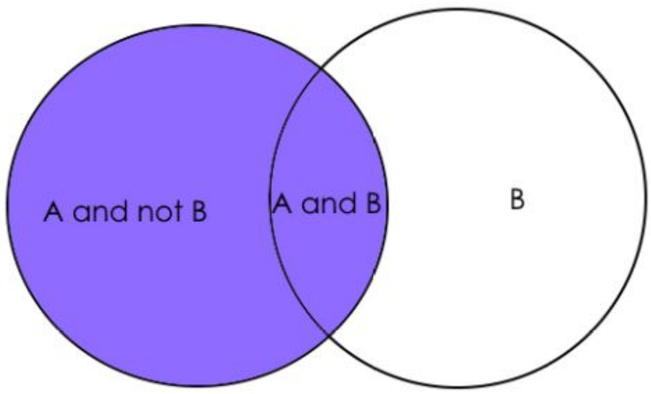

Figure 2.2: Venn Diagram Illustration of Marginal Probability

Bayes' Theorem

Modern science is about data and predictions. Observing, collecting knowledge and predicting is the scientific phenomena. Now question is how accurate is prediction? This relies on the quality of the information presented and the precision of the observations.

For example: weather forecasting, the more we understand how the weather changes, the better one can predict whether it will be sunny or rainy tomorrow, current observations and seasonal records and the weather model can be refined by any discrepancy between prediction and observation.Bayesian statistics encapsulate this process of applying prior theoretical and empirical knowledge to formulate hypotheses [9]. In other words Bayes' theorem transforms the outcomes of your test into the actual probability of the event. The mathemetically definition of Bayes theorem is,

$$P(X|Y) = \frac{P(X|Y) \times P(X)}{P(Y)} \tag{2.1.14}$$

Where 'X' and 'Y' represent separate events and $P(Y) \neq 0$.

- P(X|Y) is the probability of event 'X' given event 'Y' that is, its a conditional probability.
- P(Y|X) is the conditional probability of event 'Y' given event 'X'.
- P(X) and P(Y) are the marginal probabilities of independent events 'X' and 'Y' respectively.

Proof:

Bayes' theorem relates different conditional probabilities. Conditional probability expresses how likely it is that an event is going to take place with the condition that another event has already taken place.

Conditonal probability can be found as,

$$P(X|Y) = \frac{P(X \cap Y)}{P(Y)} \tag{2.1.15}$$

Since the probability $P(X \cap Y)$ is probability of event 'X' occurring times probability of event 'Y' given 'X'. This is equivalent to saying that $P(X \cap Y)$ is probability of event 'Y' occurring times probability of event 'X' given 'Y'. This equivalency of the two expression leads to Bayes' theorem.

$$\begin{aligned}
P(X|Y) &= \frac{P(X \cap Y)}{P(Y)}, if P(Y) \neq 0 && (2.1.16)\\
P(Y|X) &= \frac{P(Y \cap X)}{P(X)}, if P(X) \neq 0 && (2.1.17)\\
\Rightarrow P(X \cap Y) &= P(X|Y) \times P(Y) && (2.1.18)\\
&= P(Y|X) \times P(X) && (2.1.19)\\
\Rightarrow P(X|Y &= \frac{P(Y|X) \times P(X)}{P(Y)}, if P(Y) \neq 0 && (2.1.20)
\end{aligned}$$

In simple words Bayes' theorem can defined as predicting a sample's category (such as news items or client reviews). The probability of a specified sample may be calculated for different categories and the final categorization of the sample will be done on the basis of highest probability. By utilizing an previous knowledge about the features of a sample, Bayes' theorem may be used to obtain the probabilities mentioned previously.

Example 2.1.5. Suppose there is a 40% probability that it rains on Sunday. Then, if it rains on Sunday, known as event 'A', there is a further 10% chance that it rains on Monday as well, known as event 'B'. However, if it does not rain on Sunday, there is then a 80% probability that it is going to rain on Monday. There are two events 'A' and 'B' that are 'raining on Sunday' and 'raining on Monday' respectively.What are the chances of occurrence of raining on Sunday, if it rained on Monday? Below given data represents the data of event 'A' and event 'B' as follows,

Solution:

- $P(A) = 0.4 \rightarrow$ probability that it is going to rain on Sunday.

- $P(A') = 0.6 \rightarrow$ probability that it doesn't rain on Sunday

- $P(B|A) = 0.1 \rightarrow$ probability that it is going to rain on Monday, but not on Sunday.

- $P(B'|A) = 0.9 \rightarrow$probability that it does not rain on Monday, but it is going to rain on Sunday.

- $P(B|A') = 0.8 \rightarrow$probability it is going to rain on Monday, but not on Sunday.

- $P(B'|A') = 0.2 \rightarrow$ probability that it doesn't rain on Monday and Sunday.

We would first like to calculate the probability if it is going to rain on Monday, which will be,

$$P(B|A) + P(B|A') \tag{2.1.21}$$

$$P(B) = 0.4 \times 0.1 + 0.6 \times 0.8 \quad (2.1.22)$$

$$P(B) = 0.52 \rightarrow probability of raining on Monday \quad (2.1.23)$$

In this problem we are only interested in the following Probabilities,

- $P(A) = 0.4$

- $P(B) = 0.56$

- $P(B|A) = 0.1$

Now we can find the probability that it is going to rain on Sunday provided that it rained on Monday as,

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \quad (2.1.24)$$

$$or P(A|B) = \frac{0.1 \times 0.4}{0.52} \quad (2.1.25)$$

$$P(A|B) = 0.0769 \quad (2.1.26)$$

It means there are 7.69% chances of raining on Sunday given that it would have rained on Monday

2.2 Naive Bayes' Classifier

The Bayes' theorem is the foundation of Naive Bayes' classifier, NBC predict the occurrence of an event based on the previous knowledge of an event. It is so called Naive Bayes because of assumption that it assumes all events are independent of each other [8]. Despite of such strong assumption that events should not depend on each other, it was proved to be quite effective in many problems particularly where there is big amount of data is needed and where fast learning is first priority [10] and [11]. The algorithm falls under the category of supervised learning and therefore requires a training input-output data set.

2.2.1 Probabilistic Model

This section shows the proposed probabilistic mathematical model of Naive Bayes Classifier.Let X be a set of training, where each instance is represented by the n-dimensional vector $(Y_1, Y_2, ..., Y_n)$.

The pre-defined set of classes is $C = c_1, c_2, c_3, ..., c_k$

Using Bayes' Theorem, the conditional probability is

$$Posterior = \frac{Prior \times Likelihood}{evidence} \tag{2.2.1}$$

$$Or\ p(C_k|y) = \frac{p(C_k)p(y|C_k)}{p(y)} \tag{2.2.2}$$

Since we are only interested in numerator of 2.2.2 cause the denominator that is $p(y)$ which is not class C dependent and feature values is provided that is (y_i) , thus $p(y)$ is consistent effectively. The numerator would be same as joint probability,

$$p(C_k, y_1, y_2, ..., y_n) \tag{2.2.3}$$

Use the chain rule for repeated conditional probability definition

$$\begin{aligned}
&p(C_k, y_1, y_2, ..., y_n) = p(y_1, y_2, ..., y_n, C_k)\\
&= p(y_1|y_2, ..., y_n, C_k)p(y_2, ..., y_n, C_k)\\
&= p(y_1|y_2, ..., y_n, C_k)p(y_2|y_3, ..., y_n, C_k)p(y_3, ..., y_n, C_k)\\
&= ....\\
&= p(y_1|y_2, ..., y_n, C_k)p(y_2|y_3, ..., y_n, C_k)...\\
&\qquad\qquad\qquad p(y_{n-1}|yn, C_k)p(y_n|C_k)p(C_k)
\end{aligned}$$

Now the Naive assumption comes into play which assumes that all features in Y are mutually independent, depending on category C_k. This results in the approximation

$$p(y_i|x_{i+}, ..., y_n, C_k) \approx p(y_i|C_k) \tag{2.2.4}$$

The joint probability model can be written as,

$$\begin{aligned}
p(C_k|y_1, ..., y_n)\alpha p(y_i|C_k)&\\
&\approx p(C_k)p(y_1|C_k)p(y_2|C_k)p(y_3|C_k)...\\
&= p(C_k)\prod_{i=1}^{n} p(y_i|C_k)
\end{aligned}$$

Under the Naive assumption, the conditional distribution over class C is

$$p(C_k|y_1, ..., y_n) = \frac{1}{Z} p(C_k) \prod_{i=1}^{n} p(y_i|C_k) \tag{2.2.5}$$

In above equation 'Z' is evidence, can be written as $(Z = p(y) = \sum_k p(C_k)p(y|C_k))$. Its scaling factor and only depends upon $(y_1, ..., y_n)$.

Buidling Classfier

From the above derivation that assumes feature independence, the Naive Bayes classifier combines this model with a decision rule (a function that formulate the observation into an appropriate action) and the standard rule is to select most likely hypothesis know as the maximum posteriori. The Bayes' classifier can be written as.

$$\hat{y} = \underset{k\epsilon 1,...,K}{argmax} \prod_{i=1}^{n} p(y_i|C_k) \tag{2.2.6}$$

Parameter Estimation

For Naive Bayes' a class's prior parameter may be determined as

$$(prior\ for\ a\ given\ class) = \frac{number\ of\ samples\ in\ the\ class}{total\ number\ of\ samples}$$

Now let's estimate the parameters for feature's distribution and one must consider a distribution models for the features from the training set [12].

Gaussian NBC

Gaussian distribution is also known as normal distribution, it came in play if the data attributes are continuous.

if a training set having the continuous 'y' attributes then one can find the parameters such as mean and variance of 'y' attributes in each class. Suppose mean is μ_k of y attributes with respect to class C_k , σ^2 is the variance of y attributes with respect to class C_k and we have assumed observed values v. The pdf of 'v' given a class C_k is,

$$P(y = v|C_k) = \frac{1}{\sigma_k \sqrt{2\pi}} e^{-(v-\mu_k)^2/2\sigma_k^2} \tag{2.2.7}$$

Multinomial Naive Bayes' Model

Using the model of a multinomial event, the feature vectors or samples represent the frequencies of certain events generated through a multinomial $(p_1, p_2, ..., p_n)$ with pi representing the probability of occurrence of the event i.

Having said that, a feature vector; say 'x' ($where x = (x_1, x_2, ..., x_n)$) is then, a histogram with each element x_i representing the number of times the event i has taken place. The multinomial distribution is a generalized form of the binomial distribution, which provides the likelihood of any specific combination of success numbers for the different classifications.

Such a model may be used for document classification. For this particular example, the events may occurrence of a specific word in the document. Probability of observing a histogram x is,

$$P(x|C_k) = (\frac{(\Sigma_i \times x_i)!}{\Pi_i \times x_i!}) \times \prod_i \times Pki^{x_i} \tag{2.2.8}$$

Probability estimates calculate to zero for features that actually never occur in a class. This is troublesome because the other non-zero probabilities will also become zero upon multiplication.

Therefore, a pseudocount is usually incorporated in the probabilities so that the probabilities are never zero. Regularizing the Naives Bayes in this way is known as Laplace Smoothing for a pseudocount of '1' and known as Lidstone Smoothing otherwise.

Bernoulli Naive Bayes' Model

Using the model of a multivariate Bernoulli event, the inputs are actually independent boolean variables having binary values. The model is popular for the tasks of document classification just as was the case in the multinomial event model. However, the difference is in the features that are considered as occurrence of a binary term rather than term frequencies.

This means that if xi represents a boolean, then it is the occurrence or absence of the i^{th} term of the vocabulary. We can write the likelihood of a document of a given class C_k as,

$$P(y|C_k) = \prod_{i=1}^{n} \times p_{ki}^{y_i} \times (1 - pki)^{1-y_i} \tag{2.2.9}$$

with pki denoting the C_k class probability of generating a x_i term. These features of the Bernoulli model make it specially suitable for classification of short texts. One can explicitly model the absence or occurrence of terms. The frequency counts in a Naives Bayes' Bernoulli event model, unlike the multinomial model, are truncated to one.

2.3 Fixed-Point Data Type

While dealing with digital systems, numbers are represented and stored in binary scheme. In order to represent binary numbers, can either be fixed-point or floating-point data types used. A fixed-point data type consists of three basic parts that are word length in bits, the position of the binary point, and it is signed or unsigned digit. A binary fixed-point number can be rewritten in general format as,

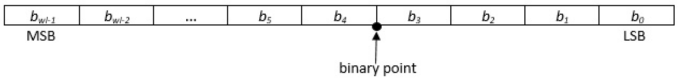

Figure 2.3: Binary representation of a generalized fixed-point number

Where

- b_i is called i^{th} binary digit.
- Word length in bits is represented as wl.
- The place of (MSB) is b_{wl-1} .
- The place of (LSB) is b_0 .
- As mentioned in above figure, the place of binary point is 4 places left of 'LSB'.

FPDT must be either signed or unsigned digit. Signed binary FP numbers may be represented in the following ways:

- Sign/magnitude
- one's Complement
- two's Complement

2's complement is one of the famous and commonly used representation of signed FP numbers and is the only representation that is being carried by Fixed-Point Designer documentation.

2.3.1 Scaling

Suppose 's' denotes slope, 'i' denotes and 'b' denotes bias, then fixed-point numbers are encoded as,

$$Real - WorldValue = (s \times i) + b \tag{2.3.1}$$

The slope can represented as,

$$s = slopeadjustmentfactor \times 2^{fixed-exponent} \tag{2.3.2}$$

The scaling of fixed-point number can be expressed by the 's' and 'b'. The scaling with a zero bias can be affected only by slope. A binary point position scales the fixed point number which is equivalent to [s b] with a null bias and slope adjustment factor should be unity. This is called a scaling by only a binary point.

$$Actual - Value - in - globe = (2^{fixed-exp}) \times interger \tag{2.3.3}$$

The FPT not only hold binary point scaling but also hold [Slope Bias] method scaling.

How to evaluate 's' and 'b'

Lets initiate with the terminal points that are required, signedness and 'wl'. Suppose upper-bound is represented by 'u-b' and lower-bound is represented by 'l-b'.

- $l - b = 999$
- $u - b = 1000$
- $signed = true$
- $wl = 16$

Let us specify an object 'fi' with a pre-defined 'wl' & singed-ness by using the range function,

$[Q_min, Q_max] = range(fi([], is_signed, word_length, 0))$.

In order to calculate 's' and 'b', system of equations needs to be solved,

$$l - b = s \times Q_min + b \tag{2.3.4}$$

$$u - b = s \times Q_max + b \tag{2.3.5}$$

Above equations can be written in matrix form as,

$$\begin{bmatrix} lower - bound \\ upper - bound \end{bmatrix} = \begin{bmatrix} Q_m in1 \\ Q_m ax1 \end{bmatrix} \times \begin{bmatrix} slope \\ bias \end{bmatrix} \tag{2.3.6}$$

2.3.2 Precision and Range

Range

The range can defined as the limit of numbers with which a FPDT and scaling can be represented. The below figure expresses the range of representable numbers with a specified 'wl' as word length, 's' as scaling and bias 'B' for a two's complement scheme.

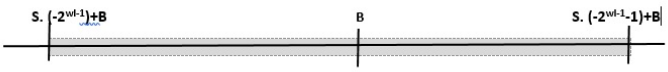

Figure 2.4: Fixed-point numbers for signed and unsigned data type, the distinct bit patterns is 2^{wl}.

Overflow Handling

Finite ranges are defined with fixed point data types. Range changes when the output of operation changes. Positive and negative overflows are given by saturation. Modulo arithmetic is a technique used in wrapping to put the overflow back into data type that can be represented.

Precision

There is a difference between continuous values in a fixed point number's precision that can be represented by data type and scaling that has same value as that of least significant bit. Fractional bit determines the least significant value.

2.3.3 Rounding Methods

When there is difference between the specified data type and scaling, rounding method is used to cast value to a representable number. Rounding operation always loses precision, it also determines the value of bias and operation cost. Fixed-Point tool currently supports the following rounding methods,

- Ceiling rounds in the direction of positive infinity to the nearest representable number.
- Convergent rounds to the nearest number representable and rounds to the nearest even number in the case of a tie.
- Fix rounds to the closest representable number in the direction of zero.
- Floor rounds to the closest representable number in the direction of negative infinity.
- Nearest rounds to the nearest number expressible. In the case of tie, the Nearest move towards the side of positive infinity .
- Round approximates to the most nearest representable number.

2.4 HDL Coder

MATLAB design, Simulink models and State flow charts can produce a hardware description level language such as Verilog and VHDL code with the use of HDL coder. The auto generated HDL code can be used to program, prototype and design FPGA or ASIC. Workflow advisor is available in HDL coder to automate Xilinx, Microsemi and Intel FPGAs programming. It can control and implement HDL architecture, highlight critical paths, and generate estimates of the use of hardware resources. HDL Coder provides standardization between Simulink model and Verilog and VHDL code generated, enabling high-integrity code verification.

Design, modeling, simulation, code generation and implementation are provided by MATLAB and Simulink for model-based design. On MATLAB, Simulink, and State flow models can be designed and simulated and then HDL Coder generates Verilog and VHDL code for FPGAs and ASICs. Alternatively, Xilinx library of bit- and cycle-true blocks can be used to build a model in Simulink targeted for Xilinx FPGAs. Xilinx System Generator can then be used to automatically generate synthesizable hardware description language (HDL) code mapped to pre-optimized Xilinx algorithms [13].

To synthesize the HDL code and generate a bit stream for the FPGA, a three-step process is followed. First step refers to the model designing with MATLAB functions and Simulink blocks. HDL codes are generated at the end of this step, which can then be synthesized using third-party tools and generate the bit stream. The 3rd step uploads a bit stream to the FPGA. With the Synthesis tool option, HDL Coder can integrate third-party tools into Workflow Advisor to create a uniform and integrated environment (Simulink Environment) for all processes, from model design to bit stream generation.

Chapter 3

Design

3.1 Naive Bayes's Classifier

Naive Bayes' Classifier was introduced almost 50 years ago and mostly suited when the amount of data set is large. We have already discussed about Naive Bayes' Classifier (NBC) in chapter 2, in this chapter we shall design the algorithm on Matlab and Simulink environment for different examples and applications, especially for medical applications NBC proved to be quit efficient for classification. The mathematical model of NBC can written as,

- Let there is 'r' dimensional vector with $Y : (y_1, y_2, ...y_n)$
- There are 'q' classes : $C_1, C_2, ...C_q$.

NBC forecasts Y attribute be a member of a class C_i if.

$$P(C_i/Y) > P(C_j/Y) for l \leq j \leq q, j <> i Maximum Posteriori Hypothesis \quad (3.1.1)$$

$$P(C_i/Y) = \frac{P(Y/C_i) \times P(C_i)}{P(Y)} Maximize P(Y/C_i) \times P(C_i) as P(Y) is constant \quad (3.1.2)$$

With high number of attributes, it is computationally costly to assess $P(Y/C_i)$

$$P(Y/C_i) = \prod_{k=1}^{r} \times P(y_k/C_i) \quad (3.1.3)$$

$$P(Y/C_i) = P(y_1/C_i) \times P(y_2/C_i \times .... \times P(y_n/C_i) \quad (3.1.4)$$

Example 3.1.1. The given data table describes that there are students and non students with a status of age, income and credit ratings with a given condition that who can buy a computer based on their status as follows

rec	Age	Income	Student	Credit-rating	Buys-Computer
r1	≤ 30	High	No	Fair	No
r2	≤ 30	High	No	Excellent	No
r3	31...40	High	No	Fair	Yes
r4	> 40	Medium	No	Fair	Yes
r5	> 40	Low	Yes	Fair	Yes
r6	> 40	Low	Yes	Excellent	No
r7	31...40	Low	Yes	Excellent	Yes
r8	≤ 30	Medium	No	Fair	No
r9	≤ 30	Low	Yes	Fair	Yes
r10	> 40	Medium	Yes	Fair	Yes
r11	≤ 30	Medium	Yes	Excellent	Yes
r12	31...40	Medium	No	Excellent	Yes
r13	31...40	High	Yes	Fair	Yes
r14	> 40	Medium	No	Excellent	No

Figure 3.1: Will a person buy a computer? if a person has following attributes

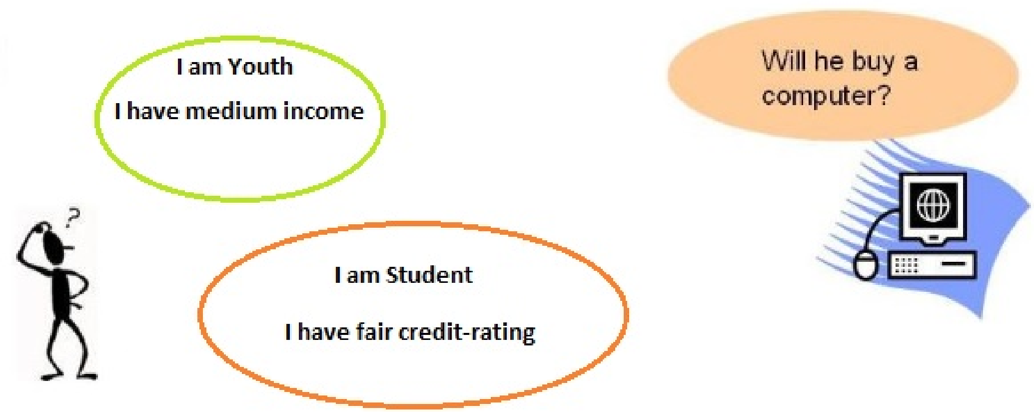

Figure 3.2: Attributes of a Person

Now let us represent above data table into meaningful numbers as follows

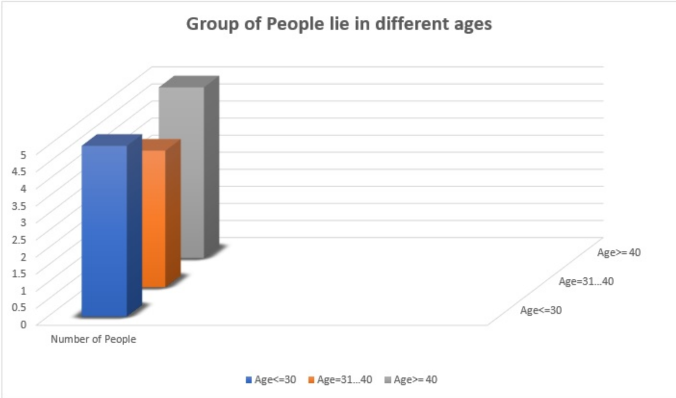

Figure 3.3: This bar chart represents a group of people of three age groups, five of them aged less than 30 years, four of them aged between 31 to 40 years and five of them aged above 40 years.

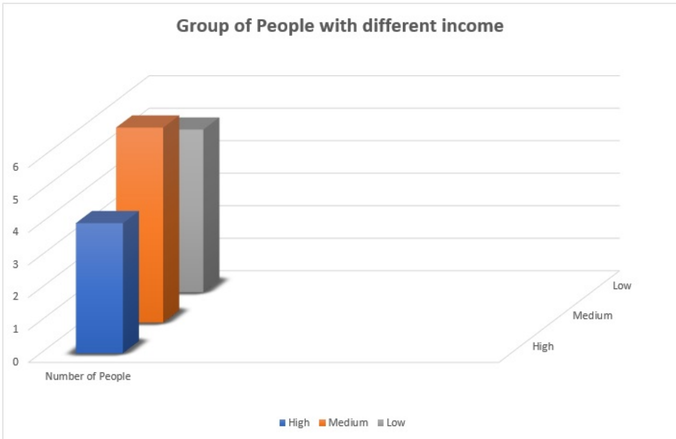

Figure 3.4: This bar chart represents number of people with different income status

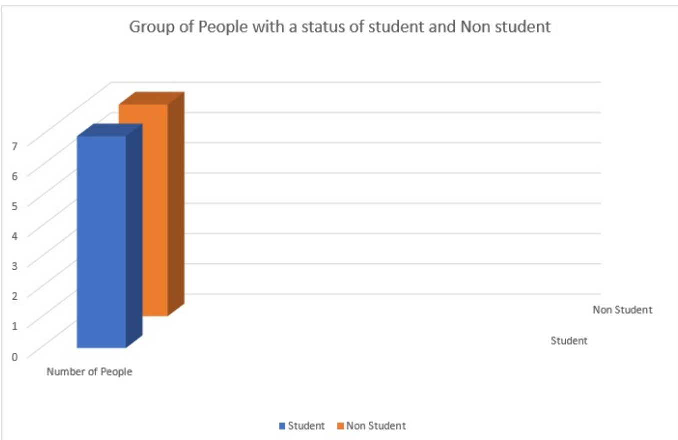

Figure 3.5: Bar chart represents the number of students and non-students

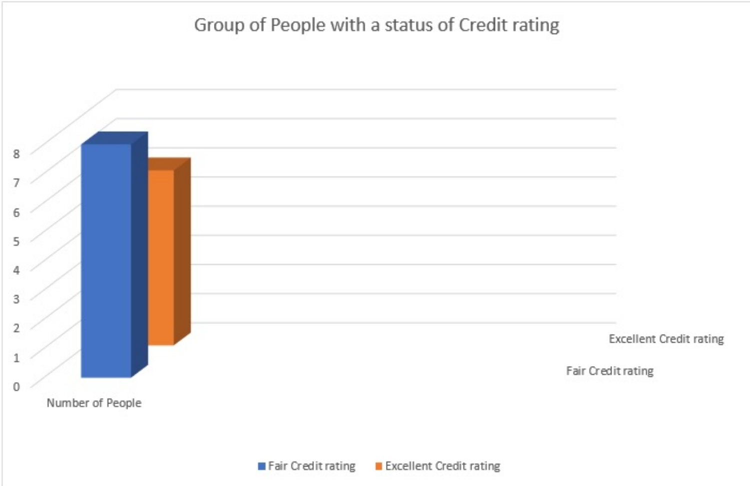

Figure 3.6: Number of people having fair and excellent credit rating.

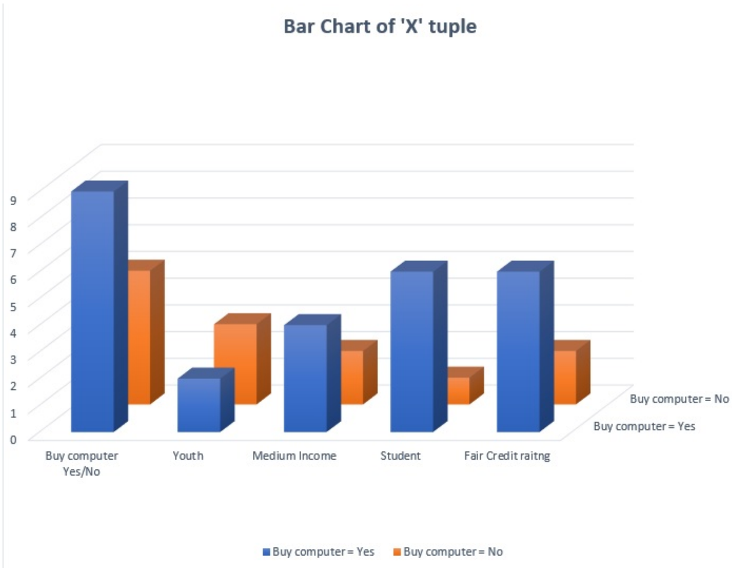

Figure 3.7: Frequency bar chart of X tuple

Solution:

Let us calculate the respective probabilities,

$$P(buys - computer = yes) \rightarrow P(C_1) = \frac{9}{14} = 0.643$$

$$P(buys - computer = no) \rightarrow P(C_1) = \frac{5}{14} = 0.357$$

$$P(youth \setminus buys_computer = yes) = \frac{2}{9} = 0.222$$

$$P(youth \setminus buys_computer = no) = \frac{3}{5} = 0.6$$

$$P(mediumincome \setminus buys_computer = yes) = \frac{4}{9} = 0.444$$

$$P(mediumincome \setminus buys_computer = no) = \frac{2}{5} = 0.4$$

$$P(student \setminus buys_computer = yes) = \frac{6}{9} = 0.667$$

$$P(student \setminus buys_computer = no) = \frac{1}{5} = 0.2$$

From the above probabilities, calculate the probability of 'X' tuple given that he buys a computer as,

$$P(X \backslash C_1) = P(youth \backslash C_1) \times P(mediumincome \backslash C_1) \times P(student \backslash medium) \times P(faircreditrating \backslash C_1) =$$

From the above probabilities, also calculate the probability of 'X' tuple given that he does not buy a computer

$$P(X \setminus Buysacomputer = No) = 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019$$

Now find class C_i that maximize $P(X \setminus C_i) \times P(C_i)$

$$\Rightarrow P(X \setminus C_1) \times P(C_1) = 0.028$$

$$\Rightarrow P(X \setminus C_2) \times P(C_2) = 0.007$$

Prediction:

As $P(X\setminus C_1) > P(X\setminus C_2)$ so he will buy a computer, further more lets check normalizing the probability that he will buy a computer.

$$P = \frac{0.028}{0.035} = 0.8$$

It means there are 80% chances that he will buy a computer and 20% chances that he will not buy a computer.

Classification of Iris Data Using Gaussian NBC

Iris data set is a multivariate data set, provided by Britisher biologist to analyze changes in morphology of flowers of three related class of species. Data set comprises of 50 samples from each of 3 species that are Iris setosa, Iris virginica and Iris versicolor and 4 features were extracted from each of sample that is sepals length, sepals width, petals length and petals width in centimeters respectively.

Based on these feature, he introduced a model called linear discriminator to differentiate between species. Few features of data set are shown here but complete data set is given in appendix 1.

Sepal Length	Sepal Width	Petal Length	Petal width	Species
5.1	3.5	1.4	0.2	'setosa'
4.9	3	1.4	0.2	'setosa'
4.7	3.2	1.3	0.2	'setosa'
4.6	3.1	1.5	0.2	'setosa'
5	3.6	1.4	0.2	'setosa'
7	3.2	4.7	1.4	'versicolor'
6.4	3.2	4.5	1.5	'versicolor'
6.9	3.1	4.9	1.5	'versicolor'
5.5	2.3	4	1.3	'versicolor'
6.5	2.8	4.6	1.5	'versicolor'
6.3	3.3	6	2.5	'virginica'
5.8	2.7	5.1	1.9	'virginica'
7.1	3	5.9	2.1	'virginica'
6.3	2.9	5.6	1.8	'virginica'
6.5	3	5.8	2.2	'virginica'

Figure 3.8: Iris Data set

Algorithm Steps

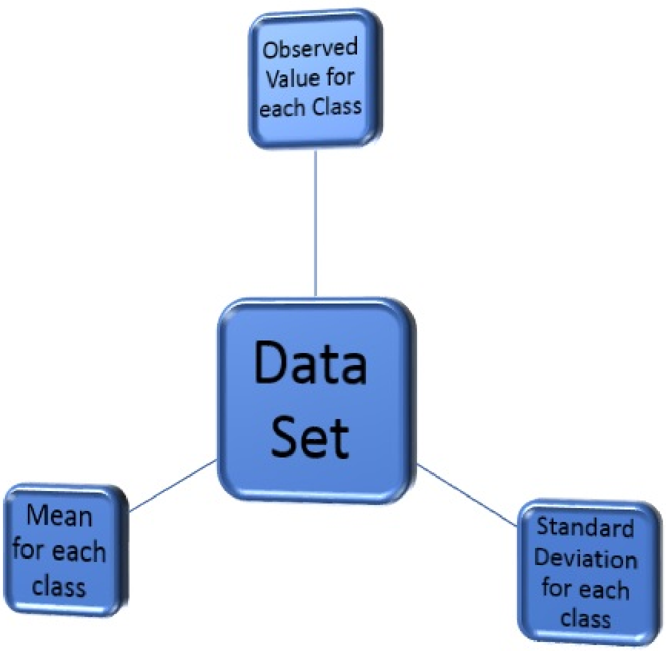

Figure 3.9: Calculation of Parameter from data set

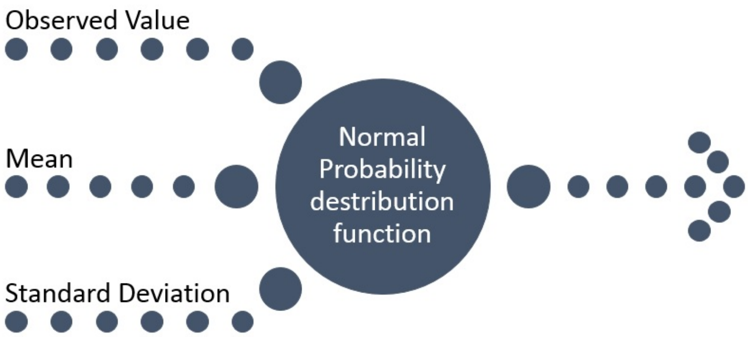

Figure 3.10: Determining Normal Probability distribution function

1 Initially the data set of Fisher's Iris is loaded in Matlab with a command of load fisheriris.

2 Find the probability of each class that is $C_1, C_2 and C_3$ or Setosa, versicolor and versicolor respectively with each of feature that are sepal length & width and petal length & width. Probability of each class is $P(C_1) = P(C_2) = P(C_3) = 0.333$.

3 Find the respective standard deviation and mean of Iris data with each feature.

4 Choose the observed value from data feature for each class.

5 Find the probabilities of respective classes and respective feature using Normal distribution function.

6 Find the probabilities that maximise the class.

7 Finally compare the probabilities, the greater probability has more chances of occurrence.

Parameters	Sepal Length	Class
Mean	5.006	Setosa
Standard Deviation	0.3525	Setosa
Normal PDF	0.0895	Setosa
Observed Value	5.8	Setosa
Mean	5.936	Versicolor
Standard Deviation	0.5162	Versicolor
Normal PDF	0.0924	Versicolor
Observed Value	7	Versicolor
Mean	6.588	Virginica
Standard Deviation	0.6359	Virginica
Normal PDF	0.0747	Virginica
Observed Value	7.9	Virginica

Table 3.1: Calculations of Parameters of Sepal length with respect to Class

Parameters	Sepal width	Class
Mean	3.428	Setosa
Standard Deviation	0.3791	Setosa
Normal PDF	0.0393	Setosa
Observed Value	4.4	Setosa
Mean	2.77	Versicolor
Standard Deviation	0.3138	Versicolor
Normal PDF	0.1694	Versicolor
Observed Value	3.4	Versicolor
Mean	2.974	Virginica
Standard Deviation	0.3225	Virginica
Normal PDF	0.0645	Virginica
Observed Value	3.8	Virginica

Table 3.2: Calculations of Parameters of Sepal width with respect to Class

Parameters	Petal Length	Class
Mean	1.462	Setosa
Standard Deviation	0.1737	Setosa
Normal PDF	0.0955	Setosa
Observed Value	1.9	Setosa
Mean	4.26	Versicolor
Standard Deviation	0.4699	Versicolor
Normal PDF	0.1718	Versicolor
Observed Value	5.1	Versicolor
Mean	5.552	Virginica
Standard Deviation	0.5519	Virginica
Normal PDF	0.0366	Virginica
Observed Value	6.9	Virginica

Table 3.3: Calculations of Parameters of Petal length with respect to Class

Parameters	Petal Width	Class
Mean	0.246	Setosa
Standard Deviation	0.1054	Setosa
Normal PDF	0.0134	Setosa
Observed Value	0.6	Setosa
Mean	1.326	Versicolor
Standard Deviation	0.1978	Versicolor
Normal PDF	0.1141	Versicolor
Observed Value	1.8	Versicolor
Mean	2.026	Virginica
Standard Deviation	0.2747	Virginica
Normal PDF	0.3276	Virginica
Observed Value	2.5	Virginica

Table 3.4: Calculations of Parameters of Sepal Width with respect to Class

Find the probabilities that maximise the class

1. $P_1 = P(C_1) \times P(sl_1) \times P(sw_1) \times P(pl_1) \times P(pw_1) = 1.50357e - 06$

2. $P_2 = P(C_2) \times P(sl_2) \times P(sw_2) \times P(pl_2) \times P(pw_2) = 1.0222e - 04$

3. $P_3 = P(C_3) \times P(sl_3) \times P(sw_3) \times P(pl_3) \times P(pw_3) = 1.3894e - 05$

4. $P = P_1 + P_2 + P_3 = 1.1762e - 04$

Normalised Probabilities are

$$P_1 = 0.0128 \tag{3.1.5}$$

$$P_2 = 0.8691 \tag{3.1.6}$$

$$P_3 = 0.1181 \tag{3.1.7}$$

Conclusion

From the calculated probabilities of respective classes that are P_1 refers to Setosa class, P_2 refers to Versicolor class and P_3 refers to Virginica. It is observed that the chances of occurrence of Versicolor are higher that is 86.9% , while the chances of occurrence of Virginica are lesser that is 11.81% and the chances of occurrence of Setosa are the least that is 1.28%, therefore Versicolor is the type of species according to given data set.

3.2 Detection of Epileptic events using Gaussain Naive Bayes' Classifier using Matlab Simulink

Epilepsy is one of the serious neurological disorder which may be found at any stage of life to a male or female, adults or children but the occurrence in children is higher than adults [14]. According to WHO report around, 50 million people across the world have been capitulated with epilepsy which makes people mentally retarded and uncommunicable due to which their family suffers from discrimination and it was also studied in WHO report that about three quarters of people living in low and middle class countries with epilepsy like Pakistan and are not diagnosed properly [15]. Epilepsy can be forecasted earlier with the help of EEG (Electroencephalogram) signals to identify the stage of seizure or non-seizure or whether it is focal (partial) EEG or non-focal EEG so that Physician can go through the proper treatment [16]. EEG is the electrical measure of brain activity which provides the statistics of brain psychological state. In this modern medical technological era, diagnostic and testing tools are present however, the area requires automated solutions capable to perform computations and producing results in milliseconds. In this thesis we have discussed the implementation of the Naive Bayes' classifier for the classification of epileptic events into seizure or non-seizure class based on prior knowledge of events.

The data sets used in the present study are provided by M.Liaquat Raza. From the Inst. of Neurophysiology Charite, Universitatmedizin, Berlin, Germany. This study focused on sK-channel agonists for testing them in acute rat slices. When 4-aminopyridine (4-AP) induced this generated seizure-like events (SLEs).

Young adult rats of 400 uM thickness were used to prepare slices and transferred into interphase chamber. Moreover, epileptic activities in slices were induced by 4-AP, 100 uM. Local field potentials were logged via microelectrode from medial entorhinal cortex. Reference papers for the datasets in the present study are [[17], [18], [19]]. The above testing and training data are 100 times scaled of original one. Below table represents 10 samples of data feature of seizure and non-seizure for training and testing, but a complete data set is given in appendix 2.

Table 3.5: Training Data

Non-Seizure Data

C V A	Entropy	Mean	R M S
5.7	249.7	25.3	1.4
13.0	100.2	5.8	0.8
14.0	106.7	5.6	0.8
10.9	111.7	7.4	0.8
13.7	104.2	5.6	0.8
6.3	153.0	15.9	1.0
13.7	118.4	6.0	0.8
11.5	99.6	6.5	0.8
8.9	163.8	11.4	1.01
11.03	116.2	0.8	0.83

Seizure Data

C V A	Entropy	Mean	R M S
1.4	127.49	28.6	3.36
1.1	493.88	77.7	8.17
0.4	132.94	63.4	4.16
1.6	142.91	29.1	3.66
1.1	588.88	93.0	9.58
1.0	428.93	83.0	8.12
100	282.04	57.6	8.87
1.7	872.13	87.5	11.44
1.13	4366.9	72.6	7.71
1.1	2595.8	51.0	5.32

Table 3.7: Testing Data

Non-Seizure Data

C V A	Entropy	Mean	R M S
14.9	117	5.4	0.81
10.7	112.3	7.5	0.8
13.8	135.5	6.5	0.9
10.7	116	7.7	0.83
13.8	143.4	6.7	0.92
11.5	137.8	7.9	0.91
9.6	158.9	10.4	1.0
11.6	117.1	7.1	0.82
10.4	112.5	7.8	0.81
6.5	242.8	19.9	1.3

Seizure Data

C V A	Entropy	Mean	R M S
0.8	19.6	3.6	0.32
0.2	12.8	5.2	0.3
1.1	1517.2	36.7	3.8
1.7	1382.6	26.3	3.5
0.7	1397.5	45.5	3.8
0.4	1123.3	56.5	3.7
1.1	7.0	1.6	0.2
1.8	6.3	1.2	0.2
0.7	879	35.8	3.0
1.0	1330.8	34.5	3.6

3.2.1 Matlab Simulink Design Model

The NBC algorithm is designed in Matlab Simulink environment, built in functions like mean, standard deviation, normal probability distribution function are not used but these functions are designed manually. Basically NBC algorithm is designed for any kind of application for a pre-defined data feature length, in this thesis the standard pre-defined data feature length is 50 data features for each class.The Simulink design comprises of three parts as given,

1 The switch circuit.

2 Normal Probability distribution function (pdf).

3 Demux Circuit.

The switching circuit

For the above application, the prediction of Epileptic events in to seizure or non-seizure class, has two classes. There are four variables, the length of each variable is 50. It means at a time a single data feature variable is to be selected so that parameters say mean and standard deviation can be evaluated for NBC, likewise one by one these feature are selected. The below figure represents the switching circuit.

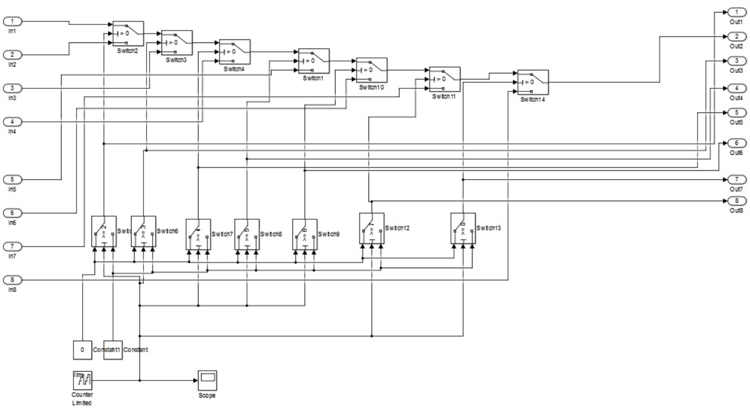

Figure 3.11: Switching Circuit

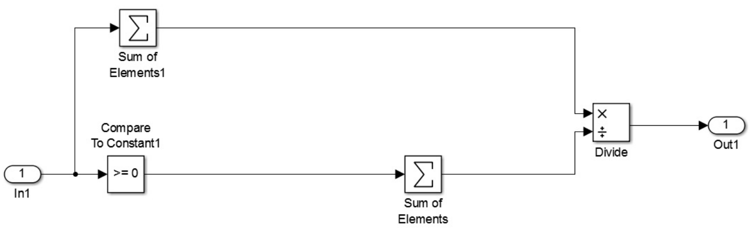

Figure 3.12: Mean Circuit

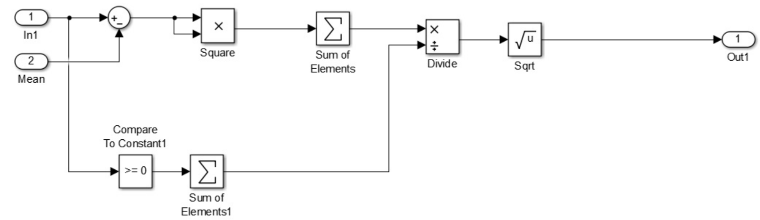

Figure 3.13: Standard Deviation Circuit

Normal Probability Distribution Function

After calculation of these parameters that is mean, standard deviation and any observed value from data feature. These values are given to normal probability distribution function to evaluate it.

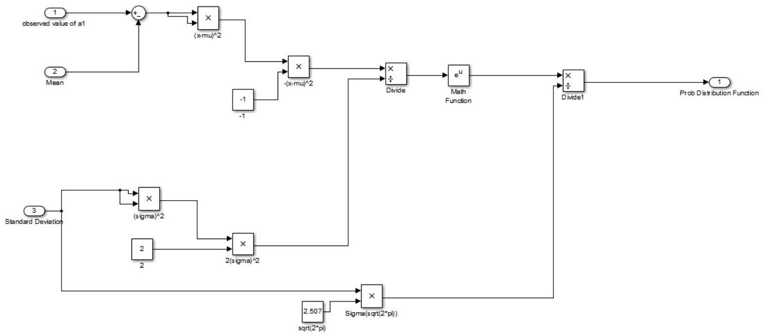

Figure 3.14: Normal PDF Circuit

Demux Demux circuit keep every calculated normal pdf in to memory places so that each probability can be used for further calculations.

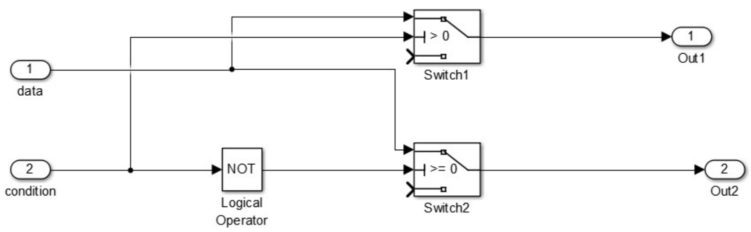

Figure 3.15: Basic Demux Circuit

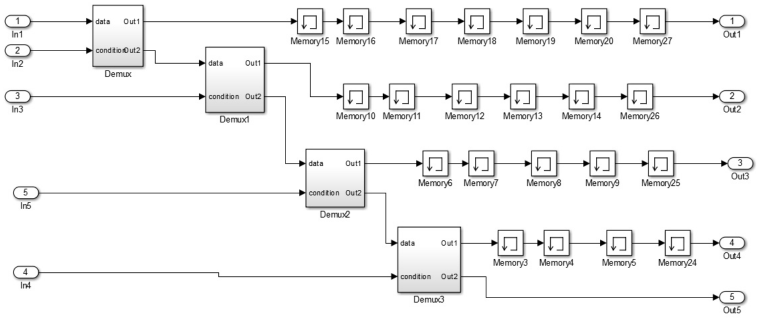

Figure 3.16: Complete Demux Circuit

Results

Matlab is versatile design tool box in which we have designed the algorithm in Simulink environment and getting almost 100 percent of accuracy, it can be shown in below confusion matrix figures that the Naive Bayes' classifier classifies all data features into its correct class, for instance a false data features are also given to classifier to check its viability.

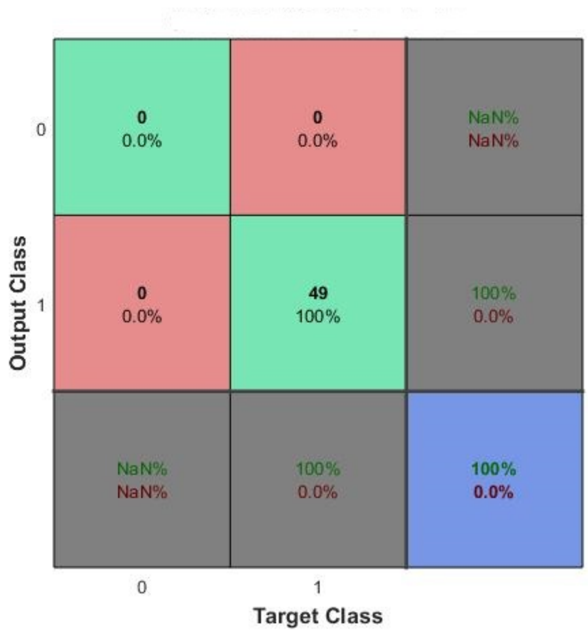

Figure 3.17: Confusion Matrix of 100% correct.

Figures [3.15, 3.16, 3.17, 3.18] of confusion matrix (CM), the green marked portion shows the correct classification while the red marked portion shows the false classification with a defined rate of percentage for both classes.Figures [3.15, 3.16, 3.17, 3.18] represents for an input data of 49 samples.

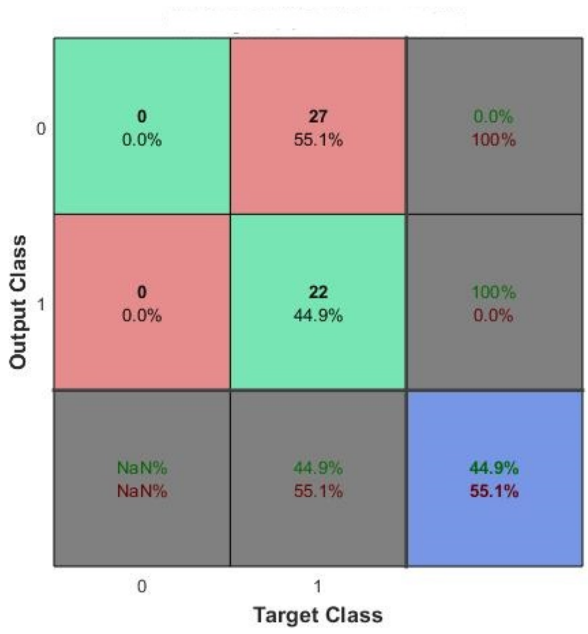

Figure 3.18: Confusion Matrix of 50% correct.

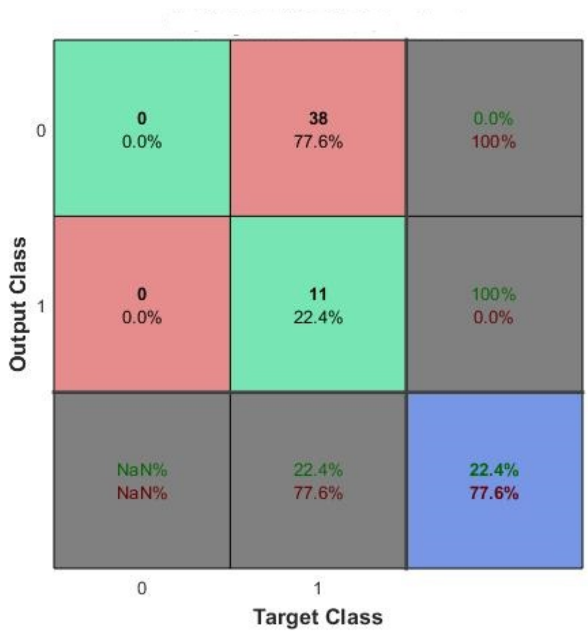

Figure 3.19: Confusion Matrix of 25% correct.

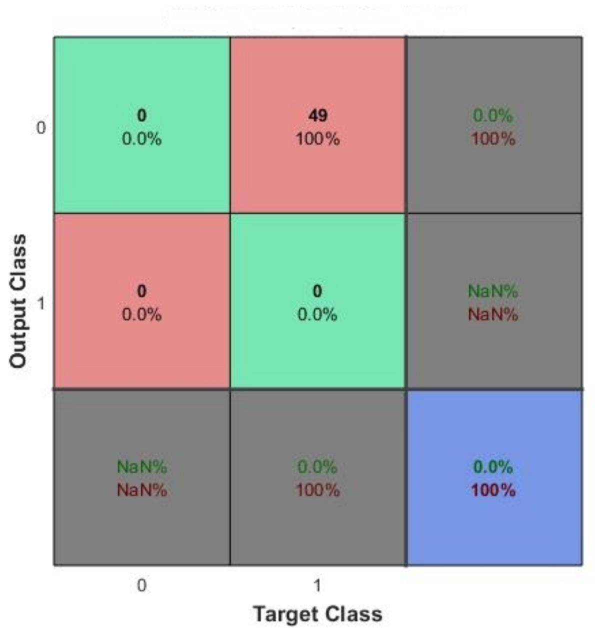

Figure 3.20: Confusion Matrix of 0% correct.

The prognosis of seizure mainly depends on its features. The data features that will be extracted from a pre-processed EEG signals are as follows

1. Continuous Varying of Amplitude (CVA)
2. Entropy
3. Mean
4. Root Mean Square

Each of the feature reveals unique information to collectively allows the detection of seizure. For example, the entropy proves useful in revealing important information for classification of EEG data [[20], [16]]. Now let us observe and analyze these parameters that affect the accuracy of classifier by varying one parameter at a time and keeping other parameters constant. In other words by varying the values of one parameter and keeping others same, we can find out the acceptable range of error in the particular parameter. For example, by varying the values of co-efficient of varying amplitude (CVA) and keeping others constant, we can find out its sensitivity and check its overall effect on the accuracy of classification. As we can see in Fig.3.19 CVA has the nonlinear behaviour which moves towards non-seizure event from value of 0.01 to 0.02 and remains steady after the value of 0.02, therefore, we can say that the classifier is not much sensitive to this parameter and a big error margin in this parameter can be accepted.

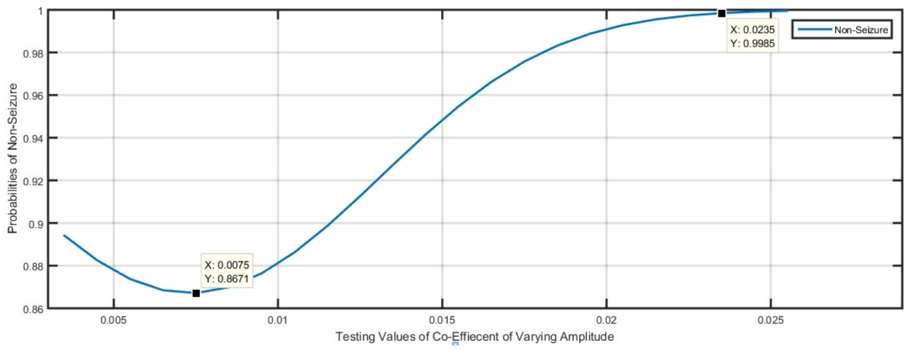

Figure 3.21: Co-efficient of Varying amplitude.

In Figure 3.20 the values of entropy are varied for a particular range to check it's sensitivity while keeping others constant and it is observed that entropy is much sensitive to move towards seizure or non-seizure events in very sharp range and then becomes constant like for non-seizure event when the value of entropy is changed from 7 to 8 the probability of non-seizure tends to zero and the seizure event tends to unity, means there will be more chances of seizure if the value of entropy approaches to 8.

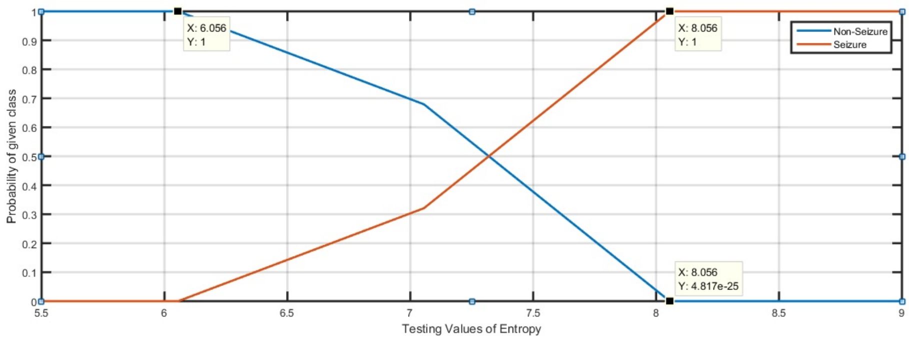

Figure 3.22: Entropy

Similarly in Fig. 3.21 the root mean square has the most sharp range of 0.037 to 0.0385 & is more sensitive as compared to entropy while Fig. 3.22 the mean has the sharper range of 0.87 to 0.88 & is sensitive in between entropy and RMS. Figures 5,6,7 and 8, a blue annotation refers to non-seizure event while a red one refers to seizure event as mentioned in figures as a label.

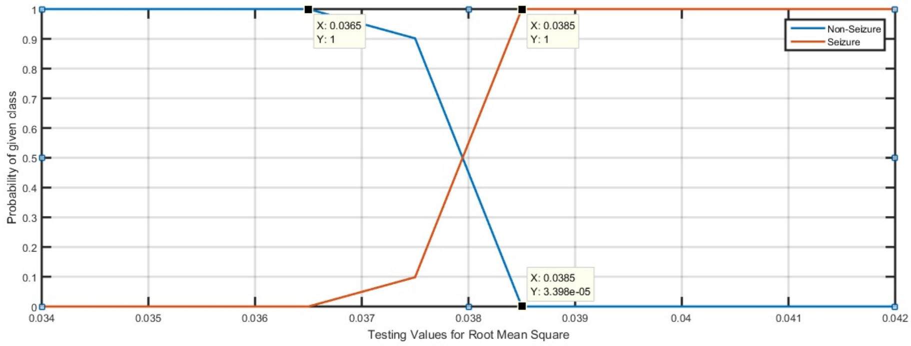

Figure 3.23: Root Mean Square

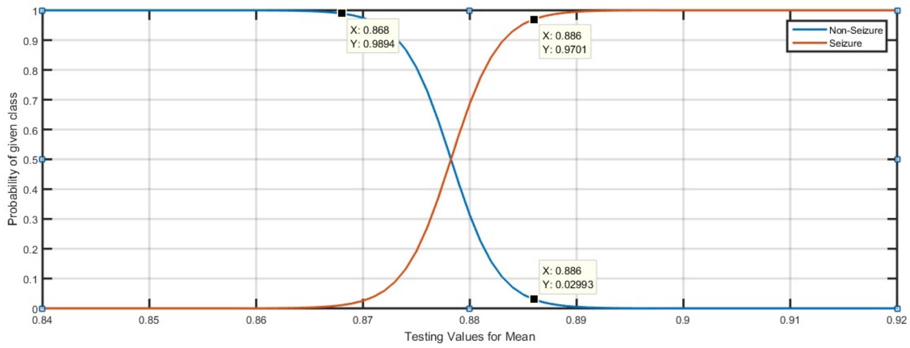

Figure 3.24: Mean

Chapter 4

Implementation

4.1 Introduction

In this thesis we have introduced the implementation of Naive Bayes' algorithm (a statistical algorithm) on embedded systems using HDL coder. The algorithm determines the various probability distribution function using Bayes' Theorem [21]. NBC is implemented on FPGA with the help of HDL coder and fixed-point tool in Simulink environment. HDL coder is the Matlab tool which converts the Simulink designs into HDL(hardware description language) that is verilog or VHDL. The main advantage of using HDL coder is that the designer can more concentrate on tuning the algorithms and designs rather than coding on HDL. Previous work [[22], [23], [22], [24]] focuses on different algorithms and analysis but there is still a space in hardware implementation of algorithms on FPGA using HDL coder. Our main focus was on the implementation and analysis of seizure detection by varying the data feature's parameters as discussed in chapter 3.

4.1.1 Algorithm Flow Chart

Fig. 4.1 gives a sequential diagrammatic summary of the methodology followed.Initially an EEG data of patient is pre-processed to achieve it's data features then data features are converted in a pre-defined significant digits. In order to make design compatible for HDL coder, a design is converted into fixed point with pre-defined bits with the help fixed point tool, in the next step algorithm's parameters are estimated that are mean and standard deviation, are given to algorithm. The last step uses HDL coder to convert design into Verilog code.

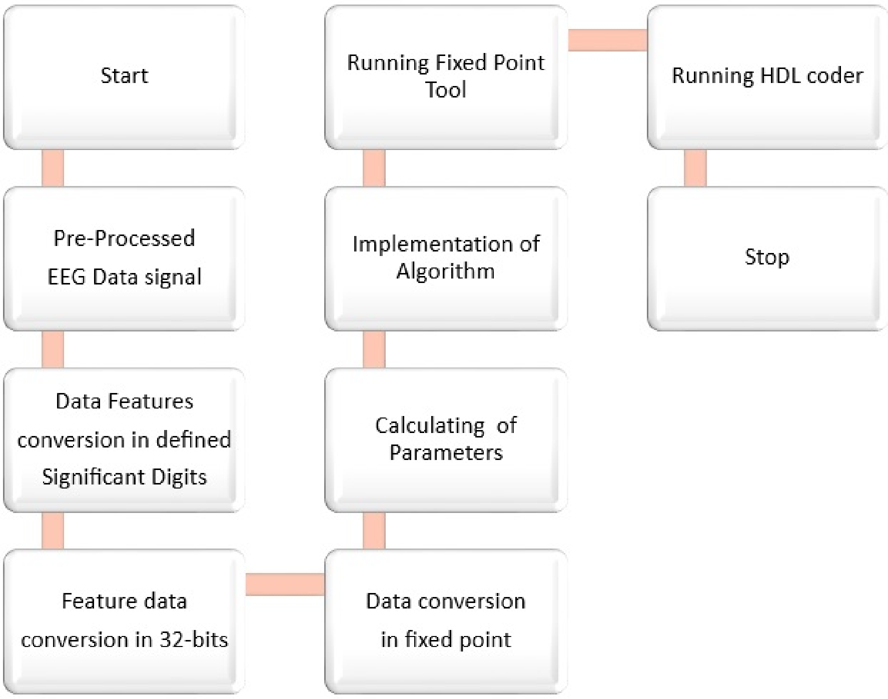

Figure 4.1: Algorithm Hierarchy

4.2 FIXED POINT MODEL GENERATION

High level operators and functions replaced with accurate hardware models A floating point MATLAB algorithm cannot accurately determine the final hardware response if the high level operators and functions in it are not replaced with accurate hardware macroarchitectures as shown in Fig.4.2

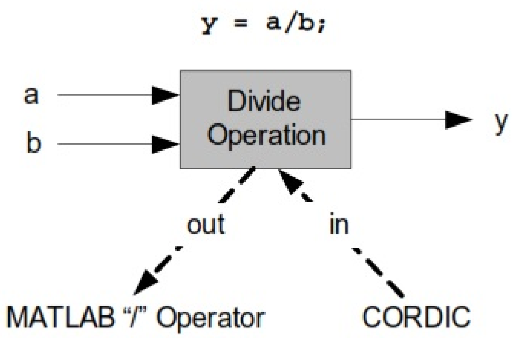

Figure 4.2: Replacing Built-in Operators and Functions

This can be seen in Fig. 4.2 that compares the responses of the MATLAB fixed point divide operator against a CORDIC algorithm (a hardware implementation) for division for random inputs of 8-bits signed vectors. Depending upon the input data, there may be a considerable difference in the calculated outputs. CORDIC is a co-ordinate rotational digital computer which uses algorithm to evaluate different functions like trigonometric functions, division and etc and converging with one digit per iteration thus also known as digit-by-digit algorithm.

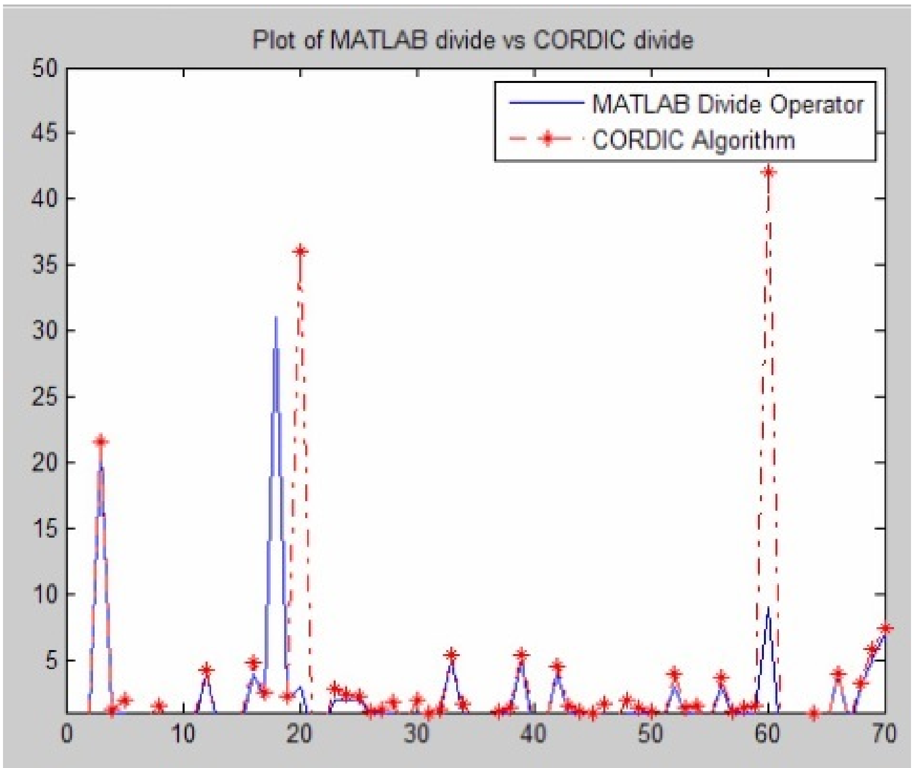

Figure 4.3: Fixed point response of Matlab "/" Vs Cordic

During the process of fixed point generation, the high level MATLAB operators and functions are automatically replaced by accurate hardware representation as shown in Fig. 3. The step however, does not require any modifications of the MATLAB code since it is transparent to the user. A synthesis directive may be used to re-define the initial micro and macro architecture.

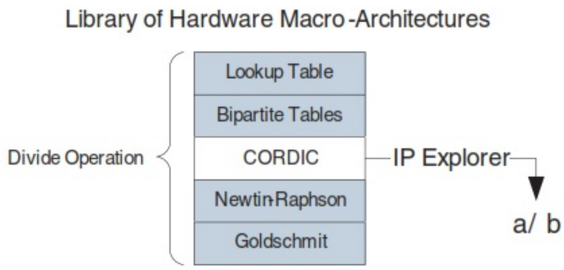

Figure 4.4: Automatic Hardware Accurate IP Insertion

Quantization can then begin after the operators and functions have been replaced with hardware architectures.

Auto-Quantization through Graphical Assistance

A Field Programmable Gate Array (FPGA) can allow for a variable word length of fixed points words unlike the DSP processor. By not restricting the variable lengths to fixed 16 or 24 bits, calculations requiring additional bits may be done without introducing the numerical errors. Clearly, its a tremendous for applications like guidance systems, navigation and radar, however, at the cost of the increased hardware.

For most of the cases, bit growth rules are pretty simple. For example in Fig.4 an addition operation casues results in a one bit increase in the answer and that in a multiplication operation is increased to the sum of input words lengths. However, for variables in an actual design, these determinations can be made in an iterative process. One cannot allow an unchecked or unnecessary bit growth on a hardware level since this is going to be hardware expensive. Various techniques may be employed to minimize the word lengths while maintaining the same level of numerical accuracy.

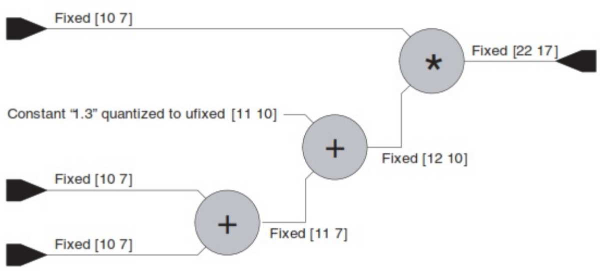

Figure 4.5: Fixed-Point Bit Growth

Automation tools may be employed for determining the quantization values for the variables and there afterwards refinements as well. The Accel DSP Synthesis Tool can act as an automation tool in this regard. This tool allows a floating-point MATLAB model during the simulations and determines the dynamic range of input data. These ranges serve as initial values to start an auto-quantization process to determine the optimal word lengths with the help of a database containing over 6000 designs.

Summary: MATLAB is a natural choice for a DSP algorithm and that is to be unencumbered by hardware constraints. However, transforming the algorithm into a fixed point model is an involved process for a FPGA implementation. This process may be supported by the automation, acceleration and visualization tools offered by the Accel DSP Synthesis Tool.

4.3 FPGA Implementation

In this research we had supplied a 32-bit length feature data (seizure or non-seizure) to the q-kintex 7 FPGA device (Target Device: xq7k325t) that is using the Naive Bayes classiïňĄer algorithm for processing in real time. qkintex7 is a high end FPGA mostly suitable for high end applications like 3G and 4G wireless. Following are some key features of q-kintex 7 as,

DSP Slices The device contains 840 DSP slices each containing four number of Look Up Tables (LUT) and eight number of flip flops. Also, each DSP slice contains a pre-adder, a 25 x 18 multiplier, an adder and an accumulator.

Block RAM There may be 445 number of Block RAMS each with a size of 36Kb or 890 number of lock RAMs each with a size of 18Kb both configurations allowing a total of 16020 Kb memory size.

Clock Management Tiles There are 10 number of Clock Management Tiles (CMT) each containing 1 number of Mixed Mode CLock Manager (MMCM) and 1 number of Phase Lock Loop (PLL).

PCIe interface There is only 1 number of interface for PCI Express support upto x8 Gen 2.

4.3.1 FPGA Results

The data we had processed using the above said algorithm contains 49 samples each of which containing 4 variables. These variables are 'co-efficient of varying amplitude' (CVA), 'entropy', 'mean' and 'root mean square' (rms) deïňĄned in chapter 3. This real time processing results in an achieved accuracy of 97% correct classification of the given data as compared to their actual classes. The remaining 3% results are that are incorrectly classified are so since the input data to the processor has been rounded up to 4 decimal places.The algorithm has the following computational and timing specifications,

- clock period is 558.467ns ($frequency : 1.791MHz$)
- Maximum combinational path delay is: $827.013ns$
- Data samples processes per second $\geq$ 360885

Also, the target device that is xq7k325t has the following utilization and distribution details.

- Number of Slice Registers: 1268
- Number of Slice LUTs: 95940 (95748 as Logic,192 as SRL memory)
- LUT Flip Flop pairs: 96375
- Total memory usage: $2869128KB$

Besides the above, the number of DSP48E1 slices used in the target devices numbers to 804. Also, the device has 12932 used number of IOs.

Chapter 5

Discussion, Future Work & Conclusion

5.1 Discussion

In this thesis we have discussed the implementation of algorithm NBC on FPGA using HDL coder. A design is bridged to verilog with minimal development time with use of HDL coder. The main purpose of using HDL coder is to enhance system reliability and save development time by 33% as compared to coding on HDL [25]. HDL also reduces the cost from 1/5 to 1/2 as compared to conventional method [26]. The Matlab Simulink design also verifies the HDL code by co-simulation with Simulink and creates the FPGA based prototypes [27]. In this application the data samples are of float type so need to be converted in fixed bit length, for that fixed point tool is to be used for converting the data of any type into defined fixed length.Design's parameters are being tuned according to availability of hardware resources and desired outcomes, at each step the design's parameters and signals need to be converted in fixed point data type and one of the best part of using fixed-point tool is that one can specify the maximum and minimum output range to get the best precision scaling. A design uses operators like division, multiplication, addiction,and exponential function so for division a reciprocal block is used with multiplication block to perform division and for the exponential function, a 1-D look-up table is used. Though the limited bit size is used in algorithm we have proved the robustness of algorithm but also provided the best results under trade off in rounding the data set to three significant digits. We have also observed that the algorithm works well even when the data size is higher in Simulink but due to limited hardware resources we have limited the size the of data set approximately 400 data samples are computed each variable is of around 50 data samples.

5.2 Future Work

As discussed in chapter 4, the resources utilized on q-kintex 7 FPGA hardware which consumes some how greater resources than a conventional method of coding on FPGA, so there is need to further optimize the algorithm and parameters, in order to achieve this we need to use S-function for every single function in Simulink Matlab. In our design there are some net delays and these can be reduced to optimal value if the frequency is further reduced than current one. Moreover every custom function used in our design should further be optimized like division, we have used reciprocal function and multiplier to perform division, but this uses two operators instead of one so there is need to design division operator which can support any fractional length while converting from floating to fixed-point. This can be accomplished using Cordic algorithms which provides the best performance.

5.3 Conclusion

The main focus of this research was on the implementation of complex algorithm on FPGA using HDL coder. A design is generalized for any kind of application with defined data feature length for classification, the system reliability is enhanced and development time is saved by 33% as compared to coding on VHDL or Verilog. We have observed and analyzed from the above results that dependency and sensitivity of seizure detection by varying parameters and found that the most sensitive parameter that is root mean square and the less sensitive parameter is co-efficient of varying amplitude.

References

[1] Declan O'Sullivan Haytham Assem. Towards bridging the gap between machine learning researchers and practitioners. *ADAPT Center, School of Computer Science and Statistics,Trinity College Dublin, Ireland*, 151.

[2] Ayon Dey. Machine learning algorithms: A review. *IJCSIT: International Journal of Computer Science and Information Technologies*, 7.

[3] Andrew W. Moore Leslie Pack Kaelbling, Michael L. Littman. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4.

[4] R. Caruana. Multitask learning. *Machine Learning*, 28.

[5] Tom Heskes Bart Bakker. Task clustering and gating for bayesian multitask learning. *Machine Learning Research*, 4.

[6] I. Rish. An empirical study of the naÃŕve bayes classifier. *Watson Research Cente.*

[7] Michael Pazzani Pedro Domingos. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29.

[8] Andrew Y.Ng and Micheal I.Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naÃŕve bayes. *Neural Information Processing Systems*, 14.

[9] Sean R Eddy. What is bayesian statistics? *A Nature Reasearch*, 22.

[10] Marvin Minsky. Steps toward artiïňĄcial intelligence. *Proceedings of the Institute of Radio Engineers*, 49.

[11] Konstantinos V. Chandrinos George Paliouras Ion Androutsopoulos, John Koutsias and Constantine D. Spyropoulos. An evaluation of naive bayesian anti-spam filtering. *11th European Conference on Machine Learning, Barcelona, Spain.*

[12] Pat Langley George H.John. Estimating continuous distributions in bayesian classiïňĄers. *Proc. Eleventh Conf. on Uncertainty in ArtiïňĄcial Intelligence.*

[13] Yongfeng Gu Kiran Kintali. Model-based design with simulink, hdl coder, and xilinx system generator for dsp.

[14] Rados C. Epilepsy and seizures can occur at any age. *NCBI*, 39.

[15] WHO. World health organization. fact sheet, epilepsy. *Fact Sheet.*

[16] V. Venkatraman N Arunkumar, K Ramkumar. Classification of focal and non-focal eeg using entropies. *Pattern Recognition Letters.*

[17] Syed Sajjad Haider Zaidi Bilal Muhammad Khan Uwe Heinemann Muhammad Khizar Abbas, Muhammad Liaquat Raza. Seizure forecasting using single robust linear feature as correlation vector of seizure-like events in brain slices preparation in vitro. *A Journal of Progress in Neurosurgery, Neurology and Neurosciences.*

[18] M. Raza and U. Heinemann. Anticonvulsant actions of ska-31 in pilocarpine treated chronic epileptic rats. *A Journal of Progress in Neurosurgery, Neurology and Neurosciences*, 333.

[19] D. de Biase T. Ravizza F. Vasuri M. Martucci C. Keller E. Bellavista T. J. Buchholz P. M . Kloetzel et al M. Mishto, M. L. Raza. The immunoproteasome 5i subunit is a key contributor to ictogenesis in a rat model of chronic epilepsy. brain, behaviour and immunity. *A Journal of Progress in Neurosurgery, Neurology and Neurosciences*, 49.

[20] Yearwood J Venkatesh S Palaniswami M Liu C Li P, Karmakar C. Detection of epileptic seizure based on entropy analysis of shortterm eeg. *PLoS ONE.*

[21] Pelin YILDIRIM and Derya BIRANT. Naive bayes classiïňĄer for continuous variables using novel method (nbc4d) and distributions. *Research Gate.*

[22] Tayab Din Memon Umema Tasleem Shaikh, Imtiaz Hussain Kalwar and Faiza Shaikh. Design of iir filter using pso algorithm and its implementation in fpga. *Indian Journal of Science and Technology*, 10.

[23] Radu Dogaru. A modified naÃŕve bayes classifier for efficient implementations in embedded systems. *ISSCS 2011 - International Symposium on Signals, Circuits and Systems.*

[24] Andrew Hunter Patrick Dickinson Hongying Meng, Kofi Appiah. Fpga implementation of naive bayes classifier for visual object recognition.

[25] E. T. Graham P. Y. Siwakoti. Design of fpga-controlled power electronics and drives using matlab simulink. *IEEE ECCE Asia Downunder.*

[26] Stephan Van Beek and Sudhir Sharma. Four best practices for protyping matlab and simulink algorithms on fpgas, veriïňĄcation horizons. *Mathworks.*

[27] Abd Kadir Mahamad M. Hairol Jabbar Norliza Othman, Farhanahani Mahmud and Nur Atiqah Adon. Cardiac excitation modeling: Hdl coder optimization towards fpga stand-alone implementation. *IEEE International Conference, Penang, Malaysia.*